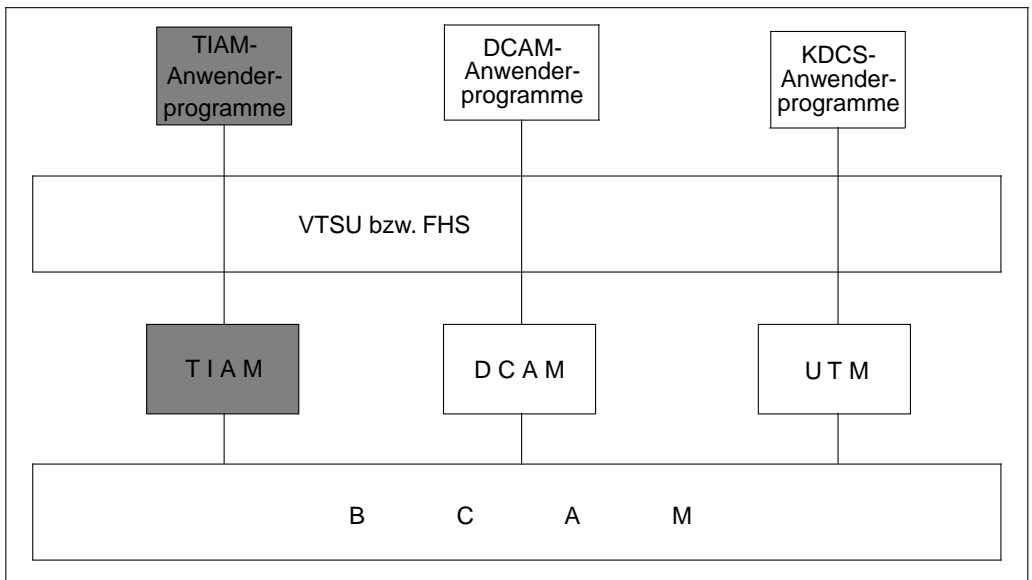


# 1 Einleitung

TIAM (Terminal Interactive Access Method) ist eine Zugriffsmethode des BS2000, die es Ihnen ermöglicht im Dialog mit dem Rechner so zu arbeiten, als stünde er Ihnen allein zur Verfügung. Sie können sämtliche Dienste und die Kommandosprache des BS2000 nutzen, die BS2000-Dienstprogramme verwenden sowie mit allen Anwendungsprogrammen arbeiten, für die Sie eine Berechtigung haben. Sie müssen das Programm, mit dem Sie arbeiten wollen, selbst laden. Sie führen eine eigene Task, die sie selbst einleiten, steuern und wieder beenden. Bevor Sie mit TIAM arbeiten können, müssen Sie zunächst die logische Verbindung zu der Kommunikationsanwendung \$DIALOG herstellen (siehe auch Handbuch "Netzzugang zu Datenstationen" [8]), die vom System bereitgestellt wird.

Zu den Hauptaufgaben, die mit dieser Form des Dialogbetriebs bearbeitet werden, zählen z.B. Programmentwicklung und Programmtest sowie das Aufbereiten von Dateien.



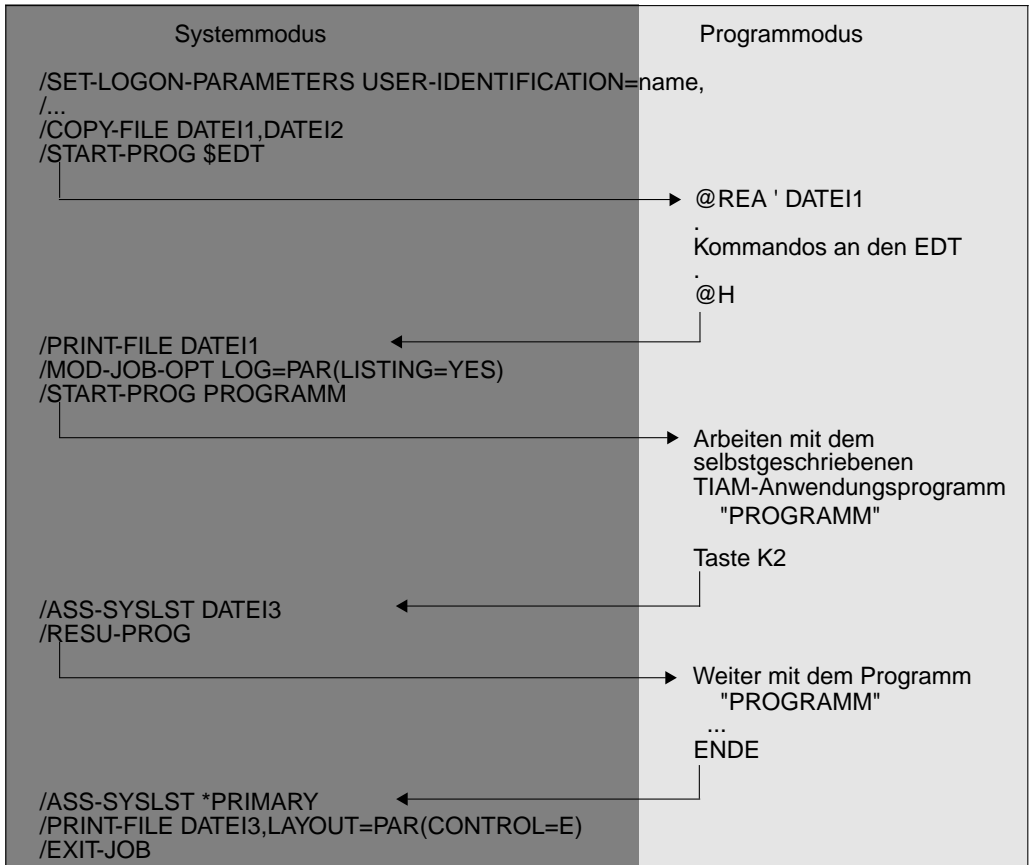
TIAM in der Systemumgebung

Sie haben zwei Möglichkeiten, mit TIAM zu arbeiten: im Systemmodus und im Programm-Modus.

Im **Systemmodus** arbeiten Sie direkt mit dem Betriebssystem. Dazu stehen Ihnen die **TIAM-Kommandos** und die Kommandosprache des BS2000 zur Verfügung. In diesem Modus stellt TIAM das Kommando MODIFY-TERMINAL-OPTIONS (und dessen ISP-Entsprechung TCHNG) bereit, das es dem Benutzer ermöglicht, die Ausführung der Task zu steuern und die Datenstationseigenschaften zu definieren. Darüber hinaus steht das Kommando SHOW-TERMINAL-OPTIONS zur Verfügung, mit dem sich der Benutzer die Datenstationseigenschaften anzeigen lassen kann. Die Kommandoschnittstellen sind in diesem Dokument definiert.

Im **Programmmodus** arbeiten Sie mit einem Programm, das Sie im Systemmodus mit dem START-PROGRAM- oder LOAD-PROGRAM-Kommando aufrufen. Dies kann z.B. ein Dienstprogramm oder ein selbst erstelltes Programm sein. Zum Erstellen dieser Programme bietet TIAM Makroaufrufe (für den Assembler-Programmierer) und COBOL-, PL/I-, C- und FORTRAN-Aufrufe.

Ein Wechsel vom Programmmodus in den Systemmodus ist jederzeit möglich.



Wechsel zwischen Systemmodus und Programmmodus

## 1.1 Zielgruppen des TIAM-Handbuchs

Dieses Handbuch wendet sich an BS2000-Anwender und Programmierer die im BS2000 mit der Zugriffsmethode TIAM arbeiten.

Zum Verständnis dieses Handbuchs sind Grundkenntnisse des BS2000 und Kenntnisse der verwendeten Programmiersprache (Assembler, COBOL, FORTRAN, PL/I und C) nötig.

## 1.2 Konzept des TIAM-Handbuchs

In diesem Handbuch werden Kommandos, Makro- und COBOL-Aufrufe beschrieben, welche die Zugriffsmethode TIAM betreffen.

Im **Kapitel 'TIAM-Kommandos'** werden die Kommandos die TIAM betreffen, im SDF-Format beschrieben. Sie sind alphabetisch geordnet.

Das **Kapitel 'TIAM-Makroaufrufe'** beschreibt die TIAM-Makroaufrufe. Mit ihnen kann die Ein- und Ausgabe gesteuert werden. Die Makroaufrufe betreffen Assembler-Programmierer. Die Aufrufe sind alphabetisch geordnet.

In den folgenden sprachspezifischen Kapiteln **'TIAM-COBOL-Schnittstelle'**, **'TIAM-FORTRAN-Schnittstelle'**, **'TIAM-PL/1-Schnittstelle'** und **'TIAM-C-Schnittstelle'** werden sowohl die Datenstrukturen für die TIAM-Aufrufe wie auch die TIAM-Aufrufe selbst für die verschiedenen Programmier-Schnittstellen beschrieben. Mit den TIAM-Aufrufen kann die Ein- und Ausgabe gesteuert werden.

Im **Kapitel 'POSIX-Schnittstellen'** wird die Unterstützung der POSIX-Schnittstellen durch TIAM beschrieben.

Im **Anhang** sind einige wichtige Tabellen und Übersichten zusammengefaßt.

## 1.3 Änderungen gegenüber Vorgängerausgabe

### VTSU

Zum Thema VTSU gibt es ein eigenes Handbuch (siehe [1] im Literaturverzeichnis). Deshalb wurden die VTSU-spezifischen Themen aus diesem Handbuch entfernt. Dies betrifft

- die VTSU-Makroaufrufe DCSTA, VTCSET, VTSUCB,
- die Datenstrukturen LINE-MODE-CONTROL-CHARACTERS und VTSU-CONTROL-BLOCK für die Programmier-Schnittstellen,
- das Kapitel „Einstellung von Betriebsparametern“,
- Das Programm PLUS zum Laden und Sichern von Programmtasten.

### Kommando: MODIFY-TERMINAL-OPTIONS

Im TIAM-Kommando `MODIFY-TERMINAL-OPTIONS` wurde der Operand `CODED-CHARACTER-SET` um den Parameter *name 1.. 8* erweitert. Mit diesem Parameter können Sie einer Datensichtstation einen Zeichensatz zuweisen.

### Kommando: SHOW-TERMINAL-OPTIONS

Das Kommando `SHOW-TERMINAL-OPTIONS` ist neu. Damit können Sie sich Bildeigenschaften ausgeben lassen.

### Makro: TSTAT

Der Operand *länge* des Makros `TSTAT` hat neue Werte.

### TIAM-C-Schnittstelle

Die Programmbeispiele im Kapitel „TIAM-C-Schnittstelle“ wurden überarbeitet.

### POSIX-Schnittstelle

Ab der Version V11.2A unterstützt TIAM die POSIX-Schnittstellen zur Daten-Ein- und Ausgabe. Dies ist im Kapitel „POSIX-Schnittstellen“ beschrieben.

## 1.4 Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei. Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter dem Dateinamen `SYSRME.produkt.version.sprache`. Die Benutzerkennung, unter der sich die Readme-Datei befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Die Readme-Datei können Sie mit dem Kommando `SHOW-FILE` oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
/PRINT-FILE FILE-NAME=dateiname,LAYOUT-CONTROL=  
PARAMETERS(CONTROL-CHARACTERS=EBCDIC)
```

*ab SPOOL V3.0A:*

```
/PRINT-DOCUMENT FROM-FILE=dateiname, DOCUMENT-FORMAT=*PAGE-FORMAT(  
CONTROL-MODE=*PAGE-MODE(LINE-SPACING=*BY-EBCDIC-CONTROL))
```



## 2 TIAM-Kommandos

Die Kommandos sind in alphabetischer Reihenfolge aufgeführt. Nach der Operandenbeschreibung folgen Beispiele, welche die Anwendung erleichtern sollen.

Zunächst erhalten Sie eine Beschreibung der Metasyntax für die SDF-Kommandobeschreibung. Ausführliche Informationen über die Verwendung von SDF-Kommandos finden Sie im Handbuch "SDF Einführung in die Dialogschnittstelle" [5].

### 2.1 SDF-Syntaxdarstellung

Das Bild zeigt ein Beispiel für die Syntaxdarstellung eines Kommandos in einem Handbuch. Das Kommandoformat besteht aus einem Feld mit dem Kommandonamen. Anschließend werden alle Operanden mit den zulässigen Operandenwerten aufgelistet. Struktureinleitende Operandenwerte und die von ihnen abhängigen Operanden werden zusätzlich aufgelistet.

<b>HELP-SDF</b> Kurzname: <b>HPSDF</b>
<b>GUIDANCE-MODE = *NO / *YES</b>
<b>,SDF-COMMANDS = *NO / *YES</b>
<b>,ABBREVIATION-RULES = *NO / *YES</b>
<b>,GUIDED-DIALOG = *YES (...)</b>
<b>*YES(...)</b>
<b>SCREEN-STEPS = *NO / *YES</b>
<b>,SPECIAL-FUNCTIONS = *NO / *YES</b>
<b>,FUNCTION-KEYS = *NO / *YES</b>
<b>,NEXT-FIELD = *NO / *YES</b>
<b>,UNGUIDED-DIALOG = *YES (...)/ *NO</b>
<b>*YES(...)</b>
<b>SPECIAL-FUNCTIONS = *NO / *YES</b>
<b>,FUNCTION-KEYS = *NO / *YES</b>

Bild 1: Syntaxdarstellung des Benutzer-Kommandos HELP-SDF

## Metasyntax

Kennzeichnung	Bedeutung	Beispiele
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Schlüsselwörter. Schlüsselwörter für konstante Operandenwerte beginnen mit *	<b>HELP-SDF</b>
<b>GROSSBUCHSTABEN in Halbfett</b>	Großbuchstaben in Halbfett kennzeichnen garantierte bzw. vorgeschlagene Abkürzungen der Schlüsselwörter.	<b>SCREEN-STEPS = *NO</b>
=	Das Gleichheitszeichen verbindet einen Operandennamen mit den dazugehörigen Operandenwerten.	<b>GUIDANCE-MODE = *YES</b>
< >	Spitze Klammern kennzeichnen Variablen, deren Wertevorrat durch Datentypen und ihre Zusätze beschrieben wird.	<b>GUIDANCE-MODE = *NO</b>
<u>Unterstreich</u>	Der Unterstrich kennzeichnet den Default-Wert eines Operanden.	<b>SYNTAX-FILE = &lt;full-filename 1..54&gt;</b>
/	Der Schrägstrich trennt alternative Operandenwerte.	<b>GUIDANCE-MODE = *NO</b>
(...)	Runde Klammern kennzeichnen Operandenwerte, die eine Struktur einleiten.	<b>NEXT-FIELD = *NO / *YES</b>
[ ]	Eckige Klammern kennzeichnen struktureinleitende Operandenwerte, deren Angabe optional ist. Die nachfolgende Struktur kann ohne den einleitenden Operandenwert angegeben werden.	<b>,UNGUIDED-DIALOG = *YES (...)/ *NO</b>
Einrückung	Die Einrückung kennzeichnet die Abhängigkeit zu dem jeweils übergeordneten Operanden.	<b>SELECT = [*BY-ATTRIBUTES](...)</b>
		<b>GUIDED-DIALOG = *YES (...)</b> <b>*YES(...)</b> <b>SCREEN-STEPS = *NO / *YES</b>

Metasyntax (Abschnitt 1 von 2)



Kennzeichnung	Bedeutung	Beispiele
<p style="text-align: center;"> </p> <p>list-poss(n):</p>	<p>Der Strich kennzeichnet zusammengehörende Operanden einer Struktur. Sein Verlauf zeigt Anfang und Ende einer Struktur an. Innerhalb einer Struktur können weitere Strukturen auftreten. Die Anzahl senkrechter Striche vor einem Operanden entspricht der Strukturtiefe.</p> <p>Das Komma steht vor weiteren Operanden der gleichen Strukturstufe.</p> <p>Aus den list-poss folgenden Operandenwerten kann eine Liste gebildet werden. Ist (n) angegeben, können maximal n Elemente in der Liste vorkommen. Enthält die Liste mehr als ein Element, muß sie in runde Klammern eingeschlossen werden.</p>	<p><b>SUPPORT = *TAPE(...)</b></p> <p style="margin-left: 20px;"><b>*TAPE(...)</b></p> <p style="margin-left: 40px;"><b>VOLUME = *ANY(...)</b></p> <p style="margin-left: 60px;"><b>*ANY(...)</b></p> <p style="margin-left: 80px;">  ...</p> <p><b>GUIDANCE-MODE = *NO / *YES</b></p> <p><b>,SDF-COMMANDS = *NO / *YES</b></p> <p>list-poss: <b>*SAM / *ISAM</b></p> <p>list-poss(40): &lt;structured-name 1..30&gt;</p> <p>list-poss(256): <b>*OMF / *SYSLST(...)</b> / &lt;full-filename 1..54&gt;</p>
<p>Kurzname:</p>	<p>Der darauf folgende Name ist ein garantierter Aliasname des Kommando- bzw. Anweisungsnamen.</p>	<p><b>HELP-SDF</b>      Kurzname: <b>HPSDF</b></p>

Metasyntax (Abschnitt 2 von 2)

## 2.2 Kurzbeschreibung der Kommandos

### SDF-Kommandos

Kommando	Kurzbeschreibung
EXIT-JOB	Dieses Kommando beendet den laufenden Auftrag, gibt die von ihm belegten virtuellen Speicherseiten und Geräte frei und stellt die Ausgabe-Systemdateien zur Ausgabe auf Drucker oder Band bereit.
LOGOFF	Dieses Kommando beendet den laufenden Auftrag, gibt die von ihm belegten virtuellen Speicherseiten und Geräte frei und stellt die Ausgabe-Systemdateien zur Ausgabe auf Drucker oder Band bereit.
MODIFY-JOB-OPTIONS	Mit diesem Kommando können Sie Vereinbarungen ändern, die die Protokollierung des Auftrags betreffen.
MODIFY-MSG-OPTIONS	Mit diesem Kommando können Meldungen empfangen oder unterdrückt werden, die der Operator an den Benutzer über MESSAGE- oder BROADCAST-Kommando sendet.
MODIFY-TERMINAL-OPTIONS	Mit diesem Kommando können Sie die Ausgabesteuerung für die Datenstation verändern.
SET-LOGON-PARAMETERS	Mit diesem Kommando können Sie einen Dialogauftrag an der Datenstation einleiten.
SHOW-TERMINAL-OPTIONS	Mit diesem Kommando können Sie sich Datenstationseigenschaften anzeigen lassen

## 2.3 Auftrag beenden: EXIT-JOB

**Anwendungsbereich:** JOB

**Privilegierung:** Alle Gruppen

### Funktionsbeschreibung

Das EXIT-JOB-Kommando beendet den laufenden Auftrag. Anschließend werden die vom Auftrag belegten virtuellen Speicherseiten und Geräte freigegeben und die Ausgabe-Systemdateien zur Ausgabe auf Drucker bereitgestellt.

Wurden während des Auftrags neue Dateigenerationen erstellt, so gibt das System die Namen der betroffenen Dateigenerationen aus, deren Basiswert sowie die Namen der ersten und der aktuellen Dateigeneration.

#### *Hinweis*

Die Ausgabe der Systemdateien auf Band wird, im Gegensatz zu dem Kommando LOGOFF, nicht unterstützt.

### Format

<pre> EXIT-JOB  MODE = *NORMAL / *ABNORMAL ,SYSTEM-OUTPUT = *ALL / *NONE / *PARAMETERS(...)   *PARAMETERS(...)         SYSYST-OUTPUT = *NONE / *PRINTER         SYSOUT-OUTPUT = *NONE / *PRINTER ,KEEP-CONNECTION = *NO / *YES </pre>
---

### Operandenbeschreibung

#### **MODE =**

Gibt an, ob der Auftrag normal oder abnormal beendet werden soll.

#### **MODE = \*NORMAL**

Der Auftrag wird normal beendet. Die Statusanzeige einer überwachenden Jobvariablen wird auf '\$T' gesetzt.

**MODE = \*ABNORMAL**

Der Auftrag wird abnormal beendet. Die Statusanzeige einer überwachenden Jobvariablen wird auf '\$A' gesetzt. Ob ein Dump erzeugt wird, ist abhängig von den innerhalb der Task vereinbarten Testoptionen (siehe Kommando MODIFY-TEST-OPTIONS, Operand DUMP).

**SYSTEM-OUTPUT =**

Steuert die Ausgabe der Systemdateien \*SYSLST und \*SYSOPT, für Stapelaufträge zusätzlich die Ausgabe der Systemdatei \*SYSOUT.

**SYSTEM-OUTPUT = \*ALL**

Alle nicht leeren Systemdateien werden ausgegeben: \*SYSLST und \*SYSOUT auf Drucker, \*SYSOPT auf Diskette.

**SYSTEM-OUTPUT = \*NONE**

Die Ausgabe der Systemdateien wird unterdrückt.

**SYSTEM-OUTPUT = \*PARAMETERS(...)**

Steuert, welche der beiden Systemdateien (\*SYSLST bzw. \*SYSOUT) auf den Drucker ausgegeben werden.

**SYSLST-OUTPUT = \*NONE**

Die Systemdatei \*SYSLST wird nicht ausgegeben.

**SYSLST-OUTPUT = \*PRINTER**

Die Systemdatei \*SYSLST wird auf den Drucker ausgegeben

**SYSOUT-OUTPUT = \*NONE**

Die Systemdatei \*SYSOUT wird nicht ausgegeben.

**SYSOUT-OUTPUT = \*PRINTER**

Die Systemdatei \*SYSOUT wird auf den Drucker ausgegeben

**KEEP-CONNECTION =**

Gibt an, ob die Rechnerverbindung weiterbestehen soll, damit sofort ein neuer Auftrag gestartet werden kann.

Dieser Operand ist nur im Dialog erlaubt, er wird im Stapelbetrieb ignoriert.

**KEEP-CONNECTION = \*NO**

Baut die Rechnerverbindung ab.

**KEEP-CONNECTION = \*YES**

Erhält die Rechnerverbindung. Ein neuer Auftrag kann sofort gestartet werden.

## Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando ausgeführt <sup>1)</sup>
	1	CMD0202	Syntaxfehler
	32	CMD0221	Systemfehler; es erfolgt keine Meldung nach SYSOUT

1) Kann nicht ausgewertet werden, da in diesem Fall die Task beendet wird!

### Hinweise

- Kommandos in Nicht-S-Prozeduren und ENTER-Prozeduren:  
Löst ein Fehler in der Kommandofolge den Spin-Off-Mechanismus aus, so wird zum nächsten der folgenden Kommandos verzweigt: EXIT-JOB, LOGOFF, SET-JOB-STEP, CANCEL-PROCEDURE, END-PROCEDURE oder EXIT-PROCEDURE.
- Wird ein EXIT-JOB-Kommando angegeben, während ein Programm geladen ist, so wird eine ggf. definierte ABEND-STXIT-Routine aktiviert. Dies gilt nicht, wenn das EXIT-JOB-Kommando über den CMD-Makro aufgerufen wird (siehe Handbuch "Makroaufrufe" [3]).

### Beispiel 1: Abnormales Auftragsende und Testoption DUMP=STD

```

/mod-test-opt dump=std
/exit-job mode=abnormal,keep-conn=yes
% IDA0N51 PROGRAM INTERRUPT AT LOCATION '00000000 (NONAME), (CDUMP)'
% IDA0N45 DUMP DESIRED? REPLY (Y = USER/AREA DUMP; Y,SYSTEM = SYSTEM DUMP;
N = NO)?n
% EXC0419 /LOGOFF AT 1223 ON 93-02-23 FOR TSN '2DXV'
% EXC0421 CPU TIME USED: 0.1210 , SERVICE UNITS: 000000000001854
% JMS0150 INSTALLATION ' H120-I', BS2000 VERSION 'V110', HOST 'D516ZE04':
PL EASE ENTER '/SET-LOGON-PARAMETERS' OR '?'

```

### Beispiel 2: Abnormales Auftragsende und Testoption DUMP=STD

```

/mod-test-opt dump=yes
/exit-job mode=abnormal,keep-connection=yes
% IDA0N51 PROGRAM INTERRUPT AT LOCATION '00000000 (NONAME), (CDUMP)'
% IDA0N53 DUMP BEING PROCESSED. PLEASE HOLD ON
% IDA0N54 'USERDUMP' WRITTEN TO FILE '$USER1.DUMP.ULF.2D2L.00001'
% IDA0N55 TITLE: 'TSN-2D2L UID-USER1 AC#-M0815RAN USERDUMP PC-00000000
EC-50 VERS-110 DUMP-TIME 12:25:23 93-02-23'
% EXC0419 /LOGOFF AT 1225 ON 93-02-23 FOR TSN '2D2L'
% EXC0421 CPU TIME USED: 0.0738 , SERVICE UNITS: 000000000001040
% JMS0150 INSTALLATION ' H120-I', BS2000 VERSION 'V110', HOST 'D516ZE04':
PLEASE ENTER '/SET-LOGON-PARAMETERS' OR '?'

```

## 2.4 Auftrag beenden: LOGOFF

**Anwendungsbereich:** JOB

**Privilegierung:** Alle Gruppen

### Funktionsbeschreibung

Das LOGOFF-Kommando beendet den laufenden Auftrag. Anschließend werden die vom Auftrag belegten virtuellen Speicherseiten und Geräte freigegeben und die Ausgabe-Systemdateien zur Ausgabe auf Drucker oder Band bereitgestellt.

Wurden während des Auftrags neue Dateigenerationen erstellt, so gibt das System die Namen der betroffenen Dateigenerationen aus, deren Basiswert sowie die Namen der ersten und aktuellen Dateigeneration.

### Hinweis

Das Kommando LOGOFF wird noch kompatibel unterstützt, aber zum Beenden des Auftrags sollte das Kommando EXIT-JOB verwendet werden. Das Kommando EXIT-JOB bietet dem Benutzer eine erweiterte Funktionalität.

### Format

<b>LOGOFF</b>
<b>KEEP-CONNECTION = *NO / *YES</b> <b>,SYSTEM-OUTPUT = *PRINT / *DELETE / *TAPE-OUTPUT</b>

### Operandenbeschreibung

#### **KEEP-CONNECTION =**

Gibt an, ob die Rechnerverbindung weiterbestehen soll, damit sofort ein neuer Auftrag gestartet werden kann.

Dieser Operand ist nur im Dialog erlaubt, er wird im Stapelbetrieb ignoriert.

#### **KEEP-CONNECTION = \*NO**

Baut die Rechnerverbindung ab.

#### **KEEP-CONNECTION = \*YES**

Erhält die Rechnerverbindung. Ein neuer Auftrag kann sofort gestartet werden.

**SYSTEM-OUTPUT =**

Gibt an, ob Systemdateien auszugeben sind und vereinbart das Ausgabemedium. SYSLST und SYSOUT werden nicht ausgegeben, wenn sie leer sind. Eine Ausgabe erfolgt ebenfalls nicht, wenn bei Systemgenerierung NOSPOOL eingestellt wurde (CLASS-2-OPTION: SSMLGOF1).

**SYSTEM-OUTPUT = \*PRINT**

Gibt die Systemdateien SYSLST und SYSOUT (Stapelbetrieb) auf Drucker aus. Für Dialogaufträge kann eine zusätzliche Abfrage ausgegeben werden, wenn dies bei Systemgenerierung eingestellt wurde (CLASS-2-OPTION: SSMLGOF1).

**SYSTEM-OUTPUT = \*DELETE**

Die Ausgabe der Systemdatei wird unterdrückt.

**SYSTEM-OUTPUT = \*TAPE-OUTPUT**

Gibt die Systemdateien auf Band aus. SYSLST und SYSOUT (Stapelbetrieb) werden auf Band in die Datei *TAPE.TSNnnnn* geschrieben, wobei *nnnn* die Auftragsnummer des mit LOGOFF beendeten Auftrags ist.

SYSOPT wird auf ein gesondertes Band ebenfalls in die Datei *TAPE.TSNnnnn* geschrieben, wobei *nnnn* eine neue Auftragsnummer ist. Diese Nummer wird auf SYSOUT ausgegeben.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando ausgeführt <sup>1)</sup>
	1	CMD0202	Syntaxfehler
	32	CMD0221	Systemfehler; es erfolgt keine Meldung nach SYSOUT

<sup>1)</sup> Kann nicht ausgewertet werden, da in diesem Fall die Task beendet wird!

*Hinweise*

- Kommandos in Nicht-S-Prozeduren und ENTER-Prozeduren: Löst ein Fehler in der Kommandofolge den Spin-Off-Mechanismus aus, so wird zum nächsten der folgenden Kommandos verzweigt: EXIT-JOB, LOGOFF, SET-JOB-STEP, CANCEL-PROCEDURE, END-PROCEDURE oder EXIT-PROCEDURE.
- Wird ein LOGOFF-Kommando angegeben, während ein Programm geladen ist, so wird eine ggf. definierte ABEND-STXIT-Routine aktiviert. Dies gilt nicht, wenn das LOGOFF-Kommando über den CMD-Makro aufgerufen wird, (siehe Handbuch "Makroaufrufe" [3]).

*Beispiel 1***/LOGOFF KEEP-CON=YES**

```
% EXC0419 LOGOFF AT 0847 ON 85-03-19, FOR TSN 1104
% EXC0421 USED CPU TIME: 4.8930 SECONDS
% PLEASE LOGON
```

Die Verbindung zum Rechner wird nicht abgebaut.

*Beispiel 2***/LOGOFF SYS-OUT=DEL**

```
% EXC0419 LOGOFF AT 0917 ON 85-03-19, FOR TSN 1154
% EXC0421 USED CPU TIME: 0.7934 SECONDS
```

Die Systemdatei SYSLST wird nicht ausgedruckt.



## 2.5 Protokollierung des Auftrags festlegen: MODIFY-JOB-OPTIONS

**Anwendungsbereich:** JOB

**Privilegierung:** STD-PROCESSING  
HARDWARE-MAINTENANCE  
SAT-FILE-EVALUATION  
SAT-FILE-MANAGEMENT  
SECURITY-ADMINISTRATION

### Funktionsbeschreibung

Das Kommando MODIFY-JOB-OPTIONS steuert

- die Ausgabeform der Systemmeldungen (Operand INFORMATION-LEVEL).
- die Ausgabe von Bedienungsplatzmeldungen (Operand OPERATOR-INTERACTION).
- die maximale Anzahl an Ausgabesätzen in SYSLST und SYSOPT (Operanden SYSLST- und SYSOPT-LIMIT).
- die Protokollierung des Auftragsablaufs (Operand LOGGING) für den laufenden Auftrag.

Die aktuellen Einstellungen können mit dem Kommando SHOW-JOB-OPTIONS ausgegeben werden.

### Format

MODIFY-JOB-OPTIONS	Kurzname: MDJO
<pre> INFORMATION-LEVEL = *UNCHANGED / *STD / *MEDIUM / *MINIMUM ,OPERATOR-INTERACTION = *UNCHANGED / *STD / *NO / *YES ,SYSLST-LIMIT = *UNCHANGED / *STD / *NO-LIMIT / &lt;integer 0..999999&gt; ,SYSOPT-LIMIT = *UNCHANGED / *STD / *NO-LIMIT / &lt;integer 0..999999&gt; ,LOGGING = PARAMETERS (...)   *PARAMETERS(...)       LISTING = *UNCHANGED / *STD / *NO / *YES       ,HARDCOPY = *UNCHANGED / *STD / *NO / *YES </pre>	

### Operandenbeschreibung

**INFORMATION-LEVEL =**  
Ausgabeform für Systemmeldungen.

**INFORMATION-LEVEL = \*UNCHANGED**

Der während des laufenden Auftrags zuletzt mit Kommando MODIFY-JOB-OPTIONS festgelegte Wert gilt (aktuelle Vereinbarung). Wurde kein Wert festgelegt, so gilt der von der Systemverwaltung festgelegte Standard-Wert.

**INFORMATION-LEVEL = \*STD**

Der von der Systemverwaltung festgelegte Standard-Wert gilt (Klasse-2-Systemparameter ECLMSG).

**INFORMATION-LEVEL = \*MEDIUM**

Systemmeldungen werden unverkürzt ausgegeben.

**INFORMATION-LEVEL = \*MINIMUM**

Systemmeldungen werden in kodierter Kurzform ausgegeben.

**OPERATOR-INTERACTION =**

Gibt an, ob Bedienungsplatzmeldungen und Operatorantworten auszugeben sind. Operator-Kommandos, die zur Steuerung des laufenden Auftrags dienen (z.B. Prioritätsänderung), sowie allgemeine Warn- und Fehlermeldungen für den Operator werden hier nicht berücksichtigt.

**OPERATOR-INTERACTION = \*UNCHANGED**

Der während des laufenden Auftrags zuletzt mit Kommando MODIFY-JOB-OPTIONS festgelegte Wert gilt (aktuelle Vereinbarung). Wurde kein Wert festgelegt, so gilt der von der Systemverwaltung festgelegte Standard-Wert.

**OPERATOR-INTERACTION = \*STD / \*NO**

Bedienungsplatzmeldungen und Operatorantworten werden nicht ausgegeben.

**OPERATOR-INTERACTION = \*YES**

Bedienungsplatzmeldungen und Operatorantworten werden ausgegeben.

**SYSLST-LIMIT =**

Maximale Anzahl von Sätzen, die während des laufenden Auftrags nach SYSLST ausgegeben werden dürfen.

Ausgabesätze auf SYSOUT werden nicht mitgezählt.

Die Angabe darf die in der Jobklassendefinition festgelegte Grenze nicht überschreiten (Grenze lässt sich über Kommando SHOW-JOB-CLASS erfragen).

Ist während der Protokollierung der angegebene Wert erreicht, so wird der Auftrag im Stapelbetrieb abnormal beendet. Im Dialogbetrieb erfragt das System, ob der Auftrag beendet oder fortgesetzt werden soll. Bei Fortsetzung des Auftrags wird der Zähler auf Null gesetzt und es gilt wieder der vereinbarte Wert.

**SYSLST-LIMIT = \*UNCHANGED**

Der während des laufenden Auftrags zuletzt mit Kommando MODIFY-JOB-OPTIONS festgelegte Wert gilt (aktuelle Vereinbarung). Wurde kein Wert festgelegt, so gilt der Maximalwert der Jobklassendefinition.

**SYSLST-LIMIT = \*STD**

Der in der Jobklassendefinition festgelegte Wert gilt.

**SYSLST-LIMIT = \*NO-LIMIT**

Unbegrenzte Anzahl von Sätzen, die während des laufenden Auftrags nach SYSLST ausgegeben werden dürfen.

Ist in der Jobklassendefinition ein niedrigerer Wert angegeben, so wird \*NO-LIMIT abgewiesen.

**SYSLST-LIMIT = <integer 0..999999>**

Legt die maximale Anzahl von Sätzen fest, die während des laufenden Auftrags nach SYSLST ausgegeben werden dürfen. Die Angabe darf die in der Jobklassendefinition festgelegte Grenze nicht überschreiten.

**SYSOPT-LIMIT =**

Maximale Anzahl von Sätzen, die während des laufenden Auftrags nach SYSOPT ausgegeben werden dürfen.

Ausgabesätze auf SYSOUT werden nicht mitgezählt.

Die Angabe darf die in der Jobklassendefinition festgelegte Grenze nicht überschreiten (Grenze lässt sich über Kommando SHOW-JOB-CLASS erfragen).

Ist während der Protokollierung der angegebene Wert erreicht, so wird der Auftrag im Stapelbetrieb abnormal beendet. Im Dialogbetrieb fragt das System ab, ob der Auftrag beendet oder fortgesetzt werden soll. Bei Fortsetzung des Auftrags wird der Zähler auf Null gesetzt und es gilt wieder der vereinbarte Wert.

**SYSOPT-LIMIT = \*UNCHANGED**

Der während des laufenden Auftrags zuletzt mit Kommando MODIFY-JOB-OPTIONS festgelegte Wert gilt (aktuelle Vereinbarung). Wurde kein Wert festgelegt, so gilt der Maximalwert der Jobklassendefinition.

**SYSOPT-LIMIT = \*STD**

Der in der Jobklassendefinition festgelegte Wert gilt.

**SYSOPT-LIMIT = \*NO-LIMIT**

Unbegrenzte Anzahl von Sätzen, die während des laufenden Auftrags nach SYSOUT ausgegeben werden dürfen.

Ist in der Jobklassendefinition ein niedrigerer Wert angegeben, so wird \*NO-LIMIT abgewiesen.

**SYSOPT-LIMIT = <integer 0..999999>**

Legt die maximale Anzahl von Sätzen fest, die während des laufenden Auftrags nach SYSOUT ausgegeben werden dürfen. Die Angabe darf die in der Jobklassendefinition festgelegte Grenze nicht überschreiten.

**LOGGING = \*PARAMETERS(...)**

Angaben über Protokollierung des Auftragsablaufs.

**LISTING =**

Gibt an, ob der Auftragsablauf zusätzlich auf SYSLST zu protokollieren ist. Bedienungsplatzmeldungen und Operator-Antworten (Operand OPERATOR-INTERACTION) werden bei Protokollierung auf SYSLST zusätzlich mit der Uhrzeit versehen.

Nicht protokolliert werden Systemmeldungen, die eine Antwort des Benutzers erfordern, sowie die Meldung ABNORMAL PROGRAM TERMINATION.

Im Zeilenmodus erfolgt die Protokollierung zeilenweise, d.h. Steuerzeichen NL werden ausgewertet.

Im Formatmodus (Menü) wird fortlaufend protokolliert, d.h. das Format wird im Protokoll zerstört dargestellt, Steuerzeichen NL werden nicht ausgewertet.

**LISTING = \*UNCHANGED**

Der während des laufenden Auftrags zuletzt mit Kommando MODIFY-JOB-OPTIONS festgelegte Wert gilt (aktuelle Vereinbarung). Wurde kein Wert festgelegt, so gilt der von der Systemverwaltung festgelegte Standard-Wert.

**LISTING = \*STD / \*NO**

Der Auftragsablauf soll nicht zusätzlich auf SYSLST protokolliert werden.

**LISTING = \*YES**

Der Auftragsablauf soll zusätzlich auf SYSLST protokolliert werden.

**HARDCOPY =**

Der Operand wird im Stapelbetrieb ignoriert.

Gibt an, ob der Auftragsablauf zusätzlich auf Hardcopy-Drucker zu protokollieren ist. Formate (Menüs) können nicht protokolliert werden.

**HARDCOPY = \*UNCHANGED**

Der während des laufenden Auftrags zuletzt mit Kommando MODIFY-JOB-OPTIONS festgelegte Wert gilt (aktuelle Vereinbarung). Wurde kein Wert festgelegt, so gilt der von der Systemverwaltung festgelegte Standard-Wert.

**HARDCOPY = \*STD / \*NO**

Der Auftragsablauf soll nicht zusätzlich auf Hardcopy-Drucker protokolliert werden.

**HARDCOPY = \*YES**

Der Auftragsablauf soll zusätzlich auf Hardcopy-Drucker protokolliert werden.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando ausgeführt
	1	CMD0202	Syntaxfehler
	32	CMD0221	Systemfehler
	64	JMS0630	Semantischer Fehler (siehe SYSOUT-Meldung)

*Beispiel 1*

```

/mod-job-opt inf-level=min,oper-Interact=yes,log=par(list=yes) _____(01)
/xxx
% CMD0186 XXX
/mod-job-opt inf-level=med _____(02)
/xxx
% CMD0186 OPERATION NAME 'XXX' UNKNOWN
/mod-job-opt log=par(list=no) _____(03)

```

- (01) Es gelten folgende Vereinbarungen:
- Kodierte Kurzform der Meldungen auf SYSOUT
  - Protokollierung der Bedienungsplatzmeldungen und Operatorantworten auf SYSOUT
  - Protokollierung auf SYSLST
- (02) Die Vereinbarungen werden geändert in:
- Unverkürzte Ausgabe der Meldungen auf SYSOUT
- (03) Die Protokollierung auf SYSLST wird ausgeschaltet.

*Beispiel 2*

```

/mod-job-opt syslst-limit=100
/show-file-attr out=*syslst
% SSM2222 SPECIFIED MAXLST LIMIT REACHED. CONTINUE? REPLY (Y=YES; N=NO)?n
% EXC0736 ABNORMAL TASK TERMINATION. ERROR CODE 'SSM2223': /HELP-MSG SSM2223
% EXC0419 /LOGOFF AT 1821 ON 93-03-17 FOR TSN '3F8W'
% EXC0421 CPU TIME USED: 81.3804 , SERVICE UNITS: 000000000678129
% SSM2075 SPOOLOUT REQUESTED FOR FILE 'SYSLST'? REPLY (Y=YES; N=NO)?

```

Für die Ausgabe nach SYSLST wird ein Satzlimit von 100 Sätzen vereinbart. Die Ausgabe des Kommandos SHOW-FILE-ATTRIBUTES wird nach SYSLST gelenkt, wobei das Satzlimit erreicht wird. Die Meldung SSM2222 wird mit N (NO) beantwortet. Die Task wird abnormal beendet.

Beispiel 3

```

/ass-syslst lst.file _____(01)
/mod-job-opt syslst-limit=10,log=par(list=yes) _____(02)
/show-file-attr sdfu.sam.,output=*syslst _____(03)
SSM2222 SPECIFIED MAXLST LIMIT REACHED. CONTINUE? REPLY (Y=YES; N=NO)?y
% SSM2222 SPECIFIED MAXLST LIMIT REACHED. CONTINUE? REPLY (Y=YES; N=NO)?y
%:N: PUBLIC: 12 FILES RES= 5535 FREE= 128 REL= 117 PAGES
/ass-syslst *primary _____(04)
/show-file lst.file _____(05)
    
```

```

      (IN) MOD-JOB-OPT SYSLST-LIMIT=10,LOG=PAR(LIST=YES)
      (IN) SHOW-FILE-ATTR SDFU.SAM.,OUTPUT=*SYSLST
AFILE STATUS SDFU.SAM.,LIST=((SYSLST),STANDARD)
    
```

FILENAME	PAM-PAGES	FREE-PAGES	SE A
:N:\$USERXY01.SDFU.SAM.ALT (OUT) % SSM2222 SPECIFIED MAXLST LIMIT REACHED. CONTINUE? REPLY (Y= (IN) Y	1854	0	
:N:\$USERXY01.SDFU.SAM.ALT.INH	18	0	
:N:\$USERXY01.SDFU.SAM.ALT.STW	39	0	
:N:\$USERXY01.SDFU.SAM.FORM	1824	2	
:N:\$USERXY01.SDFU.SAM.FORM.INH	18	0	
:N:\$USERXY01.SDFU.SAM.FORM.STW	39	1	
:N:\$USERXY01.SDFU.SAM.HFORTX	291	2	
:N:\$USERXY01.SDFU.SAM.HFORT1	618	1	
:N:\$USERXY01.SDFU.SAM.HFORT2	261	76	
+ S*SOF+	1(		1)

```

|:N:$USERXY01.SDFU.SAM.HFORT2.ORIG | 261 | 2 |
(O) % SSM2222 SPECIFIED MAXLST LIMIT REACHED. CONTINUE? REPLY (Y=
(IN) Y
|:N:$USERXY01.SDFU.SAM.HFORT3 | 156 | 43 |
|:N:$USERXY01.SDFU.SAM.HFORT3.ORIG | 56 | 1 |
    
```

PUBLIC SPACE:	12 FILES	5535	
(OUT) :N: PUBLIC:	12 FILES RES=	5535 FREE=	128 REL
(IN) SHOW-FILE LST.FILE			
(OUT) % SH00003 'DMS' REPORTED ERROR '0D99'. COMMAND NOT PROCESSED			
(IN) ASS-SYSLST *P			
% SH00301 WARNING: END OF FILE REACHED			
e S*SOF+	24(		1)

- (01) Der Systemdatei SYSLST wird die Datei *LST.FILE* zugewiesen.
- (02) Die Protokollierung nach SYSLST wird eingeschaltet und ein Satzlimit von 10 Sätzen wird vereinbart.
- (03) Die Ausgabe des Kommandos SHOW-FILE-ATTRIBUTES für alle Dateien, die mit *SDFU.SAM*. beginnen, erfolgt nach SYSLST. Das Satzlimit wird dabei zweimal überschritten (die Meldung *SSM2222* wird *Y* beantwortet).
- (04) Die Systemdatei SYSLST erhält wieder die Primärzuweisung.
- (05) Mit dem Kommando SHOW-FILE wird der Inhalt der Protokolldatei *LST.FILE* ausgegeben.

## 2.6 Empfang von Operatormeldungen steuern: MODIFY-MSG-OPTIONS

**Anwendungsbereich:** JOB

**Privilegierung:** STD-PROCESSING  
HARDWARE-MAINTENANCE  
SAT-FILE-EVALUATION  
SAT-FILE-MANAGEMENT  
SECURITY-ADMINISTRATION

### Funktionsbeschreibung

Mit dem Kommando MODIFY-MSG-OPTIONS kann der Benutzer für seinen Auftrag die Ausgabe von Operator-Meldungen unterdrücken, die dieser

- an den Benutzer sendet (auftragsbezogen): MESSAGE),
- an alle Benutzer sendet (BROADCAST).

Dringende Nachrichten, z.B. Nachrichten bei Systembeendigung, werden dennoch ausgegeben.

### Format

**MODIFY-MSG-OPTIONS**

**OPERATOR-BROADCAST = \*UNCHANGED / \*YES / \*NO**

**,OPERATOR-MSG = \*UNCHANGED / \*YES / \*NO**

**,INFO-OUTPUT = \*UNCHANGED / list-poss(2): \*STD / \*SYSTEMLINE**

### Operandenbeschreibung

**OPERATOR-BROADCAST = \*UNCHANGED / \*YES / \*NO**

Gibt an, ob der Benutzer Meldungen empfangen will, die ihm der Operator mit dem BROADCAST-Kommando sendet.

**OPERATOR-MSG = \*UNCHANGED / \*YES / \*NO**

Gibt an, ob der Benutzer Meldungen empfangen will, die ihm der Operator mit dem MESSAGE-Kommando sendet.

**INFO-OUTPUT = \*UNCHANGED / list-poss(2): \*STD / \*SYSTEMLINE**

Gibt an, wie die vom Operator gesendeten Meldungen an der Datensichtstation auszugeben sind.

Mit \*STD werden die Meldungen in der aktuellen Bildschirmzeile ausgegeben.



Mit \*SYSTEMLINE erfolgt die Ausgabe in der Anzeigenzeile (unterste Bildschirmzeile).  
Bei der Angabe beider Werte erfolgt die Ausgabe sowohl in der aktuellen Bildschirmzeile,  
als auch in der Anzeigenzeile.

### Kommando-Returncode

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando ohne Fehler ausgeführt
1	32	NBR0940	Kommando nicht ausgeführt, da kein Eintrag in den TCB möglich war

## 2.7 Datenstationseigenschaften ändern: MODIFY-TERMINAL-OPTIONS

**Anwendungsbereich:** JOB

**Privilegierung:** STD-PROCESSING  
TSOS  
HARDWARE-MAINTENANCE  
SAT-FILE-EVALUATION  
SAT-FILE-MANAGEMENT  
SECURITY-ADMINISTRATION

### Funktionsbeschreibung

Mit dem Kommando MODIFY-TERMINAL-OPTIONS kann der Benutzer die logischen Eigenschaften seiner Datenstation ändern. Das Kommando ist nur im Teilnehmerbetrieb anwendbar und wirkt nur für den Betrieb als Zeilen- oder Seiten-Datenstation. Die logischen Eigenschaften der Datensichtstation sind im System festgelegt durch Klasse-2-Systemparameter bei Systemgenerierung, durch Generierung der Datensichtstation, durch die Datensichtstation selbst und durch VTSU-Betriebsparameter. Zu Beginn der Dialogtask gelten diese festgelegten Werte, die der Benutzer innerhalb seiner Task mit dem Kommando MODIFY-TERMINAL-OPTIONS ändern kann. Folgende logische Eigenschaften kann der Benutzer einstellen:

Logische Eigenschaft	Operand	Voreinstellung durch
Art der Steuerung des Bildschirmüberlaufs	OVERFLOW-CONTROL	Klasse-2-Systemparameter TCHOFFLO
Anzahl Zeilen, nach der ein Bildschirmüberlauf erfolgt	MAXIMUM-LINES	Generierung der Datenstation
Länge von Ausgabezeilen	LINE-LENGTH	Generierung der Datenstation
Steuerung der Abbildung von Ausgaben und Art der Eingabe	WRITE-READ-MODE	Klasse-2-Systemparameter TCHREAD
Zeilenendezeichen	LINE-END-CHARACTER	VTSU-Betriebsparameter
Beachtung von Gross-/Kleinschreibung	LOWER-CASE	Generierung der Datenstation
Verwendung von Grafikzeichen	GRAPHICS	Generierung der Datenstation
Verwendung eines APL-Zeichensatzes	APL-CHARACTER-SET	Generierung der Datenstation
Unterstützung eines Hardcopy-Druckers	HARDCOPY	Generierung der Datenstation

Logische Eigenschaft	Operand	Voreinstellung durch
Ersatz-Zeichen zur Darstellung von nicht abbildbaren Zeichen	SUBSTITUTE-CHARACTER	Abhängig vom Typ der Datenstation: <ul style="list-style-type: none"> <li>– X' 4A' für Typ 8193</li> <li>– X' 44' für Typ 3270</li> <li>– X' 07' für alle anderen Typen ("Schmierzeichen")</li> </ul>
Quittungsverfahren	ACKNOWLEDGE-OUTPUT	Klasse-2-Systemparameter TCHTACK
Codiertabelle	CODED-CHARACTER-SET	Abhängig vom <ul style="list-style-type: none"> <li>– Typ der Datenstation</li> <li>– Benutzereintrag</li> <li>– VTSU-Betriebsparameter (siehe Tabelle unten)</li> </ul>

Der Default-Wert \*UNCHANGED in den entsprechenden Operanden bedeutet jeweils, daß die bisherige Vereinbarung gilt.

Die folgende Tabelle zeigt die voreingestellten Codiertabellen für CODED-CHARACTER-SET in Abhängigkeit von Stationstyp, Benutzereintrag für den Standard-Code und dem VTSU-Betriebsparameter TIAM-PERM8.

Stationstyp	7-Bit	8-Bit	8-Bit	8-Bit
Standard CCS	-	EDF03iRV	EDFxxx	EDFxxx
TIAM-PERM8	-	-	N	Y
Codiertabelle	EDF03iRV	EDF03iRV	EDF03iRV	EDFxxx

## Format

<b>MODIFY-TERMINAL-OPTIONS</b>  <b>OVERFLOW-CONTROL = *<u>UNCHANGED</u> / *NO-CONTROL / *USER-ACKNOWLEDGE / *TIME(...)</b> <b>*TIME(...)</b>   <b>TIMEOUT = *STD / &lt;integer 0..60&gt;</b> <b>,MAXIMUM-LINES = *<u>UNCHANGED</u> / &lt;integer 3..255&gt;</b> <b>,LINE-LENGTH = *<u>UNCHANGED</u> / &lt;integer 10..255&gt;</b> <b>,LINE-END-CHARACTER = *<u>UNCHANGED</u> / *NONE / &lt;c-string 1..1&gt;</b> <b>,WRITE-READ-MODE = *<u>UNCHANGED</u> / *MODIFIED-FIELDS / *NO-FIELDS</b> <b>,LOWER-CASE = *<u>UNCHANGED</u> / *YES / *NO</b> <b>,GRAPHICS = *<u>UNCHANGED</u> / *YES / *NO</b> <b>,APL-CHARACTER-SET = *<u>UNCHANGED</u> / *YES / *NO</b> <b>,HARDCOPY = *<u>UNCHANGED</u> / *NO / *LOCAL / *CENTRAL</b> <b>,SUBSTITUTE-CHARACTER = *<u>UNCHANGED</u> / *STD / &lt;c-string 1..1&gt; / &lt;x-string 1..2&gt;</b> <b>,ACKNOWLEDGE-OUTPUT = *<u>UNCHANGED</u> / *YES / *NO</b> <b>,CODED-CHARACTER-SET = *<u>UNCHANGED</u> / *7-BIT / *8-BIT-DEFAULT / &lt;name 1..8&gt;</b>
---

## Operandenbeschreibung

**OVERFLOW-CONTROL = \*UNCHANGED / \*NO-CONTROL / \*USER-ACKNOWLEDGE / \*TIME(...)**

Art der Steuerung bei Bildschirmüberlauf.

Ein Überlauf wird vom System jeweils angenommen, wenn die Anzahl der (noch) auszugebenden Bildschirmzeilen die maximale Zeilenanzahl (MAXIMAL-LINES) überschreitet. Bei Beginn einer Ausgabe wird die letzte Eingabe des Benutzers bei der Berechnung des Bildschirmüberlaufs berücksichtigt.

Erfolgt die Überlaufkontrolle durch Quittung (\*USER-ACKNOWLEDGE), zählt die vom System ausgegebene Quittungsaufforderung zu der maximal auszugebenden Anzahl von Bildschirmzeilen.

**OVERFLOW-CONTROL = \*NO-CONTROL**

Keine Überlaufkontrolle. Das System trifft keine Vorkehrungen, bei langen Bildschirmausgaben die ESCAPE-Funktion (K2-Taste) zu ermöglichen bzw. das vorzeitige Überschreiben von Daten am Bildschirm zu verhindern.

**OVERFLOW-CONTROL = \*USER-ACKNOWLEDGE**

Überlaufkontrolle durch Quittung. Bei Bildschirmüberlauf fordert das System den Benutzer an der Datensichtstation mit der Meldung *% PLEASE ACKNOWLEDGE* zur Eingabe einer Quittung auf. Der Benutzer kann damit selbst bestimmen, wann die Ausgabe nach einem

Bildschirmüberlauf fortgesetzt werden soll. Als Quittung gelten alle Eingaben, verschieden von ESCAPE und BREAK. Nach Empfang einer Quittung wird die Ausgabe mit dem nächsten Bildschirm fortgesetzt.

**OVERFLOW-CONTROL = \*TIME(...)**

Die Überlaufkontrolle erfolgt durch Zeitsteuerung. Nach Ablauf der eingestellten Wartezeit wird die Bildschirmausgabe fortgesetzt. Bei Erreichen der letzten Bildschirmzeile wird mit jeder weiteren Ausgabezeile der angezeigte Bildschirminhalt entsprechend nach oben "verschoben", wodurch Ein- oder Ausgabedaten zu Beginn des Bildschirms überschrieben werden.

**TIMEOUT = \*STD / <integer 0..60>**

Wartezeit in Sekunden, die bei Bildschirmüberlauf verstreichen soll, bevor weitere Ausgaben erfolgen.

**TIMEOUT = \*STD**

Es gilt der bei der Systemgenerierung vereinbarte Standardwert (6 Sekunden) oder der zuletzt vom Benutzer vereinbarte Wert.

*Hinweis*

Der für die Zeitsteuerung eingestellte Wert (TIMEOUT) gilt auch bei Wechsel des Bildschirmmodus (z.B. Wechsel nach Beendigung einer zeilenorientierten Ausgabe zurück in den formatorientierten Eingabemodus des geführten Dialogs).

**MAXIMUM-LINES = \*UNCHANGED / <integer 3..255>**

Maximale Anzahl von Zeilen, die bis zur nächsten Überlaufsteuerung ausgegeben werden dürfen.

**LINE-LENGTH = \*UNCHANGED / <integer 10..255>**

Der Operand wird nur noch kompatibel unterstützt. Die Angabe wird nicht ausgewertet.

**LINE-END-CHARACTER = \*UNCHANGED / \*NONE / <c-string 1..1>**

Zeilenendezeichen, das an jede Ausgabezeile der Datenstation (nur für 8110 und 3270) anzufügen ist.

Es kann ein beliebiges von der Tastatur eingebbares Zeichen sein, mit dem das logische Zeilenende bei Ausgaben angezeigt wird. Anschließend wird in der nächsten Zeile fortgefahren. Bei Eingaben wird dieses Zeichen im Eingabetext vom System als NEW LINE an das Benutzerprogramm weitergegeben. (Voreinstellung durch das System ist "\"" bzw. "ö".)

**LINE-END-CHARACTER = \*NONE**

Kein Zeilenendezeichen. Das logische Zeilenende wird bei der Bildschirmausgabe nur durch Zeilenwechsel dargestellt. Bei Eingabe steht die Funktion nicht zur Verfügung.

**WRITE-READ-MODE = \*UNCHANGED / \*MODIFIED-FIELDS / \*NO-FIELDS**

Steuert die Abbildung der Ausgabenachrichten und Art der Eingabe.

**WRITE-READ-MODE = \*MODIFIED-FIELDS**

Lesen modifizierter Felder. Die Anzeige von Daten auf dem Bildschirm erfolgt feldorientiert. An das System werden nur die modifizierten Felder übertragen.

**WRITE-READ-MODE = \*NO-FIELDS**

Lesen ungeschützter Felder. Die Anzeige von Daten auf dem Bildschirm erfolgt ohne Felder, die Eingabe von Bildanfang bzw. Schreibmarke.

**LOWER-CASE = \*UNCHANGED / \*YES / \*NO**

Gibt an, ob die Ausgabe von Kleinbuchstaben möglich sein soll.

**GRAPHICS = \*UNCHANGED / \*YES / \*NO**

Gibt an, ob die Verwendung eines Grafikzusatzes möglich sein soll.

**APL-CHARACTER-SET = \*UNCHANGED / \*YES / \*NO**

Gibt an, ob die Verwendung eines APL-Zeichensatzes möglich sein soll.

**HARDCOPY = \*UNCHANGED / \*NO / \*LOCAL / \*CENTRAL**

Gibt an, welche Art von Hardcopy-Protokollierung möglich sein soll.

**HARDCOPY = \*NO**

Kein Hardcopy-Drucker.

**HARDCOPY = \*LOCAL**

Lokaler Hardcopy-Drucker direkt an der Datenstation betriebsbereit. Der Operand wird nur noch ausgewertet, falls beim Verbindungsaufbau ein lokales Hardcopy-Gerät zugewiesen wurde. Sonst wird der Operand ignoriert.

**HARDCOPY = \*CENTRAL**

Zentraler Hardcopy-Drucker an Kanal 0,...,31 der gleichen Mehrfachsteuerung betriebsbereit, an der auch die Datensichtstation angeschlossen ist. Es gilt der Kanal, welcher bei der Generierung der Datensichtstation im PDN angegeben wurde (Standard=0).

**SUBSTITUTE-CHARACTER = \*UNCHANGED / \*STD / <c-string 1..1> / <x-string 1..2>**

Ersatzzeichen, das anstelle nicht abdruckbarer Zeichen zu setzen ist.

Nicht abdruckbare Zeichen werden in Ausgabetexten durch das angegebene Ersatzzeichen ersetzt.

**SUBSTITUTE-CHARACTER = \*STD**

Ein gerätespezifisches Schmierzeichen wird als Ersatzzeichen verwendet.

**ACKNOWLEDGE-OUTPUT = \*UNCHANGED / \*YES / \*NO**

Gibt an, ob Datenstationsausgaben systemintern zu quittieren sind.

**ACKNOWLEDGE-OUTPUT = \*NO**

Keine interne Quittung.

Unmittelbar vor EXIT-JOB bzw. LOGOFF ausgegebene Nachrichten können durch Verbindungsabbau verloren gehen.

Der eingestellte Wert für ACKNOWLEDGE-OUTPUT gilt für alle folgenden Ausgaben im Programm- und Systemmodus bis zu einer Neueinstellung durch MODIFY-TERMINAL-OPTIONS oder EXIT-JOB bzw. LOGOFF (Voreinstellung bei der Systemgenerierung).

**CODED-CHARACTER-SET = \*UNCHANGED / \*7-BIT / \*8-BIT-DEFAULT / <name 1..8>**

Gibt an, ob die Datensichtstation im 7-Bit oder im 8-Bit-Modus arbeiten soll.

**CODED-CHARACTER-SET = \*7-BIT**

Deaktiviert den 8-Bit-Zeichensatz. Die Datensichtstation arbeitet im 7-Bit-Modus.

**CODED-CHARACTER-SET = \*8-BIT-DEFAULT**

Aktiviert die 8-Bit-Codetabelle des Benutzereintrags, falls dort eine 8-Bit-Codetabelle eingetragen ist. Die Datensichtstation arbeitet dann im 8-Bit-Modus. Enthält der Benutzereintrag keine 8-Bit-Codetabelle, arbeitet die Datensichtstation weiter im 7-Bit-Modus.

**CODED-CHARACTER-SET = <name 1..8>**

Name einer 8-Bit-Codetabelle (CCS)

Abhängig von der expliziten Angabe eines Codes im VTSUCB, können folgende Situationen auftreten

Standard CCS	7	7	7	7	7	7	name	name	name1	name	name	name1
8-bit-Datenstation	n	n	n	j	j	j	n	n	n	j	j	j
MOD-TERM-OPT C-C-S =	7	8	name	7	8	name	7	8	name2	7	8	name2
Ergebnis	1)	2)	2)	3)	4)	7)	1)	2)	2)	5)	6)	8)

- 1) Das Kommando wird angenommen.  
Die Datenstation arbeitet im 7-Bit-Modus.  
Erste von TSTAT gelieferte Information: Leerzeichen  
Zweite von TSTAT gelieferte Information: Leerzeichen
- 2) Das Kommando wird abgelehnt.  
Die Datenstation arbeitet im 7-bit-Modus.  
Erste von TSTAT gelieferte Information: Leerzeichen  
Zweite von TSTAT gelieferte Information: Leerzeichen
- 3) Das Kommando wird angenommen.  
Die Datenstation arbeitet im 7-bit-Modus, wenn kein VTSUCB benutzt wird, oder wenn Leerzeichen im VTSUCB festgelegt wurden.  
Erste von TSTAT gelieferte Information: Leerzeichen  
Zweite von TSTAT gelieferte Information: Leerzeichen

- 4) Das Kommando wird abgelehnt.  
Die Datenstation arbeitet im 7-bit-Modus, wenn kein VTSUCB benutzt wird, oder wenn Leerzeichen im VTSUCB festgelegt wurden.  
Erste von TSTAT gelieferte Information: Leerzeichen  
Zweite von TSTAT gelieferte Information: Leerzeichen
- 5) Das Kommando wird angenommen.  
Die Datenstation arbeitet im 7-bit-Modus, wenn kein VTSUCB benutzt wird oder wenn Leerzeichen im VTSUCB festgelegt wurden.  
Die Datenstation arbeitet im 8-bit-Modus, wenn im VTSUCB ein gültiger erweiterter Code-Name oder \*EXTEND festgelegt wurde. In diesem Fall wird der Code 'name' benutzt.  
Erste von TSTAT gelieferte Information: name  
Zweite von TSTAT gelieferte Information: Leerzeichen
- 6) Das Kommando wird angenommen.  
Die Datenstation arbeitet im 8-Bit-Modus (mit dem Zeichensatz 'name'), wenn kein VTSUCB benutzt wird oder wenn Leerzeichen im VTSUCB festgelegt wurde. Die Datenstation arbeitet ebenfalls im 8-Bit-Modus, wenn im VTSUCB ein gültiger Zeichensatzname oder \*EXTEND festgelegt wurde. In diesem Fall wird der Zeichensatz 'name' verwendet.  
Erste von TSTAT gelieferte Information: name  
Zweite von TSTAT gelieferte Information: name
- 7) Das Kommando wird angenommen, wenn 'name' von der Datenstation unterstützt wird.  
Die Datenstation arbeitet im 8-Bit-Modus (mit dem Zeichensatz 'name'), wenn kein VTSUCB benutzt wird oder wenn im VTSUCB kein Zeichensatzname angegeben ist.  
Erste von TSTAT gelieferte Information: Leerzeichen  
Zweite von TSTAT gelieferte Information: name
- 8) Das Kommando wird angenommen, wenn 'name2' von der Datenstation unterstützt wird.  
Die Datenstation arbeitet im 8-Bit-Modus (mit dem Zeichensatz 'name2'), wenn kein VTSUCB benutzt wird oder wenn im VTSUCB kein Zeichensatzname angegeben ist.  
  
Erste von TSTAT gelieferte Information: name1  
Zweite von TSTAT gelieferte Information: name2



**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
1	0	CMD0001	Kommando erfolgreich ausgeführt
	0	CMD0001	Kommando wirkt erst, wenn die Überlaufkontrolle wieder eingeschaltet wird
2	0	TIA0500	Kommando mit Warnung ausgeführt; mindestens eine Angabe wurde wegen semantischen Fehlers ignoriert
	1	TIA0502	Syntaxfehler
	32	TIA0122	Nicht behebbarer Fehler von VTSU-B
	64	CMD0216	Privilegienfehler

*Beispiel*

```
/show-file proc.time _____(01)
```

```

/      SET-PROC-OPT      ERROR-MECHANISM=*BY-RETURNCODE
/      SET-VAR           A = 1
/REPEAT-1: REPEAT
/      WRITE-TEXT '** Es ist jetzt &(TIME()) Uhr.'
/      SET-VAR A = (A + 1)
/      UNTIL CONDITION = (A > 10)
/EXIT: EXIT-PROC
/ERROR: IF-BLOCK-ERROR
/      WRITE-TEXT '** Fehler mit Subcode1 &(SC1) und Subcode2 (SC2) *'
/      HELP-MSG MSG-ID=&(MC)
/      END-IF
% SH00301 WARNING: END OF FILE REACHED
e                                     S*SOF+      1(      1

```

```
/mod-term-opt overflow-control=*no _____(02)
```

```
/call-proc proc.time _____(03)
```

```
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
** Es ist jetzt 17:31:43 Uhr.
```

```
/mod-term-opt ov-contr=*user-ack,max-lines=3 _____(04)
```

```
/call-proc proc.time _____(05)
```

```
** Es ist jetzt 17:33:46 Uhr.
%PLEASE ACKNOWLEDGE
** Es ist jetzt 17:33:46 Uhr.
** Es ist jetzt 17:34:14 Uhr.
%PLEASE ACKNOWLEDGE
** Es ist jetzt 17:34:14 Uhr.
** Es ist jetzt 17:34:27 Uhr.
```

```

%PLEASE ACKNOWLEDGE
** Es ist jetzt 17:34:27 Uhr.
** Es ist jetzt 17:34:29 Uhr.
%PLEASE ACKNOWLEDGE
** Es ist jetzt 17:34:29 Uhr.
** Es ist jetzt 17:34:30 Uhr.
%PLEASE ACKNOWLEDGE
** Es ist jetzt 17:34:30 Uhr.
/mod-term-opt ov-contr=*time(timeout=30) _____(06)

/call-proc proc.time _____(07)
** Es ist jetzt 17:35:32 Uhr.
** Es ist jetzt 17:35:32 Uhr.
** Es ist jetzt 17:35:32 Uhr.
** Es ist jetzt 17:36:02 Uhr.
** Es ist jetzt 17:36:02 Uhr.
** Es ist jetzt 17:36:02 Uhr.
** Es ist jetzt 17:36:32 Uhr.
** Es ist jetzt 17:36:32 Uhr.
** Es ist jetzt 17:36:32 Uhr.
** Es ist jetzt 17:37:02 Uhr.

```

- (01) Das Kommando `SHOW-FILE` gibt den Inhalt der Datei `PROC.TIME` aus. Es ist eine S-Prozedur, die zehnmal die aktuelle Uhrzeit ermittelt (Builtin-Funktion `TIME()`) und mit dem Kommando `WRITE-TEXT` nach `SYSOUT` ausgibt. Mit `E` (END) wird die Ausgabe des `SHOW-FILE`-Kommandos beendet.
- (02) Die Überlaufkontrolle wird ausgeschaltet (`OVERFLOW-CONTROL=*NO-CONTROL`).
- (03) Aufruf der Prozedur `PROC.TIME`. Da kein Bildschirmüberlauf erfolgt, wird die Meldung mit Uhrzeit zehnmal hintereinander ausgegeben. Wegen der sofortigen Ausgabe ändert sich die angezeigte Uhrzeit nicht.
- (04) Die Überlaufkontrolle wird auf `*USER-ACKNOWLEDGE` mit maximal 3 Ausgabezeilen eingestellt.
- (05) Aufruf der Prozedur `PROC.TIME`. Nach jeweils 3 Zeilen fordert das System eine Quittung vom Benutzer an. Der erste Überlauf erfolgt bereits nach der ersten Zeilenausgabe, da der Prozeduraufruf und die Quittungsaufforderung bei der Berechnung der maximalen Zeilenanzahl berücksichtigt werden. Der Benutzer quittiert z.B. mit der `DÜ`-Taste. Danach setzt sich die Ausgabe fort. Wegen der Verzögerung werden unterschiedliche Uhrzeiten ausgegeben.
- (06) Die Überlaufkontrolle wird auf eine Wartezeit von 30 Sekunden eingestellt. Die maximale Zeilenzahl wird nicht verändert.
- (07) Aufruf der Prozedur `PROC.TIME`. Nach jeweils 3 Zeilen erfolgt eine Wartezeit von 30 Sekunden. Die Wartezeit kann an den ausgegebenen Uhrzeiten abgelesen werden: Sie erhöht sich nach jeweils drei Ausgabezeilen um 30 Sekunden.

## 2.8 Dialogauftrag starten: SET-LOGON-PARAMETERS

**Anwendungsbereich:** JOB

**Privilegierung:** Alle Gruppen

### Funktionsbeschreibung

Mit dem Kommando SET-LOGON-PARAMETERS leitet der Benutzer einen Dialogauftrag auf der Datensichtstation ein. SET-LOGON-PARAMETERS als erstes Kommando einer ENTER-Datei leitet bei Start mit dem Kommando ENTER-JOB einen Stapelauftrag ein.

Die Angaben im Kommando SET-LOGON-PARAMETERS identifizieren den Anwender (Prüfung der Zugriffsberechtigung zum System), charakterisieren den Auftrag und bestimmen die Protokollführung über den Auftragsablauf.

Die Angaben zur Zugriffsberechtigung werden gegen den Benutzer-Eintrag geprüft; weitere Angaben zur Jobklasse und zu den Jobattributen (Job-, Runpriorität, Systemressourcen) auch gegen den Eintrag in der Jobklassendefinition. Diese Einträge sind dem Anwender über die Kommandos SHOW-USER-ATTRIBUTES bzw. SHOW-JOB-CLASS zugänglich. Stimmen die Angaben für RUN-PRIORITY (Priorität) und CPU-LIMIT in dem Benutzer-Eintrag und der Jobklassendefinition nicht überein, wird der für den Anwender günstigere Wert zugelassen.

Für Stapelaufträge werden die Operanden des SET-LOGON-PARAMETERS-Kommandos nur ausgewertet, wenn der Stapelauftrag vom Bedienungsplatz gestartet wird. Dabei werden aber nur Operanden ausgewertet, die der Operator nicht explizit im Kommando ENTER-JOB angegeben hat.

Die Operanden JOB-PRIORITY, RERUN-AFTER-CRASH, FLUSH-AFTER-SHUTDOWN, START und REPEAT-JOB sind nur für Stapelaufträge möglich.

## Format

SET-LOGON-PARAMETERS	Kurzname: <b>STLGP</b>
<p><b>USER-IDENTIFICATION</b> = <b>*NO</b> / &lt;name 1..8&gt;</p> <p><b>,ACCOUNT</b> = <b>*NONE</b> / &lt;alphanum-name 1..8&gt;</p> <p><b>,PASSWORD</b> = <b>*NONE</b> / &lt;c-string 1..8&gt; / &lt;c-string 9..32&gt; / &lt;x-string 1..16&gt; / <b>*SECRET</b></p> <p><b>,JOB-CLASS</b> = <b>*STD</b> / &lt;name 1..8&gt;</p> <p><b>,JOB-NAME</b> = <b>*NO</b> / &lt;name 1..8&gt;</p> <p><b>,MONJV</b> = <b>*NONE</b> / &lt;full-filename 1..54 without-gen-vers&gt;</p> <p><b>,JV-PASSWORD</b> = <b>*NONE</b> / &lt;c-string 1..4&gt; / &lt;x-string 1..8&gt; / <b>*SECRET</b> / &lt;integer -2147483648..2147483647&gt;</p> <p><b>,JOB-PRIORITY</b> = <b>*STD</b> / &lt;integer 1..9&gt;</p> <p><b>,RERUN-AFTER-CRASH</b> = <b>*NO</b> / <b>*YES</b></p> <p><b>,FLUSH-AFTER-SHUTDOWN</b> = <b>*NO</b> / <b>*YES</b></p> <p><b>,START</b> = <b>*STD</b> / <b>*SOON</b> / <b>*IMMEDIATELY</b> / <b>*AT-STREAM-STARTUP</b> / <b>*WITHIN(...)</b> / <b>*AT(...)</b> / <b>*EARLIEST(...)</b> / <b>*LATEST(...)</b></p> <p><b>*WITHIN(...)</b></p> <p>      <b>HOURS</b> = <b>0</b> / &lt;integer 0..23&gt;</p> <p>      <b>,MINUTES</b> = <b>0</b> / &lt;integer 0..59&gt;</p> <p><b>*AT(...)</b></p> <p>      <b>DATE</b> = <b>*TODAY</b> / &lt;date 8..10&gt;</p> <p>      <b>,TIME</b> = &lt;time&gt;</p> <p><b>*EARLIEST(...)</b></p> <p>      <b>DATE</b> = <b>*TODAY</b> / &lt;date 8..10&gt;</p> <p>      <b>,TIME</b> = &lt;time&gt;</p> <p><b>*LATEST(...)</b></p> <p>      <b>DATE</b> = <b>*TODAY</b> / &lt;date 8..10&gt;</p> <p>      <b>,TIME</b> = &lt;time&gt;</p>	

Fortsetzung ➔

```

,REPEAT-JOB = *STD / *NO / *DAILY / *WEEKLY / *AT-STREAM-STARTUP / *PERIOD(...)
    *PERIOD(...)
        |
        |   HOURS = 0 / <integer 0..23>
        |   ,MINUTES = 0 / <integer 0..59>
,RESOURCES = *PARAMETERS (...)
    *PARAMETERS(...)
        |
        |   RUN-PRIORITY = *STD / <integer 30..255>
        |   ,CPU-LIMIT = *STD / *NO / <integer 1..32767>
        |   ,SYSLST-LIMIT = *STD / *NO / <integer 0..999999>
        |   ,SYSOPT-LIMIT = *STD / *NO / <integer 0..999999>
,LOGGING = *PARAMETERS (...)
    *PARAMETERS(...)
        |
        |   LISTING = *NO / *YES
        |   ,HARDCOPY = *NO / *YES
,JOB-PARAMETER = *NO / <c-string 1..127>
,PROTECTION = *NONE / *CANCEL

```

### Operandenbeschreibung

**USER-IDENTIFICATION = \*NO / <name 1..8>**

Benutzerkennung, unter der der Dialogauftrag laufen soll.

**ACCOUNT = \*NONE / <alphanumeric-name 1..8>**

Abrechnungsnummer der Benutzerkennung.

**PASSWORD = \*NONE / <c-string 1..8> / <c-string 9..32> / <x-string 1..16> / \*SECRET**

Kennwort der Benutzerkennung.

Die Eingabe eines „langen“ Kennworts (entspricht <c-string 9..32>) wird unterstützt. Ein Hash-Algorithmus konvertiert das „lange“ Kennwort in die interne 8 byte lange Darstellung. Zur Vereinbarung „langer“ Kennwörter siehe Kommando MODIFY-USER-PROTECTION.

Der Operand PASSWORD ist als „geheim“ definiert:

- Der eingegebene Wert wird nicht protokolliert.
- Im geführten Dialog ist das Eingabefeld automatisch dunkelgesteuert.
- Die Angabe \*SECRET ist nur im ungeführten Dialog zulässig und ermöglicht die verdeckte Eingabe des gewünschten Wertes. SDF fordert zur Eingabe des „geheimen“ Wertes auf und stellt ein dunkelgesteuertes Eingabefeld zur Verfügung.

**JOB-CLASS = \*STD / <name 1..8>**

Jobklasse, in der der Dialogauftrag ablaufen soll. Die Jobklasse muß für Dialogaufträge zugelassen sein. Die für ihn zugelassenen Jobklassen kann der Benutzer seinem Benutzereintrag für den HOME-Pubset entnehmen (Ausgabe des Kommandos SHOW-USER-ATTRIBUTES). Hier wird auch die Default-Jobklasse angezeigt, die mit \*STD voreingestellt ist. Über die Eigenschaften von Jobklassen (Jobklassendefinition) kann sich der Benutzer mit dem Kommando SHOW-JOB-CLASS informieren.

**JOB-NAME = \*NO / <name 1..8>**

Name für den Dialogauftrag. Über diesen Namen kann der Dialogauftrag angesprochen werden (z.B. mit SHOW-JOB-STATUS). Alle Aufträge, die ohne eigenen Namen aus diesem Dialogauftrag gestartet werden, erhalten ebenfalls diesen Namen.

**MONJV = \*NONE / <full-filename 1..54 without-gen-vers>**

Nur bei Einsatz des Softwareproduktes JV

Name der JV, die den Dialogauftrag überwachen soll. Eine JV kann nur vereinbart werden, wenn das kostenpflichtige Subsystem JV zur Verfügung steht.

Eine Auftragsüberwachung wird nur gestartet, wenn der Dialogauftrag vom System angenommen wird (JOB ACCEPTED).

Der Auftraggeber muß schreibberechtigt sein, da er das System beauftragt, in die JV zu schreiben. Falls die JV zum Zeitpunkt Kommandoverarbeitung nicht zugreifbar ist, wird eine Fehlermeldung nach SYSOUT ausgegeben und das Kommando zurückgewiesen. Existiert die angegebene JV (eigene Benutzerkennung) noch nicht, wird sie vom System erzeugt (siehe Standardwerte von CREATE-JV). Über die angegebene JV kann der Benutzer seinen Dialogauftrag ansprechen (siehe Handbuch „Jobvariablen“ [10]):

\$R Auftrag läuft  
 \$T Auftrag normal beendet  
 \$A Auftrag abnormal beendet

**JV-PASSWORD = \*NONE / <c-string 1..4> / <x-string 1..8> / <integer -2147483648..2147483647> / \*SECRET**

Nur bei Einsatz des Softwareproduktes JV

Kennwort der JV.

Der Operand wird nur ausgewertet, wenn Auftragsüberwachung vereinbart wurde (siehe Operand MONJV). Der Operand JV-PASSWORD ist als „geheim“ definiert:

- Der eingegebene Wert wird nicht protokolliert.
- Im geführten Dialog ist das Eingabefeld automatisch dunkelgesteuert.
- Die Angabe \*SECRET ist nur im ungeführten Dialog zulässig und ermöglicht die verdeckte Eingabe des gewünschten Wertes. SDF fordert zur Eingabe des „geheimen“ Wertes auf und stellt ein dunkelgesteuertes Eingabefeld zur Verfügung.

**RESOURCES = \*PARAMETERS(...)**

Angaben zu Run-Priorität, CPU-Zeit, Maximalanzahl SYSLST- und SYSOPT-Sätze.

**RUN-PRIORITY = \*STD / <integer 30..255>**

Run-Priorität, die der Dialogauftrag erhalten soll. Je niedriger der Wert, desto höher die Priorität. Der maximal zulässige Wert ist in der Jobklassendefinition festgelegt und kann mit dem Kommando SHOW-JOB-CLASS abgefragt werden.

**RUN-PRIORITY = \*STD**

Es gilt die für die Jobklasse festgelegte Standardpriorität.

**CPU-LIMIT = \*STD / \*NO / <integer 1..32767>**

Maximale CPU-Zeit in Sekunden, die der Dialogauftrag verbrauchen darf. Die maximal erlaubte Zeit hängt von der vereinbarten Jobklasse ab. Siehe auch "Zeitlimitierung im BS2000" im Anhang von "BS2000/OSD-BC, Kommandos, Band 3" [2].

**CPU-LIMIT = \*STD**

Es gilt der Standardwert für die gewählte Jobklasse.

**CPU-LIMIT = \*NO**

Der Dialogauftrag soll ohne Zeitbegrenzung laufen (NTL = No Time Limit). Der Operandenwert ist nur erlaubt, wenn eine entsprechende Berechtigung im Benutzereintrag vorliegt.

**SYSLST-LIMIT = STD / NO / <integer 0..999999>**

Bezeichnet die maximale Anzahl von Sätzen, die vom Auftrag in die Systemdateien SYSLST, SYSLST01, SYSLST02, ..., SYSLST99 ausgegeben werden. Datensätze in der Systemdatei SYSOUT, die gleichzeitig nach SYSLST geschrieben werden, zählen nicht mit.

Die Angabe darf die in der Jobklassendefinition festgelegte Grenze nicht überschreiten. Dies kann mit dem Kommando SHOW-JOB-CLASS abgefragt werden.

**SYSLST-LIMIT = \*STD**

Standardwert der gewählten Jobklasse. Bei Überschreitung der angegebenen Anzahl gilt:

- Im Dialogbetrieb kann der Anwender angeben, ob der Auftrag fortgesetzt oder beendet werden soll. Bei Fortsetzung wird wieder bis „Anzahl“ ausgegeben.
- Im Stapelbetrieb wird der Auftrag abnormal beendet.

**SYSLST-LIMIT = \*NO**

Anzahl der Sätze ist nicht begrenzt.

**SYSOPT-LIMIT = \*STD / \*NO / <integer 0..999999>**

Bezeichnet die maximale Anzahl von Sätzen, die vom Auftrag in die Systemdatei SYSOPT ausgegeben werden.

Die Angabe darf die in der Jobklassendefinition festgelegte Grenze nicht überschreiten. Dies kann mit dem Kommando SHOW-JOB-CLASS abgefragt werden.

**SYSOPT-LIMIT = \*STD**

Standardwert der gewählten Jobklasse. Bei Überschreitung der angegebenen Anzahl gilt:

- Im Dialogbetrieb kann der Anwender angeben, ob der Auftrag fortgesetzt oder beendet werden soll. Bei Fortsetzung wird wieder bis „Anzahl“ ausgegeben.
- Im Stapelbetrieb wird der Auftrag abnormal beendet.

**SYSOPT-LIMIT = \*NO**

Anzahl der Sätze ist nicht begrenzt.

**LOGGING = \*PARAMETERS(...)****LISTING = \*NO / \*YES**

Gibt an, ob der Auftragsablauf zusätzlich auf SYSLST zu protokollieren ist.

**JOB-PARAMETER**

Angabe zusätzlicher Attribute für die gewählte Jobklasse, sofern die Systemverwaltung solche definiert und bekanntgegeben hat.

**JOB-PARAMETER = \*NO**

Keine zusätzlichen Attribute.

**JOB-PARAMETER = <c-string 1..127>**

Angabe eines Strings mit einer Folge beliebiger Zeichen, die von der Systemverwaltung zur Kennzeichnung weiterer Jobklassenattribute vergeben worden ist.

**PROTECTION = \*NONE / \*CANCEL**

Gibt an, ob der Dialogauftrag gegen eine versehentliche Beendigung mit dem Kommando CANCEL-JOB geschützt sein soll.

**PROTECTION = \*NONE**

Der Dialogauftrag ist nicht gegen eine versehentliche Beendigung geschützt.

**PROTECTION = \*CANCEL**

Der Dialogauftrag ist gegen eine versehentliche Beendigung geschützt. In Dialogaufträgen, die diesen Dialogauftrag mit dem Kommando CANCEL-JOB beenden wollen, fordert das System zusätzlich eine Bestätigung an. Ein versehentliches Beenden dieses Dialogauftrags durch fehlerhafte Angabe der Auftragsnummer soll somit verhindert werden.

**JOB-PRIORITY = \*STD / <integer 1..9>**

Nur für Stapelaufträge

Jobpriorität, die der Stapelauftrag erhalten soll.

Je niedriger der Wert, desto höher die Priorität. Der maximal zulässige Wert ist in der Jobklassendefinition festgelegt und kann mit dem Kommando SHOW-JOB-CLASS abgefragt werden.

**JOB-PRIORITY = \*STD**

Es gilt die für die Jobklasse festgelegte Standardpriorität.



**RERUN-AFTER-CRASH = \*NO / \*YES**

Nur für Stapelaufträge

Gibt an, ob der Stapelauftrag im nächsten Systemlauf neu zu starten ist, wenn die Bearbeitung aufgrund eines Systemfehlers oder Systemlaufende abgebrochen wurde.

**FLUSH-AFTER-SHUTDOWN = \*NO / \*YES**

Gibt an, ob der Stapelauftrag aus der Auftragswarteschlange zu entfernen ist, wenn er bis Systemlaufende nicht bearbeitet wurde.

**START =**

Nur für Stapelaufträge

Startzeitpunkt des Stapelauftrags. Angaben abweichend von STD werden nur ausgeführt, wenn sie gemäß Jobklassendefinition erlaubt sind (siehe Kommando SHOW-JOB-CLASS).

**START = \*STD**

Es gilt der Standardwert für die gewählte Jobklasse.

**START = \*SOON**

Der Auftrag soll unter Berücksichtigung seiner Priorität so bald als möglich gestartet werden.

**START = \*IMMEDIATELY**

Der Auftrag soll unmittelbar gestartet werden.

**START = \*AT-STREAM-STARTUP**

Der Auftrag soll nach dem nächsten Startup des Jobschedulers gestartet werden.

**START = \*WITHIN(...)**

Der Auftrag soll innerhalb des nachfolgend angegebenen Zeitraums gestartet werden.

**HOURS = 0 / <integer 0..23>**

Anzahl Stunden.

**MINUTES = 0 / <integer 0..59>**

Anzahl Minuten.

**START = \*AT(...)**

Der Auftrag soll exakt zum nachfolgend angegebenen Zeitpunkt gestartet werden.

**DATE = \*TODAY / <date 8..10>**

Datum; es kann in der Form yy-mm-dd bzw. yyyy-mm-dd angegeben werden.

**TIME = <time>**

Uhrzeit im Format hh:mm, wobei hh = Stunden und mm = Minuten sind.  
Sekundenangabe wird ignoriert.

Der Auftrag soll frühestens zum nachfolgend angegebenen Zeitpunkt gestartet werden.

**DATE = \*TODAY / <date 8..10>**

Datum; es kann in der Form yy-mm-dd bzw. yyyy-mm-dd angegeben werden.

**TIME = <time>**

Uhrzeit im Format hh:mm, wobei hh = Stunden und mm = Minuten sind.  
Sekundenangabe wird ignoriert.

**START = \*LATEST(...)**

Der Auftrag soll spätestens zum nachfolgend angegebenen Zeitpunkt gestartet werden.

**DATE = \*TODAY / <date 8..10>**

Datum; es kann in der Form yy-mm-dd bzw. yyyy-mm-dd angegeben werden.

**TIME = <time>**

Uhrzeit im Format hh:mm, wobei hh = Stunden und mm = Minuten sind.  
Sekundenangabe wird ignoriert.

**REPEAT-JOB =**

Nur für Stapelaufträge

Zeitintervall, in dem der Stapelauftrag wiederholt werden soll. Angaben abweichend von STD werden nur ausgeführt, wenn sie gemäß Jobklassendefinition erlaubt sind (siehe Kommando SHOW-JOB-CLASS). Die Zeitbasis für die Wiederholungen ist abhängig von der Angabe im Operanden START; siehe dazu den Hinweis „Kombinationen der Operanden START und REPEAT-JOB“ bei dem Kommando ENTER-JOB. Für die Wiederholungen gilt:

- Die i-te Wiederholung ( $i \geq 1$ ) eines Auftrages wird nur dann gestartet, wenn die (i-1)-te Ausführung beendet ist.
- Abbrechen des gerade laufenden Auftrages (i) hat keine Auswirkung auf den Start von (i+1); ( $i \geq 0$ ).
- Abbruch des gesamten Auftrages: es muß sowohl der gerade laufende Auftrag (i) als auch der Folgeauftrag (i+1) abgebrochen werden, ( $i \geq 0$ ); (CANCEL-JOB-Kommando oder mit dem Kommando MODIFY-JOB..., REPEAT-JOB=NO den Auftrag (i) zum letzten Auftrag der Folge erklären).

**REPEAT-JOB = \*STD**

Es gilt der Standardwert für die gewählte Jobklasse.

**REPEAT-JOB = \*NO**

Der Stapelauftrag wird nicht wiederholt.

**REPEAT-JOB = \*DAILY**

Tägliche Wiederholung zu der mit START angegebenen Uhrzeit.

**REPEAT-JOB = \*WEEKLY**

Wöchentliche Wiederholung zu der mit START angegebenen Uhrzeit.

**REPEAT-JOB = \*AT-STREAM-STARTUP**

Wiederholung nach jedem Startup des Jobschedulers.

**REPEAT-JOB = \*PERIOD(...)**

Wiederholung nach dem angegebenen Zeitintervall.

**HOURS = 0 / <integer 0..23>**

Anzahl Stunden.

**MINUTES = 0 / <integer 0..59>**

Anzahl Minuten.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
	0	CMD0001	Kommando ausgeführt
130	64	JMS0640	Kommando abgewiesen

Ist SET-LOGON-PARAMETERS das **erste** Kommando im Dialog (Eingabe nach Verbindungsaufbau und LOGON-Aufforderung) oder in einer ENTER-Datei, wird es im Fehlerfall (SC1 ungleich 0) zurückgewiesen und die Task abgebrochen. Der Kommando-Returncode kann in diesem Fall nicht ausgewertet werden.

## 2.9 Datenstationseigenschaften anzeigen: SHOW-TERMINAL-OPTIONS

**Anwendungsbereich:** JOB

**Privilegierung:** STD-PROCESSING  
TSOS  
HARDWARE-MAINTENANCE  
SAT-FILE-EVALUATION  
SAT-FILE-MANAGEMENT  
SECURITY-ADMINISTRATION

### Funktionsbeschreibung

Mit dem Kommando SHOW-TERMINAL-OPTIONS kann sich der Benutzer die logischen Eigenschaften seiner Datenstation nach SYSOUT ausgeben lassen. Die logischen Eigenschaften der Datensichtstation sind im System festgelegt durch Klasse-2-Systemparameter bei Systemgenerierung, durch Generierung der Datensichtstation, durch die Datensichtstation selbst und durch VTSU-Betriebsparameter. Zu Beginn der Dialogtask gelten diese festgelegten Werte, die der Benutzer innerhalb seiner Task mit dem Kommando MODIFY-TERMINAL-OPTIONS ändern kann.

Das Kommando unterstützt die strukturierte Ausgabe in S-Variable (siehe Handbuch „BS2000/OSD-BC Kommandos“ Band 4, „Ausgabe in S-Variablen“ [2])

### Format

<b>SHOW-TERMINAL-OPTIONS</b>

Das Kommando hat keine Operanden.

**Kommando-Returncode**

(SC2)	SC1	Maincode	Bedeutung / garantierte Meldungen
2	0	CMD0001	Kommando erfolgreich ausgeführt
	0	TIA0200	Information über Eigenschaften der Datensichtstation nicht verfügbar.
0	32	CMD2009	Fehler bei Erzeugung von S-Variablen
	64	CMD0216	Privilegienfehler
	1	CMD0202	Syntax Error

**Ausgabeformat**

Die Ausgabefelder und die angezeigten Werte entsprechen den Operanden und Operandenwerten des Kommandos MODIFY-TERMINAL-OPTIONS.

Ausgabefeld	mögliche Werte	Bedeutung
ACKNOWLEDGE-OUTPUT	YES / NO	Quittungsverfahren
APL-CHARACTER-SET	YES / NO	Verwendung eines APL-Zeichensatzes
CODED-CHARACTER-SET	7-BIT / 8-BIT-DEFAULT / <name 1..8>	Codiertabelle
GRAPHICS	YES / NO	Verwendung von Grafikzeichen
HARDCOPY	NO / LOCAL / CENTRAL	Unterstützung eines Hardcopy-Druckers
LINE-END-CHARACTER	Schmierzeichen (<x-string 1..1>)	Zeilenendezeichen
LINE-LENGTH	<integer 10..255>	Länge von Ausgabezeilen
LOWER-CASE	YES / NO	Beachtung von Gross-/Kleinschreibung
MAXIMUM-LINES	<integer 3..255>	Anzahl Zeilen, nach der ein Bildschirmüberlauf erfolgt
OVERFLOW-CONTROL	USER-ACKNOWLEDGE / NO-CONTROL / TIME(TIMEOUT=<integer 0..60>)	Art der Steuerung des Bildschirmüberlaufs
WRITE-READ-MODE	MODIFIED-FIELDS / NO-FIELDS	Steuerung der Abbildung von Ausgaben und Art der Eingabe
SUBSTITUTE-CHARACTER	Schmierzeichen (<x-string 1..1>)	Ersatz-Zeichen zur Darstellung von nicht abbildbaren Zeichen

*Beispiel*

```
/show-term _____ (1)
```

```
%OVERFLOW-CONTROL      = USER-ACKNOWLEDGE
%MAXIMUM-LINES          = 24          LINE-LENGTH          = 80
%LINE-END-CHARACTER     = ? (X'00')  WRITE-READ-MODE     = MODIFIED-FIELDS
%LOWER-CASE             = YES          GRAPHICS             = NO
%APL-CHARACTER-SET     = NO           HARDCOPY            = LOCAL
%SUBSTITUTE-CHARACTER  = ? (X'07')  ACKNOWLEDGE-OUTPUT = NO
%CODED-CHARACTER-SET   = 7-BIT
```

```
/mod-term ov-contr=*time(timeout=10) _____ (2)
```

```
/show-term _____ (3)
```

```
%OVERFLOW-CONTROL      = TIME(TIMEOUT = 10)
%MAXIMUM-LINES          = 24          LINE-LENGTH          = 80
%LINE-END-CHARACTER     = ? (X'00')  WRITE-READ-MODE     = MODIFIED-FIELDS
%LOWER-CASE             = YES          GRAPHICS             = NO
%APL-CHARACTER-SET     = NO           HARDCOPY            = LOCAL
%SUBSTITUTE-CHARACTER  = ? (X'07')  ACKNOWLEDGE-OUTPUT = NO
%CODED-CHARACTER-SET   = 7-BIT
```

```
/
```

- (1) Ausgabe der logischen Eigenschaften der Datensichtstation.
- (2) Die Überlaufkontrolle soll zeitgesteuert nach jeweils 10 Sekunden erfolgen.
- (3) Erneute Ausgabe der logischen Eigenschaften der Datensichtstation.

### 3 TIAM-Makroaufrufe

Die Makroaufrufe sind in alphabetischer Reihenfolge aufgeführt. Nach der Operandenbeschreibung folgen Beispiele, die die Anwendung erleichtern sollen. Die VTSU-Makros sind im VTSU-Benutzerhandbuch [1] beschrieben. Alle VTSU-spezifischen Informationen finden Sie dort.

#### 3.1 Verwendete Metasprache

Um eine möglichst einfache Handhabung zu bieten, sind in dieser Beschreibung Zeichen als sogenannte Metasymbole verwendet, die bereits weitgehend aus anderen Benutzerhandbüchern des BS2000 bekannt sind. Sie sind in der folgenden Tabelle erläutert.

Formale Darstellung	Erläuterung	Beispiel
GROSS- BUCHSTABEN	Großbuchstaben bezeichnen Konstanten, die in dieser Form vom Anwender eingegeben werden müssen.	DEVICE=DISKETTE  Anzugeben ist DEVICE=DISKETTE
Klein- buchstaben	Kleinbuchstaben bezeichnen Variablen, die bei der Eingabe vom Anwender durch aktuelle Werte ersetzt werden müssen, d.h. ihr Inhalt kann von Fall zu Fall verschieden sein.	SECTORS=anzahl  Anzugeben ist z.B. SECTORS=5 oder SECTORS=10 etc.
{ }	Geschweifte Klammern schließen Alternativen ein, d.h. aus den eingeschlossenen Größen kann oder muß eine Angabe ausgewählt werden.	TAPE={ YES } { NO }  Anzugeben ist TAPE=YES oder TAPE=NO

[ ]	Eckige Klammern schließen Wahlangaben ein, d.h. Angaben, die weggelassen werden können. Steht bei Wahlangaben das Komma innerhalb der Klammer, so muß es nur bei Verwendung dieser Wahlangabe geschrieben werden. Steht es hingegen außerhalb der Klammer, so muß es stets geschrieben werden, auch wenn die Wahlangabe nicht gemacht wird. (Runde Klammern müssen eingegeben werden!)	dateiname[,ERASE]  Anzugeben ist z.B. FILE,ERASE oder FILE oder XYZ,ERASE etc.
....	Die Unterstreichung hebt den Standardwert hervor. Das ist der Wert, den das System einsetzt, wenn der Anwender keine Angaben macht.	$\left\{ \begin{array}{l} \underline{\text{SAM}} \\ \text{ISAM} \end{array} \right\}$  Anzugeben ist SAM oder ISAM oder nichts (=SAM)
...	Punkte bedeuten eine Wiederholung. Sie zeigen an, daß die davorstehende Einheit mehrmals wiederholt werden kann.	(archivnr,...)  Anzugeben ist z. B. (PVT003) oder (PVT003,PVT456) usw.
()	Ein Ausdruck, der zur Darstellung von Variablen benutzt wird, steht in runden Klammern. Diese Darstellung soll auf einen Blick den Wertebereich zeigen. Da dazu mehrere Zeichen notwendig sind, ist auch diese formale Eingrenzung erforderlich.	(0 < länge < 9)
≤	Beziehung zweier Werte: Der links stehende Wert sei kleiner oder gleich dem rechts stehenden; der rechts stehende Wert sei größer oder gleich dem links stehenden.	nummer ≤ 2047 0 ≤ position
≥	Der links stehende Wert sei größer oder gleich dem rechts stehenden; der rechts stehende Wert sei kleiner oder gleich dem links stehenden.	anzahl ≥ 1
< >	Das oben erläuterte gilt ebenfalls, jedoch ohne die Gleichsetzung.	0 < länge menge < 9



**Kurzbeschreibung der Makroaufrufe**

<b>Makroaufrufe</b>	<b>Kurzbeschreibung</b>
CUPAB	generiert Adressierungshilfen (DSECTs) für die Operandentabellen der Makroaufrufe RDATA, WROUT und WRTRD.
RDATA	liest einen Satz von der Systemdatei SYSDTA.
TCHNG	ändert die Eigenschaften der logischen Datenstation.
TSTAT	übergibt Informationen über die Eigenschaften der Dialogstationen.
WROUT	überträgt einen Satz in die Systemdatei SYSOUT.
WRTRD	überträgt im Dialogbetrieb eine Nachricht zur Datenstation und übernimmt anschließend von dort eine Nachricht; siehe auch CUPAB.

## 3.2 Operandentabelle für Dialogkommunikation definieren: CUPAB

Mit dem Makroaufruf CUPAB kann der Anwender die Felder und Flags in den Operandentabellen der Aufrufe RDATA, WROUT und WRTRD symbolisch ansprechen. Der Makroaufruf CUPAB generiert hierzu einen Pseudoabschnitt (Dummy section, DSECT) für den Aufruf im 24-Bit-Adressierungsmodus. Um 31-Bit-fähige DSECTs zu erzeugen, müssen die Makros RDATA, WRTRD, WROUT mit MF=D (MF=D,Präfix) benutzt werden.

Name	Operation	
[name]	CUPAB	D

- name** Wenn im Namensfeld ein Eintrag angegeben ist, wird dieser Eintrag als der DSECT-Name verwendet. Enthält das Namensfeld keinen Eintrag, dann wird CUPAB als DSECT-Name generiert.
- D** Bewirkt die Generierung eines Pseudoabschnittes (DSECT). Wenn dieser Operand nicht angegeben ist, wird eine MNOTE-Meldung gegeben. Die Generierung der DSECT wird davon aber nicht beeinträchtigt.

Für die Operandentabelle des Aufrufs RDATA gelten folgende Feldnamen:

Feldname	Byte	Bedeutung
<u>CURAREAW</u>	0-3	Ganzwort, das CUREREDIT1 und CURAREA enthält
CUREREDIT1	0	Eingabe-Aufbereitungsbyte 1
CURAREA	1-3	Adresse des Anwender-Einlesebereiches
CURFTB	4	Flag
CUREREDIT2	5	Eingabe-Aufbereitungsbyte 2
CURALEN	6-7	Länge des Anwender-Einlesebereiches
<u>CURERRW</u>	8-11	Ganzwort, das CURACI und CURERROR enthält
CURACI	8	Indikator für SYSDTA-Zuweisungsänderung
CURERROR	9-11	Fehleradresse
L@RDATA		Länge der Operandentabelle für RDATA

Für die Operandentabelle des Aufrufs WROUT gelten folgende Feldnamen:

Feldname	Byte	Bedeutung
<u>CUWMSGW</u>	0-3	Ganzwort, das CUWEDIT1 und CUWMSG enthält
CUWEDIT1	0	Ausgabe-Aufbereitungsbyte 1
CUWMSG	1-3	Adresse der Nachricht im Anwendungsprogramm
<u>CUWERRW</u>	4-7	Ganzwort, das CUWEDIT2 und CUWERROR enthält
CUWEDIT2	4	Ausgabe-Aufbereitungsbyte 2
CUWERROR	5-7	Fehleradresse
L@WROUTB		Länge der Operandentabelle für WROUT

Für die Operandentabelle des Aufrufs WRTRD gelten folgende Feldnamen:

Feldname	Byte	Bedeutung
<u>CUBMSGW</u>	0-3	Ganzwort, das CUBOEDT1 und CUBMSG enthält
CUBOEDT1	0	Ausgabe-Aufbereitungsbyte 1
CUBMSG	1-3	Adresse des Nachrichtenausgabebereiches
<u>CUBAREAW</u>	4-7	Ganzwort, das CUBIEDT1 und CUBAREA enthält
CUBIEDT1	4	Eingabe-Aufbereitungsbyte 1
CUBAREA	5-7	Adresse des Eingabebereiches
CUBOEDT2	8	Ausgabe-Aufbereitungsbyte 2
CUBIEDT2	9	Eingabe-Aufbereitungsbyte 2
CUBALEN	10-11	Länge des Eingabebereiches
<u>CUBERRW</u>	12-15	Ganzwort, das CUBERROR enthält
reserviert	12	-
CUBERROR	13-15	Fehleradresse
L@WRTRDB		Länge der Operandentabelle für WRTRD

## Symbolische Konstanten für die Aufbereitungsbytes 1 und 2

CUPAB definiert außer den Feldnamen für die Operandentabellen zu RDATA, WROUT und WRTRD auch symbolische Konstanten (EQU-Werte für Assembler) für die Werte der Operanden edit, edit1 und edit2.

Die folgenden Tabellen geben eine Übersicht über die Namen der durch CUPAB definierten symbolischen Konstanten, ihre aktuellen Werte, ihre Entsprechungen in den symbolischen Aufbereitungsoperanden bei Angabe des Operanden MODE= und über ihre Zulässigkeit in den Mode-Arten.

## a) Ausgabe-Aufbereitungsbyte 1 (Output Edit Option Byte 1)

CUPAB-Name	Bit	entsprechender MODE-Operand	zulässig (X) bzw.obligatorisch (1) bei MODE=			
			COMP	LINE	FORM	PHYS
CWR1CODE	2 <sup>0</sup>	OTRSUP=	X	0	0	0
CWR1LNET	2 <sup>1</sup>	OLINEND=	X	0	1	1
	2 <sup>2</sup>	reserviert für MODE=	0	1	0	1
CWR1RSET	2 <sup>3</sup>	OMANUAL=	X	0	0	0
CWR1HOM	2 <sup>4</sup>	OHOM=	0	X	0	0
CWR1PTPE	2 <sup>5</sup>	OPTAPE=	X	0	0	0
	2 <sup>6</sup>	reserviert für MODE=	0	0	1	1
CWR1HARD	2 <sup>7</sup>	OHCOPY=	X	X	0	X

## b) Ausgabe-Aufbereitungsbyte 2 (Output Edit Option Byte 2)

CUPAB-Name	Bit	entsprechender MODE-Operand	zulässig (X) bzw. obligatorisch (1) bei MODE=			
			COMP	LINE	FORM	PHYS
CWR2HDR	2 <sup>0</sup>	OHDR=	X	0	1	X
CWR2NOLC	2 <sup>1</sup>	ONOLOGC=	0	X	0	0
CWR2EXT	2 <sup>2</sup>	EXTEND=	0	X	0	0
CWR2INFO	2 <sup>3</sup>	OINFO=	0	X	0	0
	2 <sup>4</sup>	reserviert	0	0	0	0
CWR2POSN	2 <sup>5</sup>	ONOPOSN=	0	X	0	0
CWR2TRAN	2 <sup>5</sup>	OTRANS=	0	0	0	X
CWR2BEL	2 <sup>6</sup>	OBELL=	0	X	0	0
CWR2ETB	2 <sup>7</sup>	OETB=	0	0	0	X

c) Eingabe-Aufbereitungsbyte 1 (Input Edit Option Byte 1)

CUPAB-Name	Bit	entsprechender MODE-Operand	zulässig (X) bzw.obligatorisch (1) bei MODE=			
			COMP	LINE	FORM	PHYS
CRD1CODE	2 <sup>0</sup>	ITRSUP=	X	0	0	X
CRD1LNET	2 <sup>1</sup>	ILINEND=	X	0	1	1
CRD1BACK	2 <sup>2</sup>	IGETBS=	X	X	X	X
CRD1RSET	2 <sup>3</sup>	IMANUAL=	X	0	0	0
CRD1LCT	2 <sup>4</sup>	ILCASE=	X	X	X	X
	2 <sup>5</sup>	reserviert für MODE=	0	1	0	1
	2 <sup>6</sup>	reserviert für MODE=	0	0	1	1
CRD1HDR	2 <sup>7</sup>	IHDR=	X	0	1	X

d) Eingabe-Aufbereitungsbyte 2 (Input Edit Option Byte 2)

CUPAB-Name	Bit	entsprechender MODE-Operand	zulässig (X) bzw. obligatorisch (1) bei MODE=			
			COMP	LINE	FORM	PHYS
CRD2GFC	2 <sup>0</sup>	IGETF=	0	X	0	0
	2 <sup>1</sup>	reserviert	0	0	0	0
CRD2CFD	2 <sup>2</sup>	ICFD=	0	X	0	0
CRD2GIC	2 <sup>3</sup>	IGETIC=	0	X	0	0
	2 <sup>4</sup>	reserviert	0	0	0	0
CRD2EXT	2 <sup>5</sup>	EXTEND=	0	X	0	0
	2 <sup>6</sup> und 2 <sup>7</sup>	} reserviert	0	0	0	0

Bei allen nicht reservierten Bits gilt für die Operanden:

MODE-Operand	zugehöriges Bit
= Y	gesetzt (1)
= N	rückgesetzt (0)

Die Bedeutung der MODE-Operanden (und damit der zugehörigen Bits der Eingabe- und Ausgabe-Aufbereitungsbytes) kann aus den Operandenbeschreibungen der Aufrufe RDATA, WROUT und WRTRD ersehen werden.

*Beispiel*

```

                                EXTERNAL SYMBOL DICTIONARY
BCUPAB  START
        CUPAB D
*
*           COMMUNICATIONS USER PARAMETER BLOCK
*
CUPAB   DSECT
*
*           RDATA PARAMETER BLOCK
*
CURAREAW DS   0A                WORD NAME FOR CURAREA
*
CUREDIT1 DC   AL1(0)           EDIT OPTION 1
CURAREA  DC   AL3(0)           READ AREA ADDRESS
*
CURFTB   DC   AL1(0)           FLAG TABLE BYTE
CUREDIT2 DC   AL1(0)           EDIT OPTION 2
CURALEN  DC   H'0'             READ AREA LENGTH
*
CURERRW  DS   0A                WORD NAME FOR CURERROR
*
CURACI   DC   AL1(0)           ASSIGNMENT CHANGE INDICATOR
CURERROR DC   AL3(0)           ERROR RETURN ADDRESS
*
L@RDATA EQU  *-CURAREAW        RDATA PARAMETER BLOCK LENGTH
*
*           WROUT PARAMETER BLOCK
*
*           ORG   CURAREAW
*
CUWMSGW  DS   0A                WORD NAME FOR CUWMSG
*
CUWEDIT1 DC   AL1(0)           EDIT OPTION 1
CUWMSG   DC   AL3(0)           OUTPUT MESSAGE ADDRESS
*
CUWERRW  DS   0A                WORD NAME FOR CUWERROR
*
CUWEDIT2 DC   AL1(0)           EDIT OPTION 2
CUWERROR DC   AL3(0)           ERROR RETURN ADDRESS
*
L@WROUT EQU *-CUWMSGW          WROUT PARAMETER BLOCK LENGTH
*
*           WRTRD PARAMETER BLOCK
*
*           ORG   CURAREAW
*
CUBMSGW  DS   0A                WORD NAME FOR CUBMSG
*
CUBOEDT1 DC   AL1(0)           OUTPUT EDIT OPTION 1
CUBMSG   DC   AL3(0)           OUTPUT MESSAGE ADDRESS
*
CUBAREAW DS   0A                WORD NAME FOR CUBAREA
*
CUBIEDT1 DC   AL1(0)           INPUT EDIT OPTION 1
CUBAREA  DC   AL3(0)           READ AREA ADDRESS
*
CUBOEDT2 DC   AL1(0)           OUTPUT EDIT OPTION 2
CUBIEDT2 DC   AL1(0)           INPUT EDIT OPTION 2

```

```

CUBALEN DC H'0' READ AREA LENGTH
*
CUBERRW DS 0A WORD NAME FOR CUBERROR
*
CUBERROR DC AL1(0) ERROR RETURN ADDRESS
L@WRTRDB EQU *-CUBMSGW WRTRD PARAMETER BLOCK LENGTH
*
*
* OUTPUT EDIT OPTION BYTE 1 - NAMES FOR BIT SETTINGS
*
CWR1CODE EQU 1 OTRSUP : CODE TRANSLATION
CWR1LNET EQU 2 OLINEND: LINE END TREATMENT
CWR1MBT1 EQU 4 MODE BIT 1
CWR1RSET EQU 8 OMANUAL: RESET FROM PAPER TAPE
CWR1HOM EQU 16 OHOM : HOMOGENEOUS OUTPUT
CWR1PTPE EQU 32 OPTAPE : PAPER TAPE CONTROL
CWR1MBT2 EQU 64 MODE BIT 2
CWR1HARD EQU 128 OHCOPY : HARDCOPY
CWR1MMSK EQU 68 MODE= MODE MASK
CWR1COMP EQU 0 COMP : COMPATIBLE MODE
CWR1LINE EQU 4 LINE : LINE MODE
CWR1FORM EQU 64 FORM : FORMAT MODE
CWR1PHYS EQU 68 PHYS : PHYSICAL MODE
*
* OUTPUT EDIT OPTION BYTE 2 - NAMES FOR BIT SETTINGS
*
CWR2HDR EQU 1 OHDR : HEADER PRESENT
CWR2NOLC EQU 2 ONOLOGC: NO LOGIC CHARS INTERPRET.
CWR2EXT EQU 4 EXTEND : EXTENDED LINE OUTPUT
CWR2INFO EQU 8 OINFO : INFORMATIVE MESSAGE
CWR2POSN EQU 32 ONOPOSN: NO POSITIONING
CWR2TRAN EQU 32 OTRANS : TRANSPARENT MODE
CWR2BEL EQU 64 OBELL : ACCOUSTIC ALARM
CWR2ETB EQU 128 OETB : ETB INSTEAD OF ETX
*
* INPUT EDIT OPTION BYTE 1 - NAMES FOR BIT SETTINGS
*
CRD1CODE EQU 1 ITRSUP : CODE TRANSLATION
CRD1LNET EQU 2 ILINEND: LINE END TREATMENT
CRD1BACK EQU 4 IGETBS : BACKSPACE
CRD1RSET EQU 8 IMANUAL: RESET TO MANUAL CONTROL
CRD1LCT EQU 16 ILCASE : LOWER CASE TRANSLATION
CRD1MBT1 EQU 32 MODE BIT 1
CRD1MBT2 EQU 64 MODE BIT 2
CRD1HDR EQU 128 IHDR : HEADER REQUIRED
CRD1MMSK EQU 96 MODE= MODE MASK
CRD1COMP EQU 0 COMP : COMPATIBLE MODE
CRD1LINE EQU 32 LINE : LINE MODE
CRD1FORM EQU 64 FORM : FORMAT MODE
CRD1PHYS EQU 96 PHYS : PHYSICAL MODE
*
* INPUT EDIT OPTION BYTE 2 - NAMES FOR BIT SETTINGS
*
CRD2GFC EQU 1 IGETFC : GET FUNCTION CODE
CRD2CFD EQU 4 ICFD : CONFIDENTIAL INPUT DATA
CRD2GIC EQU 8 IGETIC : GET IDENTITY CARD
CRD2EXT EQU 32 EXTEND : EXTENDED LINE INPUT
*,CUPAB 002 910215 53121055
END

```

### 3.3 Daten lesen von SYSDTA: RDATA

Mit RDATA kann der nächste Datensatz von SYSDTA gelesen werden. Die Datei, die als SYSDTA erklärt ist, kann ein Diskettengerät, ein Element einer PLAM-Bibliothek, eine SDF-P-Variable, eine katalogisierte SAM- oder ISAM-Datei oder insbesondere die prozeßführende Datenstation sein. Der Satz (im Fall der Datenstation: die Nachricht) wird in einem Bereich des Anwendungsprogramms als Satz variabler Länge übertragen.

Der Makro CUPAB generiert eine DSECT für die Operandentabelle des RDATA für den Aufruf im 24-Bit-Adressierungsmodus.

Der Makroaufruf RDATA akzeptiert den MF-Operanden (siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen").

Der Operand PARMOD steuert die Makroauflösung. Damit kann entweder die 24-Bit-Schnittstelle oder die 31-Bit-Schnittstelle generiert werden.

Für den 31-Bit-Adressierungsmodus ist folgendes zu beachten:

- Am Anfang der Operandenliste wird ein Standardheader (siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen") generiert. Der Anwender muß dafür sorgen, daß dieser Header korrekt initialisiert ist. Bei Verwendung von dynamisch erzeugten Operandenlisten (Zugriff über CSECT/DSECT) werden für den Standardheader keine symbolischen Namen und Equates erzeugt. Deshalb müssen bei dynamischer Versorgung der Operandenliste die Initialisierungswerte für den Standardheader aus einer mit MF=L erzeugten Operandenliste übernommen werden.
- Eine CSECT/DSECT wird mit MF=C/D bzw. MF=(C,p)/(D,p) generiert.  
p = Präfix (max. 3 Zeichen, ein längeres Präfix wird auf drei Zeichen gekürzt);  
Voreinstellung p = CUR. Das Präfix verändert nur die Feldnamen, nicht die symbolischen Namen bei den Equates.

#### *Hinweis zur Kompatibilität*

Der Makroaufruf RDATA ab der TIAM-Version V10.0A (gekoppelt an BS2000 V10.0A) erzeugt die Headerversion 2 im Standardheader. Damit ist er in BS2000-Versionen kleiner V10.0 nicht ablauffähig und verzweigt dort auf den Fehlerausgang. Soll ein Programm in mehreren BS2000-Versionen ausführbar sein, so muß das Programm in der niedrigsten Version erstellt sein (es wird nur Aufwärtskompatibilität garantiert).



**Format 1**

Operation	Operanden
RDATA	<p>satz, fehler[,länge][,edit][,A]</p> $\left. \begin{array}{l} 24[,MF=\left\{ \begin{array}{l} L \\ (E,\dots) \end{array} \right\}] \\ \\ [,PARMOD=\left\{ \begin{array}{l} C \\ (C,p) \\ D \\ (D,p) \\ L \\ (E,\dots) \end{array} \right\}] \\ \\ 31[,MF=\left\{ \begin{array}{l} C \\ (C,p) \\ D \\ (D,p) \\ L \\ (E,\dots) \end{array} \right\}] \end{array} \right\}$ <p>[,KEYOUT=<math>\left\{ \begin{array}{l} N \\ Y \end{array} \right\}</math>]</p> <p>[,KEYPOS=<math>\left\{ \begin{array}{l} N \\ Y \end{array} \right\}</math>] [,KEYLEN=<math>\left\{ \begin{array}{l} N \\ Y \end{array} \right\}</math>]</p> $\left[ \begin{array}{l} ,MODE=COMP [,ITRSUP=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] [,ILINEND=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] \\ \\ [,ILCASE=\left\{ \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\}] [,IHDR=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] \\ \\ [,IGETBS=\left\{ \begin{array}{l} Y[ES] \\ N[O] \end{array} \right\}] \\ \\ ,MODE=LINE [,ILCASE=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] [,IGETBS=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] \\ \\ [,IGETFC=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] [,IGETIC=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] \\ \\ [,ICFD=\left\{ \begin{array}{l} N[O] \\ Y[ES] \end{array} \right\}] \end{array} \right]$ <p>[,RC=<math>\left\{ \begin{array}{l} OLD \\ NEW \end{array} \right\}</math>]</p> <p>[,VTSUCBA=addr]</p> <p>[,TIMER=wert]</p>

Die Operanden edit, ITRSUP, ILINEND, MODE, RC, VTSUCBA und TIMER werden nur ausgewertet, wenn SYSDATA der Datenstation zugeordnet ist. Die Operanden KEYOUT und KEYPOS werden nur ausgewertet, wenn SYSDATA einer ISAM-Datei zugeordnet ist.

Die Operanden RC, VTSUCBA und TIMER werden nur bei der 31-Bit-Schnittstelle unterstützt.

satz                      Symbolische Adresse des Feldes, in das der einzulesende Datensatz übertragen wird. Das Feld beginnt mit dem Satzlängenfeld.

Satzaufbau:

Byte	0-1:	Länge des Satzes + 4 byte Satzlängenfeld
	2-3:	reserviert
	4-n:	Datensatz

*Beispiel*

SATZ	DS	0CL74
LAENGE	DS	CL2
RESERV	DS	CL2
DATEN	DS	CL70

fehler                      Symbolische Adresse im Anwendungsprogramm, zu der verzweigt wird:

- im Fehlerfall
- wenn der Operand A gesetzt ist

Im Fehlerfall enthält Register R14 die Adresse des Befehls, der dem RDATA-Aufruf folgt. Der Fehlercode wird im Register R15 übergeben.

31-Bit-Schnittstelle: Bei Angabe fehler = 0 (Adresse X'00..0') wird das Programm mit dem Befehl fortgesetzt, der dem RDATA-Aufruf folgt.

länge                      Größe des Einlesebereichs einschließlich 4 byte für das Satzlängenfeld. Die maximal zulässige Länge beträgt 32767 byte. Fehlt der Operand, wird das Längenattribut des Einlesebereichs angenommen.

edit                      Aufbereitungsfunktion (Edit-Options) für eine Eingabe-Nachricht von der Datenstation. Dieser Operand ist nicht erforderlich, wenn Standardfunktionen (alle Edit-Bits=0) verwendet werden, bei einer MODE-Angabe oder bei Nutzung des VTSU-Control-Blocks. Durch Direktangabe (X'xx') kann nur das 1. Edit-Byte für Eingabe auf die beim CUPAB-Makro beschriebene Bedeutung gesetzt werden.

*Hinweis*

Dieser Parameter wird nur noch aus Gründen der Kompatibilität unterstützt.

Die Edit-Optionen sollten über MODE-Angaben (siehe unten) oder über den VTSU-Control-Block (VTSUCBA) gesteuert werden.

A	Das Anwendungsprogramm wird von der anfänglichen Standardzuweisung und von jeder folgenden Zuweisung für SYSDTA verständigt. Verständigt wird über die Fehleradresse, sobald der Lesevorgang beendet ist. Der Zuweisungsschlüssel wird im 31-Bit-Modus im Feld CURAIND der RDATA-Operandenliste zurückgegeben, im 24-Bit-Modus im linksbündigen Byte im Register R15.
PARMOD	Steuert die Makroauflösung. Es wird entweder die 24-Bit-Schnittstelle oder die 31-Bit-Schnittstelle generiert. Wenn PARMOD nicht angegeben wird, erfolgt die Makroauflösung entsprechend der Angabe für den Makro GPARMOD (siehe Handbuch "Makroaufrufe an den Ablaufteil" [3]) oder der Voreinstellung für Assembler (= 24-Bit-Adressierung).
=24	Die 24-Bit-Schnittstelle wird generiert. Datenlisten und Befehle benutzen 24-Bit-Adressen (Adreßraum ≤ 16MB).
=31	Die 31-Bit-Schnittstelle wird generiert. Datenlisten und Befehle benutzen 31-Bit-Adressen (Adreßraum ≤ 2GB). Datenlisten beginnen mit dem Standardheader.
MF	
=L	Generiert eine Parameterliste, Felder werden gemäß Parameterangaben versorgt.
=(E,...)	Generiert den Befehlsteil.
=C	Generiert eine CSECT.
=(C,p)	Generiert eine CSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = CUR.
=D	Generiert eine DSECT.
=(D,p)	Generiert eine DSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = CUR.

KEYOUT	Dieser Operand wird nur ausgewertet, wenn SYSDTA einer ISAM-Datei zugeordnet ist.
=Y	Vor der Übergabe des Satzes an das Anwendungsprogramm wird der ISAM-Schlüssel entfernt (siehe Hinweis).
=N	Der ISAM-Schlüssel wird nicht entfernt.
KEYPOS	Dieser Operand wird nur ausgewertet, wenn SYSDTA einer ISAM-Datei zugeordnet ist.
=Y	24-Bit-Modus: Die Position des ISAM-Schlüssels (minus 1) wird im Register 0 wie folgt abgespeichert: <ul style="list-style-type: none"> <li>– in den beiden mittleren Byte, wenn KEYLEN=Y, bzw.</li> <li>– in den beiden rechtsbündigen Byte, wenn KEYLEN=N ist.</li> </ul> 31-Bit-Modus: Die Position des ISAM-Schlüssels (minus 1) wird im Feld CURKEYP (RDATA-Operandenliste) abgespeichert.
=N	Die Position des ISAM-Schlüssels wird nicht abgespeichert.
KEYLEN	Dieser Operand wird nur ausgewertet, wenn SYSDTA einer ISAM-Datei zugeordnet ist.
=Y	24-Bit-Modus: Die Länge des ISAM-Schlüssels (minus 1) wird im rechtsbündigen Byte des Registers 0 gespeichert (siehe Hinweis). 31-Bit-Modus: Die Länge des ISAM-Schlüssels (minus 1) wird im Feld CURKEYL (RDATA-Operandenliste) abgespeichert.
=N	Die Länge des ISAM-Schlüssels wird nicht abgespeichert.

*Hinweis*

Dieselbe Bedeutung wie 'Y' hat auch die Angabe einer Zeichenfolge, die mit 'Y' beginnt. Jedes von 'Y' verschiedene Zeichen wird wie die Angabe von 'N' interpretiert; auch jede Zeichenfolge, die nicht mit 'Y' beginnt, wirkt wie die Angabe von 'N' und verursacht eine MNOTE-Meldung.

Folgende Operanden werden nur ausgewertet, wenn SYSDTA der Datensichtstation zugeordnet ist.

Die MODE-Angaben zusammen mit den Edit-Optionen werden nur noch aus Kompatibilitätsgründen unterstützt. Sie werden jetzt im VTSU-Control-Block (VTSUCB) zusammengefaßt. Künftige Erweiterungen erfolgen nur noch im VTSUCB.

## MODE

**=COMP** Kompatibler Betriebsmodus. Vom Anwendungsprogramm können über symbolische Operanden sämtliche Edit-Options entsprechend der Edit-Option-Tabelle (siehe Makro CUPAB) verwendet werden. Eventuell im Operanden "edit" gemachte Angaben werden ignoriert. Steuerzeichen im Text werden unbesehen dem Anwendungsprogramm übergeben. Diese Betriebsweise ist zu früheren Versionen kompatibel (kompatibel unterstützte Datenstationen: 8110, 8151 und 8152). Für die übrigen Datenstationen wird dieser Modus wie MODE=LINE behandelt. Alle Edit-Options, außer ILCASE werden abgewiesen (RC: X'08').

**=LINE** Ist SYSDTA eine Datenstation, so wird diese als logische Zeilendatenstation behandelt. Die Nachricht kann mit logischen Steuerzeichen strukturiert sein (siehe Makro VTCSET). Der gerätespezifische Nachrichten-Vorspann wird nicht mitgeliefert. Die Operanden für die Nachrichtenaufbereitung werden ausgewertet. Ist SYSDTA eine katalogisierte Datei oder ein Diskettengerät - also keine Datenstation - so werden die Operanden zur Nachrichtenaufbereitung nicht ausgewertet. Zum Beispiel werden Kleinbuchstaben nicht in Großbuchstaben übersetzt, wenn SYSDTA eine katalogisierte Datei ist und ILCASE den Wert NO (Standardwert) hat. Logische Steuerzeichen in der Nachricht werden ebenfalls nicht ausgewertet.

## ITRSUP

**=YES** Die Übersetzung von Gerätecode in EBCDIC wird unterdrückt. Das Anwendungsprogramm erhält also die Nachricht im Gerätecode.

**=NO** Die Übersetzung von Gerätecode in EBCDIC wird nicht unterdrückt. Das Anwendungsprogramm erhält die Nachricht im EBCDIC-Code.

## ILINEND

**=YES** Die Zeichen Wagenrücklauf/Zeilenvorschub werden in das Anwendungsprogramm übertragen.

**=NO** Die Wagenrücklauf-/Zeilenvorschubzeichen werden dem Anwendungsprogramm nicht übergeben.

## IGETBS

=YES

Die Zeichen 'Underline'(X'6D') werden dem Anwendungsprogramm übergeben. Es findet keine Auswertung durch das System statt (nur für Schreibstationen).

=NO

'Underline' wird nicht in das Anwendungsprogramm übertragen. Statt dessen wird vom System die Korrekturfunktion durchgeführt (nur für Schreibstationen).

## ILCASE

=YES

Dem Anwendungsprogramm werden auch Kleinbuchstaben übergeben.

=NO

Alle Kleinbuchstaben werden dem Anwendungsprogramm als Großbuchstaben übergeben.

## IHDR

=YES

Der gesamte Nachrichtenvorspann wird an das Anwendungsprogramm übergeben.

=NO

Der Nachrichtenvorspann wird nicht an das Anwendungsprogramm übergeben.

## IGETFC

=YES

Das 5. Byte des Einlesebereichs soll den normierten Funktionstastencode enthalten. Dieser identifiziert die zur Auslösung der Datenübertragung an der Datenstation betätigte Taste. Eine Tabelle der normierten Funktionstastencodes finden Sie im Anhang (siehe Seite 201.)

=NO

Es soll kein Funktionstastencode übergeben werden.

## IGETIC

=YES

Die Eingabe soll vom angeschlossenen Ausweisleser erfolgen. Die Eingabedaten können nur aus der Ausweisinformation oder aus dem Kurztelegramm K14 bestehen. Diese Angabe ist nur bei den Datenstationen 8160, 9749, 975x, 9763 und 3270 mit einem definierten Ausweisleser möglich (siehe auch Makro TSTAT TYPE=TCHAR).

Der Operand IGETIC wird ignoriert, wenn gleichzeitig ICFD=YES angegeben wird. Diese Edit-Option wird ignoriert (IGETIC auf NO zurückgesetzt), wenn kein Ausweisleser angeschlossen ist.

=NO

Die Eingabequelle soll nicht verändert werden.

## ICFD

=YES

Die Eingabedaten sind vertraulich und sollen an der Datenstation unsichtbar bleiben. Dies erfolgt je nach Datenstation durch Dunkelsteuerung bzw. Löschen des Bildschirms oder durch Überschreiben der Eingabezeile bei Schreibstationen. Beim Löschen des Bildschirms wird das Bildschirmformat auf 24x80 zurückgesetzt.

=NO

Es sollen keine Vorkehrungen zum Schutz vertraulicher Daten getroffen werden.

## RC

=NEW

Dieser Operand ist nur für die 31-Bit Schnittstelle zulässig. Der Returncode wird in Register 15 und im Standardheader (mit vollem 4-Byte-Returncode) abgelegt. Ein 4-Byte-Returncode wird nur zurückgeliefert, wenn SYSDTA von der Datensichtstation liest. In allen anderen Fällen wird nur ein 1-Byte-Returncode zurückgeliefert, unabhängig vom Wert des Returncodes.

=OLD

Der Returncode wird in Register 15 abgelegt. Die drei linksbündigen Byte werden ausgeblendet.

## VTSUCBA

=addr

Adresse eines mit MF=L erzeugten VTSUCB. Dieser Operand ist nur für die 31-Bit Schnittstelle zulässig. Bei Benutzung des Operanden VTSUCBA wird der Operand MODE und die folgenden EDIT-Optionen ignoriert. In der Parameterliste wird ihr Wert auf X'FF' gesetzt. Alle gewünschten EDIT-Optionen müssen im VTSUCB angegeben werden.

Standardwert: keine Verwendung des VTSUCB

## TIMER

=wert

Dieser Operand ist nur für die 31-Bit Schnittstelle zulässig.

In diesem Feld kann der Benutzer angeben, ob ein TIMER benutzt werden soll oder nicht (UNLIMITED). Der Timer legt die maximale Wartezeit für die Eingabe fest. Erfolgt innerhalb der festgelegten Wartezeit keine Eingabe, wird ein Rückgabewert zurückgegeben. Der Wert wird in Sekunden angegeben.

Standardwert: UNLIMITED (es wird kein Timer benutzt).

## Format 2

Operation	Operanden
RDATA	(1)

- (1) Register R1 enthält die Adresse der Operandenliste. Die Liste ist auf Wortgrenze ausgerichtet.

## Funktionsweise

Beim Ablauf des Makros werden im Fall von Format 1 die spezifizierten Operanden in einer Operandentabelle abgespeichert und die Anfangsadresse dieser Tabelle in Register 1 geladen. Im Fall von Format 2 wird die im Anwendungsprogramm spezifizierte Tabelle verwendet.



## Aufbau der Operandenliste

Adressierungsmodus	Byte	Inhalt
24-Bit-Modus	0	Input Edit Byte 1, gemäß Edit-Option-Tabelle für Eingabe.
	1 - 3	Adresse des Feldes, in das der eingelesene Datensatz übertragen wird (Operand einlesebereich).
	4	Flag, Bit $2^7=1/0$ entspricht KEYOUT=Y/N, Bit $2^6=1/0$ entspricht KEYPOS=Y/N Bit $2^5=1/0$ entspricht KEYLEN=Y/N
	5	Input Edit Byte 2, gemäß Edit-Option-Tabelle für Eingabe.
	6 - 7	max. Länge des einzulesenden Datensatzes (Operand länge).
	8	SYSDTA-Zuweisungsindikator, Bit $2^0=1$ entspricht Operand A.
	9 - 11	Adresse, zu der im Fehlerfall verzweigt werden soll (Operand fehler).
	31-Bit-Modus	0 - 7
8 - 11		Adresse, zu der im Fehlerfall verzweigt wird (Operand fehler).
12 - 15		Adresse des Feldes, in das der eingelesene Datensatz übertragen wird (Operand einlesebereich).
16		Input Edit Byte 1
17		Input Edit Byte 2
18		Zuweisungsschlüssel für SYSDTA; Equates im Makro erzeugt mit MF=D.
19		Flag das die Verwendung des VTSUCB und das Returncode-Verhalten anzeigt; Equates im Makro erzeugt mit MF=D.
20 - 21		max. Länge des einzulesenden Datensatzes (Operand länge).
22		Flag Aufbereitungsbyte des ISAM-Schlüssels
23		SYSDTA-Zuweisungsindikator (Operand A)
24 - 25		Position des ISAM-Schlüssels.
26 - 27		Länge des ISAM-Schlüssels.
28 - 31		Adresse des VTSUCB
32 - 33		Werte des Timers
34 - 35		reserviert (X' 0000' )

Die Werte für Input Edit Byte 1/2 sind der beim Makro CUPAB angegebenen Tabelle (24-Bit-Modus) oder einer mit MF=C/D erzeugten Datenliste (31-Bit-Modus) zu entnehmen.

#### *Hinweis*

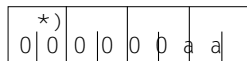
Wird während der Lese-Operation ein "BREAK" erkannt, so wird der P1-Befehlszähler auf den Anfang der Makroauflösung zurückgesetzt, so daß nach der Behandlung der Unterbrechung der Makroaufruf wiederholt wird.

### Rückinformation (Quittungen) und Fehleranzeigen

- Für ein Gerät oder für den Betriebsmodus MODE= nicht zugelassene Edit-Options werden vom System soweit wie möglich korrigiert.
- Während der Makrobearbeitung enthält Register R1 die Adresse der Operandenliste.
- Über die Ausführung des Makros RDATA wird im Register R15 ein Returncode übergeben.

a) 24-Bit-Modus:

Register R15:



\*) ggf. Zuweisungsschlüssel bei Angabe des Operanden A.

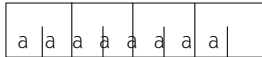
X' aa'	Erläuterung
X' 00'	Normale Beendigung
X' 04'	Nicht behebbarer Fehler
X' 08'	Operandenfehler
X' 0C'	Abschneiden des Satzes, Satzlänge $\geq$ spezifizierte Länge
X' 10'	Dateiende (/EOF)
X' 14'	SYSDTA ist nicht zugewiesen
X' 18'	Fehler während des Datenträger-Zugriffs
X' 20'	Ungültiges Edit-Option-Byte; Fehler wurde vom System korrigiert
X' 38'	Problem im Zusammenhang mit POSIX

## b) 31-Bit-Modus:

Bei Verwendung der 31-Bit-Schnittstelle kann man die Returncodes in Register 15 (RC=OLD) oder in Register 15 und zusätzlich im Standardheader ablegen (RC=NEW).

**RC=OLD**

Register R15:



Zusätzlich, zu den unter a) beschriebenen Returncodes (rechtsbündig, links 0), können der Returncode X'24' (Fehler im VTSUCB), X'34' (Timer-Ablauf) sowie die systemweit festgelegten Werte auftreten.

**RC=NEW**

Fehlermeldungen werden in Register 15 und in den Standard-Header eingetragen, siehe folgende Tabelle (Equates für MAINCODE und SUBCODE siehe Makroauflösung).

SUBCODE		MAINCODE		Bedeutung
2	1	2	1	
X' 00'	X' 00'	X' 00'	X' 00'	erfolgreiche Bearbeitung
			X' 20'	– keine Zusatzinformation
			X' 20'	– Operandenfehler, korrigiert durch TIAM/VTSU
X' 00'	X' 00'	X' 00'	X' 14'	SYSFILE-Fehler
			X' 18'	– SYSDTA nicht zugewiesen
			X' 18'	– Fehler während des Datenträger-Zugriffs
X' 00'	X' 01'	X' 00'	X' 08'	Parameterfehler
X' 07'			X' 08'	– Operandenfehler, nicht näher spezifiziert
X' 08'			X' 08'	– RESERVED-Felder nicht 0
			X' 08'	– der Timerwert ist nicht im erlaubten Bereich von 10 bis 3600 Sekunden
X' 00'	X' 20'	X' 00'	X' 04'	interner Fehler
X' 02'			X' 04'	– nicht näher spezifiziert
X' 05'			X' 04'	– BCAM-Nachricht verloren
X' 06'			X' 04'	– Eingabenachricht zu lang
			X' 04'	– negative Transportquittung

SUBCODE		MAINCODE		Bedeutung
2	1	2	1	
X' 00'	X' 40'	X' 00'	X' 0C' X' 10' X' 34'	keine Klassenreaktion – Abschneiden des Eingabesatzes – Dateiende – Timer-Ablauf (innerhalb der festgelegten Wartezeit erfolgte keine Eingabe)
X' 01'	X' 80'	X' 00'	X' 04'	interner Engpaß – BCAM-Engpaß
X' 09' X' 0A'	X' 80' X' 40'	X' 00' X' 00'	X' 38' X' 38'	Fehler im Zusammenhang mit POSIX – Ein/Ausgabe-Serialisierungsfehler – Wenn die LOGON-Task im System-Modus ist, sind keine Ein/Ausgaben von mit fork() erzeugten Prozessen möglich
aus VTSUCB	aus VTSUCB	aus VTSUCB	X' 24'	VTSU hat Fehler erkannt (für MAINCODE 1 siehe Fehlerinformation im VTSUCB-Header, beschrieben im VTSU-Handbuch [1])

### Hinweise

- Ein Satz wird abgeschnitten, wenn der zu übertragende Satz größer ist als im Längenoperanden angegeben. Der Satz wird nur entsprechend dem Längenoperanden in den angegebenen Einlesebereich übertragen. Der Satzrest geht verloren. Ist der Satz kürzer als der Einlesebereich, wird er in den Einlesebereich linksbündig eingetragen. Der verbleibende Rest des Einlesebereichs wird nicht mit Leerzeichen aufgefüllt. Das Programm wird ohne Fehleranzeige weiter ausgeführt.
- Bei Disketten gilt:  
Wenn SYSDTA einem Diskettengerät des Typs FD3170 zugewiesen ist, dürfen die Sätze, die vom RDATA-Makro gelesen werden, nicht länger als 128 byte sein.  
Wenn SYSDTA einem Diskettengerät des Typs FD3171 zugewiesen ist, dürfen die Sätze, die vom RDATA-Makro gelesen werden, nicht länger als 1024 bzw. 2048 byte sein.
- Eine Bedingung für Dateiende kann auf folgende Weise eintreten:  
Datenstationsbetrieb:  
Durch Auslösen der Funktion ESCAPE (Übergang vom Programmmodus in den Systemmodus) und Absenden des Kommandos EOF mit anschließendem Kommando /RESUME.

Andere Eingabemedien:

- a) wenn die Dateien SYSDTA und SYSCMD die gleichen sind, und ein Kommando (nicht BREAK) gelesen wird.

*Hinweis*

Als Kommando gilt ein Satz, der mit '/' beginnt; der '/' darf nicht durch symbolische Operanden ersetzt werden.

- b) wenn die Dateien SYSDTA und SYSCMD nicht gleich sind, und /EOF in den Spalten 1 - 4 des Satzes erkannt wird.
- Die Anzeige "SYSDTA nicht zugewiesen" erscheint, wenn alle Sätze von der Diskette eingelesen wurden und erneut ein RDATA-Makro abgesetzt wird.
  - Die Anzeige "Fehler während des Datenträgerzugriffs" erscheint, wenn SYSDTA einem Diskettengerät zugewiesen wird, und die gewünschte Diskette ist nicht zugreifbar.
  - Zuordnung von SYSDTA
    - a) 24-Bit-Modus:  
Im linksbündigen Byte des Registers R15 wird eine Anzeige über die Zuordnung von SYSDTA übergeben.

Register R15:



- b) 31-Bit-Modus:  
Der SYSDTA-Zuweisungsschlüssel wird im Feld CURAIND der RDATA Parameterliste zurückgeliefert. Die RDATA-CSECT/DSECT enthält dafür entsprechende Equates.

### Zuweisungsschlüssel

X' bb'	Erläuterung
X' 00'	Keine Änderung von SYSDTA
X' 01'	SYSDTA ist ein Diskettengerät
X' 04'	SYSDTA ist eine sequentielle Datei
X' 08'	SYSDTA ist eine indiziert-sequentielle Datei
X' 14'	SYSDTA ist eine Datenstation
X' 18'	SYSDTA ist eine SDF-P-Variable
X' 20'	SYSDTA ist ein Element einer PLAM-Bibliothek

## Makroauflösung

```

RDATA MF=C,PARMOD=31
*
*          RDATA PARAMETER BLOCK
*
RDATA FHDR MF=(C,*NO-NAMES)
*
RDATA DS   A           0  INTERFACE IDENTIFIER
      DS   A           4  GENERAL RETURN CODE
*
* MAIN RETURN CODES AS IN FHDR, ADDITIONAL VALUES:
CURURE EQU 4           UNRECOVERABLE ERROR
CUROPE EQU 8           OPERAND ERROR
CURCUTE EQU 12         MESSAGE IS CUT
CUREOF EQU 16          END OF FILE
CURNDTA EQU 20         NO SYSDTA 1
CURACCE EQU 24         ACCESS ERROR
CUROPEC EQU 32         OPERAND ERROR, CORRECTED
CURVTE  EQU 36         VTSU ERROR 1
CURTOUT EQU 52         TIME OUT DURING READ
CURPOSE EQU 56         ERROR IN POSIX CONTEXT
* SUB1 RETURN CODE AS IN FHDR
* SUB2 RETURN CODES:
CURRCSN EQU 0          RC_S2_NOT_SPECIFIED
CURBCSH EQU 1          BCAM SHORTAGE
CURBCLL EQU 2          BCAM LETTER LOST
CURMTL  EQU 5          MESSAGE TOO LONG
CURNACK EQU 6          NEGATIVE TACK 1
CURRFNZ EQU 7          RESERVED FIELDS NOT ZERO
CURWT   EQU 8          WRONG TIMER
CURLOCE EQU 9          SERIALIZATION LOCK ERROR
CURFTIO EQU 10         NO FORKED TASK IOS POSSIBLE.
*
CURERRW DC   A(0)      ERROR RETURN ADDRESS
CURAREAW DC   A(0)      READ AREA ADDRESS
CUREDIT1 DC  AL1(0)    EDIT OPTION 1
CUREDIT2 DC  AL1(0)    EDIT OPTION 2
CURAIND DC   AL1(0)    SYSDTA ASSIGNMEN
CURANO  EQU  X'00'     - NO CHANGE
CURADISC EQU X'01'     - FLOPPY DISC
CURASAM EQU X'04'     - SAM FILE
CURAISAM EQU X'08'     - ISAM FILE
CURACARD EQU X'0C'     - CARD READER (OBSOLETE)
CURADSS EQU X'14'     - TERMINAL
CURAVAR EQU X'18'     - SDF-P VARIABLE
CURAPLAM EQU X'20'     - PLAM LIBRARY
CURFB1  DC   AL1(0)    FLAG BYTE 1
CURVTSI EQU X'80'     - VTSU INDICATOR
CURRCIND EQU X'40'     - RC INDICATOR
CURALEN DC   H'0'     READ AREA LENGTH
CURFTB  DC   AL1(0)    FLAG TABLE BYTE
CURACI  DC   AL1(0)    ASSIGNMENT CHANGE INDICATOR
CURKEYP DC   H'0'     KEY-POSITION
CURKEYL DC   H'0'     KEY-LENGTH
CURVTSU DC   A(0)     VTSUCB ADDRESS
CURTVAL DC   H'0'     INPUT TIMER VALUE
CURRTIA DC   H'0'     RES_FOR_TIAM
*

```

```

CURL RDA EQU *--RDATAP          RDATA PARAMETER BLOCK LENGTH
*
*          INPUT EDIT OPTION BYTE 1 - NAMES FOR BIT SETTINGS
*
CRD1CODE EQU 1          ITRSUP : CODE TRANSLATION
CRD1LNET EQU 2          ILINEND: LINE END TREATMENT
CRD1BACK EQU 4          IGETBS : BACKSPACE
CRD1RSET EQU 8          IMANUAL: RESET TO MANUAL CONTROL
CRD1LCT  EQU 16         ILCASE : LOWER CASE TRANSLATION
CRD1MBT1 EQU 32         MODE BIT 1
CRD1MBT2 EQU 64         MODE BIT 2
CRD1HDR  EQU 128        IHDR   : HEADER REQUIRED
CRD1MMSK EQU 96         MODE=  : MODE MASK
CRD1COMP EQU 0          COMP   : COMPATIBLE MODE
CRD1LINE EQU 32         LINE   : LINE MODE
CRD1FORM EQU 64         FORM   : FORMAT MODE
CRD1PHYS EQU 96         PHYS   : PHYSICAL MODE
*
*          INPUT EDIT OPTION BYTE 2 - NAMES FOR BIT SETTINGS
*
CRD2GFC  EQU 1          IGETFC : GET FUNCTION CODE
CRD2CFD  EQU 4          ICFD   : CONFIDENTIAL INPUT DATA
CRD2GIC  EQU 8          IGETIC : GET IDENTITY CARD
CRD2EXT  EQU 32         EXTEND  : EXTENDED LINE INPUT

```

### Beispiel 1a

In diesem Beispiel wird das Format 1 ohne den VTSUCB verwendet.

```

RDATAT  START 0
*
          BALR 10,0
          USING *,10
*
RDATA INPUT1,FEHL,40,PARMOD=31,MODE=LINE,ILCASE=Y,ICFD=Y
*
FEHL    NOP    0
          TERM
*
INPUT1  DS     OCL40
LAENGE  DS     OL2
UNUSED  DS     CL2
DATEN   DS     CL36
*
          END

```



*Beispiel 1b*

In diesem Beispiel wird das Format 1 mit dem VTSUCB verwendet.

```

RDATA1  START  0
*
          BALR  10,0
          USING *,10
*
          RDATA INPUT1,FEHL,40,PARMOD=31,VTSUCBA=VTSUPAR
*
FEHL     NOP    0
          TERM
*
VTSUPAR VTSUCB MODE=LINE,LOW=YES,SPECIN=C
*
INPUT1  DS     OCL40
LAENGE  DS     CL2
UNUSED  DS     CL2
DATEN   DS     CL36
*
          END

```

*Beispiel 2*

Verwendung des Formats 2

```

RDATA2  START  0
*
          BALR  10,0
          USING *,10
*
          LA    1,PARAM
          RDATA (1),PARMOD=31
*
END      NOP    0
          TERM
*
          DS    0F
PARAM    RDATA INPUT,END,40,MF=L,PARMOD=31,MODE=LINE, +
          ILCASE=Y,ICFD=Y
*
INPUT    DS     OCL40
LENGTH  DS     L2
UNUSED  DS     CL2
DATEN   DS     CL36
*
          END

```

### 3.4 Datenstationseigenschaften ändern: TCHNG

Mit dem Makroaufruf TCHNG können Eigenschaften der logischen Datenstation durch das Anwendungsprogramm geändert werden. Die Wirkung des Aufrufs TCHNG bleibt bis zur Beendigung des betreffenden Anwenderprogramms oder bis zum nächsten TCHNG-Aufruf erhalten und gilt für die Aus- und Eingaben auf die Datenstation mit den Makros RDATA, WROUT und WRTRD. Alle nicht im TCHNG-Makro angegebenen Werte bleiben unverändert.

Operation	Operanden
TCHNG	$[ \text{OFLOW} = \left\{ \begin{array}{l} \text{SYS} \\ \text{USER} \end{array} \right\} ] [ \text{, EDOPT} = \underline{\text{DYN}} ]$ $[ \text{, EDOPT} = \text{STAT} } \left\{ \begin{array}{l} \text{, MODE} = \text{LINE} [ \text{, OHCOPY} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] [ \text{, OHOM} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] [ \text{, OINFO} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] \\ \\ [ \text{, ONOPOSN} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] [ \text{, OBELL} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] [ \text{, IGETBS} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] \\ \\ [ \text{, ILCASE} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] [ \text{, IGETFC} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] \\ \\ [ \text{, IGETIC} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] [ \text{, ICFD} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] \\ \\ \text{, MODE} = \text{FORM} [ \text{, IGETBS} = \left\{ \begin{array}{l} \text{N} \\ \text{Y} \end{array} \right\} ] [ \text{, ILCASE} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} ] \end{array} \right\} ]$ $[ \text{, SUB} = \left\{ \begin{array}{l} \underline{\text{OUT}} \\ \text{OUTIN} \end{array} \right\} ]$ $[ \text{, INFOLIN} = \left\{ \begin{array}{l} \underline{\text{N[O]}} \\ \text{Y[ES]} \end{array} \right\} ]$ $[ \text{, CLEAR} = \left\{ \begin{array}{l} \underline{\text{Y[ES]}} \\ \text{N[O]} \end{array} \right\} ]$

## OFLOW

=SYS

Bei langen Ausgaben des Anwendungsprogramms soll das System eine Überlaufkontrolle durchführen, um einen Informationsüberlauf an der Datenstation zu verhindern. Die Art der Systemüberlaufkontrolle bestimmt der Datenstationsbenutzer durch das Systemkommando MODIFY-TERMINAL-OPTIONS (Voreinstellung und Wirkung siehe dort).

=USER

Das System soll keine Vorkehrungen treffen, einen Informationsüberlauf bei langen Ausgaben des Anwendungsprogramms zu verhindern. Eine Überlaufkontrolle kann so individuell vom Anwendungsprogramm durchgeführt werden.

*Hinweis*

Meldungen des Betriebssystems unterliegen nach wie vor der vom System oder mittels MODIFY-TERMINAL-OPTIONS-Kommando festgelegten Überlaufkontrolle

## EDOPT

=DYN

Bei nachfolgenden Aufrufen der Makros RDATA, WROUT und WRTRD sollen die dort angegebenen Edit-Options-Werte ausgewertet werden (Voreinstellung durch das System).

=STAT

Dieser Parameter wird bei Verwendung des VTSUCB ignoriert.

Bei nachfolgenden Aufrufen der Makros RDATA, WROUT und WRTRD sollen die im TCHNG-Makro angegebenen Edit-Options-Werte für MODE, OBELL, OHCOPY, OHOM, OINFO, ONOPOSN, IGETBS, ILCASE, IGETFC, IGETIC und ICFD verwendet werden. Diese Festlegung gilt bis zu einer erneuten Änderung durch TCHNG oder bis Programmende. Wird ein Programmablauf unterbrochen (mit K2) wirken die angegebenen Edit-Options-Werte auch auf alle Ausgaben die während dieser Unterbrechung über WROUT und WRTRD erfolgen (einige Systemkomponenten verwenden WROUT).

MODE=  
OHCOPY=  
OHOM=  
OBELL=  
ONOPSN=  
OINFO=  
IGETBS=  
ILCASE=  
IGETFC=  
IGETIC=  
ICFD=

Zur Bedeutung dieser Operanden siehe Makro WRTRD bzw. WROUT. Fehlt ein Operand, wird jeweils der unterstrichene Wert angenommen.

## SUB

=OUT

Das definierte Ersatzzeichen wird bei LINE-MODE-Eingaben an das Anwendungsprogramm durchgereicht.

=OUTIN

Das definierte Ersatzzeichen wird bei LINE-MODE-Eingaben als logisches Steuerzeichen SUB an das Anwendungsprogramm übergeben.

## INFOLIN

=YES

An Datenstationen ohne hardwaremäßige Systemzeile sollen ankommende Nachrichten für eine spezielle Informationszeile (OINFO=YES bzw. MODE=INFO) in der letzten Bildschirmzeile abgebildet werden, wenn am Bildschirm gerade eine formatierte oder physikalische Nachricht abgebildet ist (siehe Makroaufruf WROUT).

=NO

Ankommende Nachrichten für eine spezielle Informationszeile sollen an Datenstationen ohne Systemzeile wie normale Line-Mode-Nachrichten abgebildet werden.

## CLEAR

=YES

Beim Wechsel des Ausgabemodus

LINE → FORM

FORM → LINE, COMP

PHYS → LINE, FORM, COMP

COMP → FORM

soll der Bildschirm gelöscht werden und auf das Standardbildschirmformat zurückgesetzt werden. Wird der Moduswechsel nicht nach einer Eingabe gewechselt, wartet das System t Sekunden (siehe Kommando MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=) vor dem Löschen, damit der Anwender seine letzte Ausgabe lesen kann.

=NO

Beim Wechsel des Ausgabemodus wird der Bildschirm nicht gelöscht. Die nachfolgende Nachricht wird ohne Wartezeit ausgegeben.

*Hinweis*

Im Anwendungsprogramm muß festgelegt sein, daß der Bildschirminhalt keine Funktionen des neueingestellten Modus beeinträchtigt, z.B. durch vormodifizierte Felder.

*Hinweis*

Beim Wechsel des Bildschirmformats durch das Programm bleibt eine zuvor vom Anwender eingestellte logische Zeilenanzahl im neuen Format erhalten.

**Rückinformation (Quittungen) und Fehleranzeigen**

Im rechtsbündigen Byte von Register 15 wird ein Rücksprungschalter RS übergeben, die restlichen drei Byte sind gelöscht.

RS	
X'00'	Normale Beendigung
X'04'	Nicht behebbarer Fehler
X'08'	Operandenfehler
X'0C'	Der aufrufende Prozeß ist kein Teilnehmerprozeß
X'10'	Operandenfehler
X'14'	Falsche Edit-Options-Angabe

### 3.5 Datenstationseigenschaften abfragen: TSTAT

Mit dem Makro TSTAT kann man Informationen über die Datenstation anfordern. Dabei wird der generierte Gerätetyp geliefert.

Der MF-Operand wird akzeptiert (siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen").

Eine CSECT/DSECT wird mit MF=C/D bzw. MF=(C,p)/(D,p) generiert. p = Präfix (max. 3 Zeichen lang, ein längeres Präfix wird auf drei Zeichen gekürzt), Voreinstellung p = TST.

Operation	Operanden
TSTAT	$\left\{ \begin{array}{l} \text{TCHAR} \\ \text{PHDIM} \\ \text{LIDIM} \\ \text{VDTEYP}] \\ \text{EDOPT} \\ \text{OFLOW} \\ \text{STNAM} \\ \text{PRNAM} \\ \text{ALL} \\ \text{MONCS} \\ \text{PERPH} \\ \text{BASIC} \end{array} \right\}, \text{bereich}[, \text{länge}][, \text{MF} = \left\{ \begin{array}{l} \text{L} \\ \text{(E, \dots)} \end{array} \right\}]$
oder	---
TSTAT	$\text{MF} = \left\{ \begin{array}{l} \text{C} \\ \text{(C, p)} \\ \text{D} \\ \text{(D, p)} \end{array} \right\}$

TCHAR	fragt nach dem physikalischen Typ der Datenstation. Es wird der Typ geliefert, unter dem die Datenstation im PDN generiert wurde.
PHDIM	fragt nach den physikalischen Eigenschaften der Datenstation (Zeilenmodus).
LIDIM	fragt nach den logischen Eigenschaften der Datenstation (Zeilenmodus).
VDT[YP]	fragt nach dem logischen Typ der Datenstation.
EDOPT	fragt nach den statischen Edit-Options.
OFLOW	fragt nach der Art der Steuerung bei Bildschirmüberlauf.
STNAM	fragt nach dem Stationsnamen.

PRNAM	fragt nach dem Prozessornamen.
ALL	Sämtliche Informationen außer zu MONCS, PERPH und BASIC werden ausgegeben.
MONCS	fragt nach Informationen über den Monitor und über die Zeichensätze der Datenstation.
PERPH	fragt nach Informationen über die angeschlossene Peripherie
BASIC	fragt nach Grundinformationen über die Datenstation
bereich	symbolische Adresse eines Bereichs, in dem die angeforderte Information abgespeichert wird. Der Bereich muß auf Halbwortgrenze ausgerichtet sein.
länge	gibt die Länge des Bereichs an. Der Ausgabebereich muß mindestens die folgende Länge haben: 64 Byte für ALL 14 Byte für MONCS 64 Byte für BASIC Weitere Informationen dazu finden Sie bei der Beschreibung des Makros DCSTA im VTSU-Handbuch (Siehe [1]). 8 Byte für andere Optionen  Fehlt diese Angabe wird das Längenattribut "Bereich" verwendet. Ist die Bereichslänge bei den Parametern ALL, BASIC oder MONCS kleiner als die angebotene Terminalinformation, werden nur so viele Teilinformationen zur Verfügung gestellt, wie in den Bereich passen.
MF	
=L	Generiert eine Parameterliste, Felder werden gemäß Parameterangaben versorgt.
=(E,...)	Generiert den Befehlsteil.
=C	Generiert eine CSECT.
=(C,p)	Generiert eine CSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = TST.
=D	Generiert eine DSECT.
=(D,p)	Generiert eine DSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = TST.

*Hinweis*

Bei Verwendung einer CSECT/DSECT sind die Felder der Parameterliste wie folgt zu versorgen:

TSTMTYP	immer auf TSTSTMAC setzen,
TSTSTTYP	mit gewünschtem Statustyp gemäß Equates versorgen,
TSTMLEN	Länge des Ausgabebereiches im Anwendungsprogramm (länge),
TSTMADR	Adresse des Ausgabebereiches im Anwendungsprogramm (bereich).

Die Operanden TCHAR, PHDIM, LIDIM, VDTYP, EDOPT, OFLOW, STNAM, PRNAM, ALL, MONCS, PERPH und BASIC entsprechen denen, die bereits beim Makroaufruf DCSTA beschrieben sind (siehe VTSU-Handbuch [1]) .

**Funktionsweise**

Der Bereich, der die Informationen aufnehmen soll, kann vom aufrufenden Programm selbst definiert werden, es kann ihn aber auch durch den Aufruf des Makros DCSTA C,... definieren lassen (siehe Beispiel 2). Definiert das Programm den Empfangsbereich selbst, kann es die Adressierung mit Hilfe einer DSECT durchführen, die durch den Aufruf DCSTA D,... generiert wird.

In den Tabellen, die bei dem Makro DCSTA aufgeführt sind, werden die symbolischen Namen angegeben, die der Makro DCSTA generiert. Namen, die in Klammern stehen, sind keine symbolischen Adressen, sondern symbolische Konstanten, die das Prüfen einzelner Bits erleichtern (siehe Beispiel 2).

**Rückinformation (Quittungen) und Fehleranzeigen**

Im rechtsbündigen Byte von Register 15 wird ein Rücksprungschalter RS übergeben, die restlichen drei Byte sind gelöscht.

RS	
X'00'	normale Beendigung
X'04'	nicht behebbarer Fehler
X'08'	Parameterfehler
X'0C'	Keine Datenstation vorhanden
X'10'	Länge des Empfangsfeldes zu klein. Wenn die Operanden ALL, MONCS, BASIC angegeben wurden, wird nur ein Teil der Information geliefert. Bei allen anderen Operanden wird nichts übertragen.
X'14'	Die gewünschte Information ist (teilweise) nicht verfügbar.



**Makroauflösung**

```

                                EXTERNAL SYMBOL DICTIONARY
TSTAT1  START
        TSTAT MF=D
        #INTF INTNAME=TSTAT,INTCOMP=001,REFTYPE=REQUEST
TSTPAR  DSECT
*
*      TSTAT PARAMETER BLOCK
*
TSTMTYP DC   AL1(0)           MACRO TYPE
TSTSTMAC EQU   1             TSTAT-MACRO
*
TSTSTTYP DC   AL1(0)           STATUS TYPE REQUESTED:
*
        @STIN ,MF=D,PRE=TST
*
*      TSTAT EQUATES
*
TSTTCHAR EQU   1             - TERMINAL CHARACTERISTICS
TSTPHDIM EQU   2             - PHYSICAL TERMINAL ATTRIBUTES
TSTLIDIM EQU   3             - VIRTUAL TERMINAL ATTRIBUTES
TSTVDT  EQU   4             - VIRTUAL DEVICE TYPE
TSTEDOPT EQU   5             - STATIC EDIT OPTIONS
TSTOFLOW EQU   6             - OVERFLOW CONTROL
TSTSTNAM EQU   7             - STATION NAME
TSTPRNAM EQU   8             - PROCESSOR NAME
TSTMONCS EQU   9             - MONITOR AND CHARACTER SETS
TSTPERPH EQU  10             - PERIPHERIE OF TERMINAL
TSTBASIC EQU  11             - BASIC TERMINAL INFORMATIONS
TSTALL  EQU  255             - ALL INFORMATION REQUESTED
*
*      *,@STIN      999      921011  53531032
TSTMLEN DC   H'0'           AREA LENGTH (MIN. 8 BYTES)
*
TSTMADR DC   A(0)           AREA ADDRESS FOR TSTAT INFO
        END

```

*Beispiel 1*

```

TSTAT1  START
        PRINT NOGEN
        BALR 10,0
        USING *,10
        TSTAT VDTYP,LOG,8 _____ (01)
        TSTAT OFLOW,UEL,8 _____ (03)
DTH1    TERM
        *,VERSION 900
*
LOG     DS     CL8
UEL     DS     CL8
        END

/START-PROG $ASSEMBH
% BLS0500 PROGRAM 'ASSEMBH', VERSION '1.1A00' OF '1992-04-30' LOADED
% BLS0552 COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1990. ALL RIGHTS
% RESERVED
% ASS6010 V 1.1A00 OF BS2000 ASSEMBH- READY
//COMPILE SOURCE=*L(L=LIB.EX.MANUAL,E=TSTAT1),MAC-LIB=$XXX.LIB.M.SYSLIB.V11.0T10,
MOD-LIB=LIB.EX.MANUAL(E=TSTAT1),LIST=P(OUT=*L(L=LIB.EX.MANUAL,E=TSTAT1)),TEST-SUP=YES
% ASS6011 ASSEMBLY TIME: 1056 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
% ASS6006 LISTING GENERATOR TIME: 760 MSEC
//END
% ASS6012 END OF ASSEMBH-
/LOAD-PROG *M(L=LIB.EX.MANUAL,E=TSTAT1),TEST-OPT=AID
% BLS0517 MODULE 'TSTAT1' LOADED
%insert dth1
/RESUME-PROG
STOPPED AT LABEL: DTH1 , SRC_REF: 38 , SOURCE: TSTAT1 , PROC: TSTAT1
/%d log %x18
** ITN: #'00080107' *** TSN: 2UXT *****
CURRENT PC: 00000024 CSECT: TSTAT1 *****
V'00000036' = LOG + #'00000000'
00000036 (00000000) 5B010000 00000000 $..... _____ (02)
/%d uel %x18
V'0000003E' = UEL + #'00000000'
0000003E (00000000) 02060000 00000000 ..... _____ (04)
/RESUME-PROG
/MOD-JOB-OPT LOG=P(L=NO)

```

- (01) Der logische Typ der Datenstation wird abgefragt.
- (02) Der Prozeß läuft an einer Datensichtstation (Bit 2<sup>6</sup> gesetzt), an der Zeilenmodus, Formatmodus und physikalischer Modus erlaubt sind (Bits 2<sup>0</sup>, 2<sup>1</sup> und 2<sup>3</sup> gesetzt).
- (03) Die Art der Steuerung bei Bildschirmüberlauf wird abgefragt.
- (04) Das System (Bit 2<sup>5</sup>=0) steuert den Überlauf. Wenn der Bildschirm voll ist, fordert das System an der Datenstation eine Quittung an, bevor es ihn überschreibt (Bit 2<sup>1</sup>=1).

## Beispiel 2

```

                                EXTERNAL SYMBOL DICTIONARY
TSTAT2  START
        PRINT NOGEN
        BALR 3,0
        USING *,3
        TSTAT TCHAR,STACHAR,8 _____ (01)
           CLI STADVTP,STAD6180
        TSTAT PHDIM,PHYSFELD,8 _____ (02)
        TSTAT LIDIM,FILIDIM,8 _____ (03)
DTH1    TERM
        *,VERSION 100
        PRINT GEN
        DCSTA C,TYPE=TCHAR _____ (04)
        IDLKG VER=002,ID=TCHAR,SECT=C,P=STA,ALIGN=D 002
        *,VERSION 002
        CNOP 0,8
        STACHAR DS 0D
*
*
*
        DEFINE TERMINAL CHARACTERISTICS FIELDS
*
*
        STASTTCH DS 0D                TERMINAL CHARACTERISTICS AREA
        STAMNTCH DS 0D                MINIMUM TERMINAL CHARICS. AREA
*
*
        STAPTTYPC DC AL1(0)           PARTNERTYPE
        STADVTP DC AL1(0)            DEVICE TYPE
        STATCHR2 DC AL1(0)           TERMINAL CHARACTERISTICS BYTE 2
        STATCHR3 DC AL1(0)           TERMINAL CHARACTERISTICS BYTE 3
        STATCHR4 DC AL1(0)           TERM. CHARACTERISTIC BYTE 4 901
        STACHRS DC AL1(0)            TERM. CHAR FROM STATION 920
        STACTRLU DC AL1(0)           CONTROL UNIT FOR PRINTER 701
        STACHCAD DC AL1(0)           CENTRAL HARDCOPY ADDRESS
*
*
        DEFINE PARTNER TYPES (PTTYP)
*
*
        STADCAMP EQU X'00'           PARTNER IS A PROGRAM
        STADCAMT EQU X'01'           PARTNER IS A TERMINAL
*
        DCDEVCH STA
*
*
        DEFINE DEVICE TYPES (DVTYP)
*
*
        STAD8103 EQU X'02'           TELETYPE 8103
        STAD8150 EQU X'04'           VIDEO TERMINAL 8150
        STAD8153 EQU X'05'           *NO VTSU* VIDEO TERMINAL 8153
        STADHOST EQU X'08'           INTELLIGENT PARTNER
        STAD8151 EQU X'15'           VIDEO TERMINAL 8151
        STAD8152 EQU X'16'           VIDEO TERMINAL 8152
        STAD8110 EQU X'17'           SS-8110
        STAD6154 EQU X'18'           *NO VTSU* VIDEO 8161 54 CHAR PER LINE
        STAD6164 EQU X'19'           *NO VTSU* VIDEO 8161 64 CHAR PER LINE
        STAD6180 EQU X'1A'           *NO VTSU* VIDEO 8161 80 CHAR PER LINE
        STAD8161 EQU X'1A'           *NO VTSU* VIDEO 8161
        STAD8121 EQU X'1C'           PRINTER STATION 8121
        STADPT80 EQU X'1D'           *AS 8103* TELETYPE PT80
        STAD1000 EQU X'1E'           *AS 8103* TELETYPE T1000
        STADT100 EQU X'23'           *AS 8103* TELETYPE T100
        STAD100E EQU X'26'           *AS 8103* FS100-E
        STAD8122 EQU X'2B'           PRINTER STATION 8122

```

```

STAD8162 EQU X'2C' VIDEO 8162
STAD8160 EQU X'2D' VIDEO 8160
STAD8124 EQU X'2E' PRINTER STATION 8124
STAD8167 EQU X'2F' *AS 8160* VIDEO 8167
STADAP EQU X'30' *AS HOST* AP-STATION
STAD9750 EQU X'35' VIDEO 9750 OR 9749
STAD9003 EQU X'36' PRINTER STATION 9003
STAD9770 EQU X'39' *AS 8151* DS 9770
STAD9002 EQU X'3B' PRINTER STATION 9002
STAD3974 EQU X'3D' VIDEO TERMINAL 3974
STAD9751 EQU X'3F' *AS 8160* DSS 9751
STAD9752 EQU X'40' *AS 9750* DSS 9752
STAD9753 EQU X'41' *AS 9750* DSS 9753
STAD9001 EQU X'42' PRINTER 9001
STAD9731 EQU X'43' *AS 3974* GRAFIC STATION 9731
STAD9004 EQU X'45' PRINTER 9004
STAD9754 EQU X'4C' *AS 8160* VIDEO 9754
STAD9755 EQU X'4E' DSS-9755 922
STAD9763 EQU X'4F' DSS-9763
STADBTXF EQU X'55' *AS HOST* BTX-STATION T-3000 (FELDVERS.)
STADBTXE EQU X'56' *AS HOST* BTX-EDITIER-STATION (DIENST)
STADBTXA EQU X'57' *AS HOST* BTX-ABFRAGE-STATION (DIENST)
STADUTC EQU X'5A' UTC FUER TELETEx
STAD9012 EQU X'5B' PRINTER 9012
STAD9013 EQU X'5C' PRINTER 9013
STAD3270 EQU X'5E' DSS-3270
STAD0131 EQU X'65' PRINTER 9001-31
STAD0189 EQU X'66' PRINTER 9001-8931
STAD9022 EQU X'68' PRINTER 9022
STAD1118 EQU X'6B' PRINTER 9011-18
STAD1119 EQU X'6C' PRINTER 9011-19
STAD3287 EQU X'6E' PRINTER 3287 956
STADPCL EQU X'70' PRINTERS PCL 960
STAD9021 EQU X'70' PRINTERS 9021 / 9022-200, HP LJ960
STAD9014 EQU X'72' PRINTER 9014 001
STAD9026 EQU X'73' PRINTER 9026 (HDLC,COMP.9025) 001
STADFE EQU X'78' DSS-FE BOF-FRONT-END 999
*
* DEFINE TERMINAL CHARACTERISTICS BYTE 2 (TCHR2) BITS
*
STATC2EX EQU 8 SECONDARY CHARACTER SET
STATC2LC EQU 32 LOWER CASE
STATC2DT EQU 64 GERM KEYB WITH GERM NAT CHAR901
STATC2DF EQU 128 BYTE 2 DEFINED
*
* DEFINE TERMINAL CHARACTERISTICS BYTE 3 (TCHR3) BITS
*
STATC3H1 EQU 1 HARDCOPY BIT 1 (LOCAL)
STATC3H2 EQU 2 HARDCOPY BIT 2 (CENTRAL)
STATC3HC EQU 3 HARDCOPY BITS
STATC3IC EQU 4 IDENTITY CARD READER
STATC3FD EQU 8 FLOPPY DISK
STATC3AP EQU 16 APL CAPABILITY
STATC3GF EQU 32 GRAPHICS
STATC3DZ EQU 64 DEZENTRAL FORMATING
STATC3DF EQU 128 BYTE 3 DEFINED
*
* DEFINE TERMINAL CHARACTERISTICS BYTE 4 (TCHR4) BITS
STATC4C0 EQU 1 4 COLOURS(ITALIC/HALFBRIGHT)
STATC4ZF EQU 2 NEW ZAT AND FAT POSSIBLE 920

```

```

STATC4ST EQU 4 STATUS QUERY POSSIBLE 920
STATC4HI EQU 8 HARDWARE INFOLINE AVAILABLE 920
STATC4C8 EQU 16 8 COLOURS 953
STATC4HP EQU 32 HP LASER JET II 954
STATC4DF EQU 128 BYTE 4 DEFINED
*
* DEFINE TERM CHAR FROM STATION BYTE (TCHRS) BITS 920
*
STATCSDT EQU 1 GERMAN KEYBOARD 920
STATCSHC EQU 2 LOCAL HARDCOPY PRINTER 920
STATCSIC EQU 4 ID-CARD RAEDER 920
STATCSDF EQU 128 TERM CHAR FROM STAT RECEIVED920
*
*,DCDEVCH 001 920406 53113038
*
*,DCSTA 002 920819 53531014
PHYSFELD DCSTA C,TYPE=PHDIM _____ (05)
PHYSFELD IDLKG VER=002,ID=PHDIM,SECT=C,P=STA,ALIGN=D 002
*
CNOP 0,8
PHYSFELD DS 0D
*
* DEFINE PHYSICAL TERMINAL ATTRIBUTES FIELDS
*
STASTPV DS 0D PHYSICAL TERMINAL ATTR. AREA
STAMNPV DS 0D MINIMUM PHYS. TERM. ATTR. AREA
*
STALLEN DC H'0' PHYSICAL LINE LENGTH
STANOLIN DC H'0' PHYSICAL NUMBER OF LINES
STAMAXDB DC H'0' MAX. PHYSICAL DEVICE BUFFER
DC 2ALL(0) RESERVED FOR FUTURE DEVELOPMENT
*,DCSTA 002 920819 53531014
DCSTA C,FI,TYPE=LIDIM _____ (06)
IDLKG VER=002,ID=LIDIM,SECT=C,P=FI,ALIGN=D 002
*
CNOP 0,8
FILIDIM DS 0D
*
* DEFINE VIRTUAL TERMINAL ATTRIBUTES FIELDS
*
FISTLV DS 0D VIRTUAL TERMINAL ATTR. AREA
FIMNLV DS 0D MINIMUM VIRTUAL TERM ATTR AREA
*
FILLLEN DC H'0' VIRTUAL LINE LENGTH
FILNOLN DC H'0' VIRTUAL NUMBER OF LINES
FILMAXB DC H'0' MAXIMUM VIRTUAL DEVICE BUFFER
DC 2ALL(0) RESERVED FOR FUTURE DEVELOPMENT
*,DCSTA 002 920819 53531014
END
    
```

- (01) Der physikalische Typ der Datenstation wird abgefragt. Der Empfangsbereich, der durch den Aufruf DCSTA C,... generiert wird, hat standardmäßig den Namen STATCHAR. Zur Demonstration wird das Feld STADV TYP mit Hilfe der symbolischen Konstanten STAD6180 abgefragt.
- (02) Die physikalischen Eigenschaften der Datenstation werden abgefragt. PHYSFELD ist der vom Anwender gewählte Name des Empfangsbereichs.
- (03) Die logischen Eigenschaften der Datenstation werden abgefragt. Bei dem Namen des Empfangsbereichs FILIDIM ist der Standard-Präfix 'STA' durch den Präfix 'FI' ersetzt.
- (04) Der Empfangsbereich für die Abfrage des physikalischen Typs wird generiert, zusammen mit den symbolischen Konstanten zur Abfrage von Bitwerten.
- (05) Der Empfangsbereich für die Abfrage der physikalischen Eigenschaften wird generiert.
- (06) Der Empfangsbereich für die Abfrage der logischen Eigenschaften wird generiert. Der Präfix der Feldnamen soll 'FI' lauten (Standard 'STA').

```

/START-PROG $ASSEMBH
% BLS0500 PROGRAM 'ASSEMBH', VERSION '1.1A00' OF '1992-04-30' LOADED
% BLS0552 COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1990. ALL RIGHTS
% RESERVED
% ASS6010 V 1.1A00 OF BS2000 ASSEMBH- READY
//COMPILE SOURCE=*L(L=LIB.EX.MANUAL,E=TSTAT2),MAC-LIB=$XXXX.LIB.M.SYSLIB.V11.0T10,
MOD-LIB=LIB.EX.MANUAL(E=TSTAT2),LIST=P(OUT=*L(L=LIB.EX.MANUAL,E=TSTAT2)),TEST-SUP=YES
% ASS6011 ASSEMBLY TIME: 1411 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
% ASS6006 LISTING GENERATOR TIME: 877 MSEC
//END
% ASS6012 END OF ASSEMBH-
/LOAD-PROG *M(L=LIB.EX.MANUAL,E=TSTAT2),TEST-OPT=AID
% BLS0517 MODULE 'TSTAT2' LOADED
%insert dth1
/RESU-PROG
STOPPED AT LABEL: DTH1 , SRC_REF: 52 , SOURCE: TSTAT2 , PROC: TSTAT2
/d statchar %x18
** ITN: #'00080107' *** TSN: 2UXT *****
CURRENT PC: 00000038 CSECT: TSTAT2 *****
V'00000050' = STATCHAR + #'00000000'
00000050 (00000000) 0135A081 88000000 ...ah... — (07)
/d physfeld %x18
V'00000058' = PHYSFELD + #'u0000000'
00000058 (00000000) 00500018 0FF80000 .&...8.. — (08)
/d filidim %x18
V'00000060' = FILIDIM + #'00000000'
00000060 (00000000) 00500018 07800000 .&..... — (09)
/RESU-PROG
/MOD-JOB-OPT LOG=P(L=N0)

```

- (07) Der Prozeß läuft an einer Datensichtstation 9750.
- (08) X'0050' Die physikalische Zeilenlänge beträgt 80 Zeichen.  
X'0018' Die physikalische Zeilenanzahl beträgt 24 Zeilen.  
X'0FF8' Der physikalische Gerätepuffer faßt 4088 Zeichen.
- (09) X'0050' Die logische Zeilenlänge beträgt 80 Zeichen (Zeilenmodus).  
X'0018' Die logische Zeilenanzahl beträgt 24 Zeilen (Zeilenmodus).  
X'0780' Der logische Zeichenpuffer faßt 1920 Zeichen  
(= 24 Zeilen mal 80 Spalten).

Weitere Beispiele siehe Makro DCSTA im VTSU-Handbuch [1].

## 3.6 Daten ausgeben nach SYSOUT: WROUT

Mit dem Makro WROUT kann eine Nachricht in die Datei SYSOUT übertragen werden. Die Datei, die als SYSOUT erklärt ist, kann ein Element einer PLAM-Bibliothek, eine SDF-P-Variable, eine katalogisierte SAM- oder ISAM-Datei oder insbesondere die prozeßführende Datenstation sein. Wird der Operand LOGGING=PAR(LISTING=YES) im SET-LOGON-PARAMETERS- oder /MODIFY-JOB-OPTIONS-Kommando angegeben, so werden die Ausgaben des WROUT-Makros auch nach SYSLST übertragen.

Der Makro CUPAB generiert eine DSECT für die Operandentabelle des WROUT für den Aufruf im 24-Bit-Adressierungsmodus.

Der Makroaufruf WROUT akzeptiert den MF-Operanden (siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen").

Der Operand PARMOD steuert die Makroauflösung. Damit kann entweder die 24-Bit-Schnittstelle oder die 31-Bit-Schnittstelle generiert werden.

Für den 31-Bit-Adressierungsmodus ist folgendes zu beachten:

- Am Anfang der Operandenliste wird ein Standardheader (siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen") generiert. Der Anwender muß dafür sorgen, daß dieser Header korrekt initialisiert ist. Bei Verwendung von dynamisch erzeugten Operandenlisten (Zugriff über CSECT/DSECT) werden für den Standardheader keine symbolischen Namen und Equates erzeugt. Deshalb müssen bei dynamischer Versorgung der Operandenliste die Initialisierungswerte für den Standardheader aus einer mit MF=L erzeugten Operandenliste übernommen werden.
- Eine CSECT/DSECT wird mit MF=C/D bzw. MF=(C,p)/(D,p) generiert.  
p = Präfix (max. 3 Zeichen, ein längeres Präfix wird auf 3 Zeichen gekürzt);  
Voreinstellung p = CUW. Das Präfix verändert nur die Feldnamen, nicht die symbolischen Namen bei den Equates.

### *Hinweis zur Kompatibilität*

Der Makroaufruf WROUT ab der TIAM-Version V10.0A (gekoppelt an BS2000 V10.0A) erzeugt die Headerversion 2 im Standardheader. Damit ist er in BS2000-Versionen kleiner V10.0 nicht ablauffähig und verzweigt dort auf den Fehlerausgang. Soll ein Programm in mehreren BS2000-Versionen ausführbar sein, so muß das Programm in der niedrigsten Version erstellt sein (es wird nur Aufwärtskompatibilität garantiert).



**Format 1**

Operation	Operanden
WROUT	<p>satz,fehler[,edit]</p> $\left[ \begin{array}{l} 24[,MF=\left\{ \begin{array}{l} L \\ (E,\dots) \end{array} \right\}] \\ \\ [,PARMOD=\left\{ \begin{array}{l} C \\ (C,p) \\ D \\ (D,p) \\ L \\ (E,\dots) \end{array} \right\}] \\ \\ 31[,MF=\left\{ \begin{array}{l} L \\ (E,\dots) \end{array} \right\}] \end{array} \right]$ $\left[ \begin{array}{l} ,MODE=COMP [,OTRSUP=\left\{ \frac{N[O]}{Y[ES]} \right\}] [,OLINEND=\left\{ \frac{N[O]}{Y[ES]} \right\}] \\ \\ [,OHCOPY=\left\{ \frac{N[O]}{Y[ES]} \right\}] [,OHDR=\left\{ \frac{N[O]}{Y[ES]} \right\}] \\ \\ ,MODE=LINE [,OHCOPY=\left\{ \frac{N[O]}{Y[ES]} \right\}] [,OHOM=\left\{ \frac{N[O]}{Y[ES]} \right\}] \\ \\ [,OINFO=\left\{ \frac{N[O]}{Y[ES]} \right\}] [,ONOPSN=\left\{ \frac{N[O]}{Y[ES]} \right\}] \\ \\ [,OBELL=\left\{ \frac{N[O]}{Y[ES]} \right\}] [,ONOLOGC=\left\{ \frac{N[O]}{Y[ES]} \right\}] \\ \\ [,EXTEND=\left\{ \frac{N[O]}{Y[ES]} \right\}] \\ \\ ,MODE=FORM \\ \\ ,MODE=PHYS [,OHDR=\left\{ \frac{N[O]}{Y[ES]} \right\}] [,OETB=\left\{ \frac{N[O]}{Y[ES]} \right\}] \\ \\ [,OTRANS=\left\{ \frac{N[O]}{Y[ES]} \right\}] \end{array} \right]$ <p>[,RC=<math>\left\{ \begin{array}{l} OLD \\ NEW \end{array} \right\}</math>]</p> <p>[,VTSUCBA=addr]</p>

Die Operanden edit, MODE, RC und VTSUCBA werden nur ausgewertet, wenn SYSOUT der Datenstation zugewiesen ist.

Die Operanden RC und VTSUCBA werden nur für die 31-Bit-Schnittstelle unterstützt

satz Symbolische Adresse des auszugebenden Datensatzes. Der Satz beginnt mit dem Satzlängenfeld, gefolgt von dem Vorschubsteuerzeichen und der auszugebenden Nachricht.

Satzaufbau:

Byte	0-1:	Länge der Nachricht + 4 byte Satzlängenfeld, die Länge muß größer als 5 sein.
	2-3:	reserviert
	4:	Vorschubsteuerzeichen; wird nur bei Ausgabe auf Drucker ausgewertet
	5-n:	Nachricht

*Beispiel*

satz	DC	Y(satzend-satz)	
	DS	CL2	reservierte Byte
	DC	X'nm'	Druckvorschubzeichen
	DC	C'datensatz'	zu übertragender Datensatz
satzend	EQU	*	

*Hinweis*

Wenn die Systemdatei SYSOUT eine katalogisierte Datei ist, so werden Sätze, die größer als 2044 byte sind, abgeschnitten (Register R15=X'0C') und nur der erste Teil des Satzes in die Datei geschrieben. Bei EAM-Dateien werden Sätze, die größer als 2040 byte sind, ebenfalls abgeschnitten.

fehler Symbolische Adresse einer Fehleroutine im Anwendungsprogramm, zu der im Fehlerfall verzweigt wird.

Im Fehlerfall enthält Register R14 die Adresse des Befehls, der dem WROUT-Aufruf folgt. Der Fehlercode wird im Register R15 übergeben.

31-Bit-Schnittstelle: Bei Angabe fehler = 0 (Adresse X'00..0') wird das Programm mit dem Befehl fortgesetzt, der dem WROUT-Aufruf folgt.

edit	<p>Aufbereitungsfunktion (Edit-Options) für eine Nachricht, die zur Datenstation übertragen werden soll. Die Aufbereitung wird übergangen, wenn SYSOUT keine Datenstation ist. Dieser Operand ist nicht erforderlich, wenn Standardfunktionen (alle Edit-Bits=0) verwendet werden, bei einer MODE-Angabe oder bei Nutzung des VTSU-Control-Blocks. Durch Direktangabe (X'xx') kann nur das 1. Edit-Byte für Ausgabe auf die beim CUPAB-Makro beschriebene Bedeutung gesetzt werden.</p> <p><i>Hinweis</i></p> <p>Dieser Operand wird nur noch aus Gründen der Kompatibilität unterstützt. Die Edit-Optionen sollten über MODE-Angaben (siehe unten) oder den VTSU-Control-Block (VTSUCBA) gesteuert werden.</p>
PARMOD	<p>Steuert die Makroauflösung. Es wird entweder die 24-Bit-Schnittstelle oder die 31-Bit-Schnittstelle generiert. Wenn PARMOD nicht angegeben wird, erfolgt die Makroauflösung entsprechend der Angabe für den Makro GPARMOD (siehe Handbuch "Makroaufrufe an den Ablaufteil" [3]) oder der Voreinstellung für Assembler (= 24-Bit-Adressierung).</p>
=24	Die 24-Bit-Schnittstelle wird generiert. Datenlisten und Befehle benutzen 24-Bit-Adressen (Adreßraum ≤ 16MB).
=31	Die 31-Bit-Schnittstelle wird generiert. Datenlisten und Befehle benutzen 31-Bit-Adressen (Adreßraum ≤ 2GB). Datenlisten beginnen mit dem Standardheader.
MF	
=L	Generiert eine Parameterliste, Felder werden gemäß Parameterangaben versorgt.
=(E,...)	Generiert den Befehlsteil.
=C	Generiert eine CSECT.
=(C,p)	Generiert eine CSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = CUW.
=D	Generiert eine DSECT.
=(D,p)	Generiert eine DSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = CUW.

Die MODE-Angaben zusammen mit den Edit-Optionen werden nur noch aus Kompatibilitätsgründen unterstützt. Sie werden jetzt im VTSU-Control-Block (VTSUCB) zusammengefaßt. Künftige Erweiterungen erfolgen nur noch im VTSUCB.

MODE	Dieser Operand wird nur ausgewertet, wenn SYSDTA der Datenstation zugeordnet ist.
=COMP	<p>Kompatibler Betriebsmodus. Eventuell direkt im Operanden edit gemachte Angaben werden ignoriert. Steuerzeichen in der Ausgabenachricht sind zulässig, werden aber vom System nicht auf Richtigkeit geprüft. Diese Betriebsweise ist kompatibel zu früheren Versionen des Betriebssystems.</p> <p>Für die Geräte 8160, 8162, 9749, 975x, 9763, 3270 und X.29-Datenstationen wird dieser Modus wie MODE=LINE behandelt. Die Edit-Option OLINEND wird ignoriert. Die Edit-Options OTRSUP und OHDR werden abgewiesen (RS: X'08').</p>
=LINE	<p>Die aktuelle Datenstation soll als logische Zeilen- bzw. Seiten-Datenstation behandelt werden. Die Nachricht kann durch logische Steuerzeichen strukturiert werden (siehe Makroaufruf VTCSET). Ist SYSOUT keine Datenstation, so werden nur die logischen Steuerzeichen NL und NP ausgewertet, z.B bei Ausgaben auf Drucker im Stapelbetrieb.</p> <p>Weitere Steuerzeichen sind unzulässig und werden vom System in ein vom Anwender definiertes Ersatzzeichen umgewandelt (siehe Kommando MODIFY-TERMINAL-OPTIONS SUBSTITUTE-CHARACTER=).</p>
=FORM	Format-Modus. Das Anwendungsprogramm arbeitet mit der Software-Komponente FHS bzw. Formatsteuerung, die auch die datenstationsgerechte Aufbereitung der Ausgabenachricht vornimmt.
=PHYS	Die Nachricht soll physikalisch, d.h. ohne Aufbereitung durch das System an die Datenstation ausgegeben werden. Damit können spezielle Gerätefunktionen angesprochen werden, für die der LINE- oder FORM-Modus nicht ausreicht. Wird keine der zulässigen Edit-Optionen angegeben, stellt das System der Nachricht einen gerätespezifischen Standard-Nachrichtenkopf voran.
OTRSUP	
=YES	Die Übersetzung der Nachricht wird unterdrückt. Das Programm muß in diesem Fall die Nachricht im Gerätecode liefern.
= <u>NO</u>	Die Übersetzung der Nachricht von EBCDIC in den Gerätecode wird nicht unterdrückt, d.h. das Programm liefert die Nachricht in EBCDI-Code. Im System wird die Nachricht in den Gerätecode übersetzt.

## OLINEND

=YES

Die Ausgabe der Nachricht zur Datenstation erfolgt ohne vom System bereitgestellte Steuerzeichen für Wagenrücklauf/ Zeilenvorschub. Die Steuerung muß vom Anwendungsprogramm vorgenommen werden.

=NO

Jede Ausgabenachricht beginnt bei einer Datenstation immer mit einer neuen Zeile. Die dafür notwendigen Steuerzeichen werden vom System der Nachricht vorangestellt bzw. bei Erreichen des physikalischen Zeilenendes in die Nachricht eingeschoben, wenn es die betreffende Datenstation erforderlich macht.

## OHCOPY

=YES

Die Ausgabenachricht für eine Datensichtstation wird gleichzeitig über ein dort angeschlossenes Hardcopy-Gerät (Drucker) ausgedruckt. Das Hardcopy-Gerät muß generiert werden oder mittels MODIFY-TERMINAL-OPTIONS zugewiesen werden.

**Bei 3270:**

Der Inhalt des gesamten Bildschirms wird über das Hardcopy-Gerät ausgedruckt. Dadurch werden eventuell auch frühere Aus- und Eingaben ausgedruckt. Bei mehreren direkt aufeinanderfolgenden Ausgaben wird die Hardcopy-Funktion nur bei der letzten Ausgabe ausgelöst. Der Hardcopy-Abdruck erfolgt nur dann, wenn für die Datensichtstation beim Verbindungsaufbau ein Hardcopy-Gerät generiert wurde.

Wurde OINFO=YES/MODE=INFO oder EXTEND=YES/MODE=EXTEND angegeben, erfolgt kein Hardcopy-Abdruck.

Wird OHCOPY=YES/MODE=YES verwendet und enthält die Nachricht das logische Steuerzeichen SPA, EPA, CHS oder NUM, so wird nicht die gesamte Nachricht, sondern nur der letzte ungeschützte Teil der Nachricht abgedruckt.

Wird gleichzeitig OVERFLOW-CONTROL=NO (MODIFY-TERMINAL-OPTIONS-Kommando) verwendet, kann es vorkommen, daß nur ein Teil der Ausgabe auf dem Hardcopy-Gerät wiedergegeben wird.

=NO

Es findet keine Druckausgabe über ein Hardcopy-Gerät statt.

## OHDR

=YES

Die Nachricht enthält einen anwenderindividuellen Nachrichtenkopf (Nachrichtenkopf ist im US-ASCII-Code anzugeben), den das System dem Ausgabertext voranstellt. Die Länge des Nachrichtenkopfs +1 muß im ersten Byte der Nachricht binär angegeben werden.

**Bei 3270:**

Der Nachrichtenkopf wird im EBCDI-Code angegeben und besteht aus dem CMD-Byte und dem WCC-Byte. Vor diesem Nachrichtenkopf muß ein Byte mit dem Inhalt X'01' gesetzt werden (Länge des TRANSDATA-Nachrichtenkopfes + 1).

*Hinweis*

Bei Ausgabe auf die Datensichtstationen 8160, 975x, 9763 und hieran lokal angeschlossene Drucker ist zu beachten, daß das System (MODE=LINE bzw. COMP) oder FHS mit MODE=FORM keinen Nachrichtenkopf verwendet (PARAM0, PARAM1), sondern mit Parameterangaben (PAG) arbeitet. Die Unterschiede zwischen diesen beiden Arbeitsweisen finden Sie beschrieben in den Handbüchern für Datensichtstationen bzw. Drucker.

=NO

Den Nachrichtenkopf (im US-ASCII-Code) stellt das System dem Ausgabertext voran.

## OHOM

=YES

Nur bei Datensichtstationen 816x, 9749, 975x und 9763 anzuwenden.

Die Nachricht soll unstrukturiert, homogen ausgegeben werden, d.h. die gesamte Nachricht wird als eine Ausgabeeinheit betrachtet. Die Nachrichtenlänge ist durch die Größe des Ausgabepuffers im System beschränkt.

## Wirkung bei Betriebsart 1:

Durch Modifikation eines Zeichens einer Ausgabenachricht kann die gesamte Nachricht wieder zurückübertragen werden, sofern diese keine logischen Anzeigesteuerzeichen enthält.

*Hinweis*

Ist SYSOUT einer Datei zugeordnet wird der WROUT nicht ausgeführt und die SYSLST-Protokollierung unterdrückt.

=NO

Die Nachricht soll strukturiert, nicht homogen ausgegeben werden, d.h. als Ausgabeeinheit wird eine logische Zeile betrachtet. Die Nachrichtenlänge ist nicht beschränkt, falls die logischen Zeilen nicht länger als 255 Zeichen sind.

Wirkung bei Betriebsart 1:

Einzelne logische Zeilen können getrennt modifiziert und damit gezielt zurückübertragen werden.

OINFO

=YES

Die Nachricht kann in einer speziellen Informationszeile abgebildet werden, ohne an der Datenstation wichtige Daten zu zerstören.

Die Angabe ist vor allem für die Benutzerprogramme gedacht, die "asynchron" Nachrichten an Datenstationen senden, ohne die aktuelle Datenstationsanzeige zu kennen.

Die Abbildung erfolgt

- geschützt in einer Hardware-Anzeigezeile (z.B. DSS 9749, 975x, 976x) oder
- geschützt in der letzten Bildschirmzeile (z.B. DSS 816x, 9751, 9753, 3270), wenn es im Anwendungsprogramm festgelegt wird (siehe Makroaufruf TCHNG, Operand INFOLIN), nach vorhergegangener Ausgabe mit MODE=FORM oder MODE=PHYS.
- in allen anderen Fällen, wie eine normale Line-Mode-Nachricht.

Ist die Nachricht länger als eine Bildschirmzeile, wird sie aufgeteilt und Zeile für Zeile ausgegeben. Das System berücksichtigt dabei die durch das Kommando /MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=TIME eingestellte Wartezeit.

Bei OINFO=YES wird die Angabe OHCOPY=YES ignoriert, d.h. es werden weder die Informationszeile noch der Bildschirminhalt abgedruckt.

Nach der nächsten Eingabe, wird die Hardware-Anzeigezeile zurückgesetzt.

=NO

Die Nachricht wird nicht in der speziellen Informationszeile ausgegeben.

ONOPSN

=YES

Die Nachricht beginnt am Anfang der aktuellen Zeile (gilt nur für Schreibstationen).

<u>=NO</u>	Die Nachricht beginnt am Anfang der nächsten Zeile.
OBELL	
<u>=YES</u>	Bei der Ausgabe ertönt am Ende der Nachricht ein akustisches Signal (nur bei den Datenstationen 9749, 975x, 9763, 816x und bei 3270 mit einem speziellen Gerätezusatz).
<u>=NO</u>	Bei der Ausgabe ertönt kein akustisches Signal.
ONOLOGC	
<u>=YES</u>	Logische Steuerzeichen werden nicht ausgewertet. Alle Zeichen, die im EBCDI-Code kleiner als X'40' sind, werden durch SUB ersetzt. Nur Zeichen größer oder gleich X'40' werden durchgelassen.
<u>=NO</u>	Alle logischen Steuerzeichen werden ausgewertet und spezielle physikalische Steuerzeichen werden durchgelassen (siehe VTCSET z.B. ESC, DC4). Andere Zeichen kleiner X'40' werden durch SUB ersetzt. Zeichen größer oder gleich X'40' werden durchgelassen.
EXTEND	
<u>=YES</u>	(Nur für Datenstationen 9749, 975x, 9763, 816x und 3270)  Diese Angabe unterstützt die Verwendung von geschützten und ungeschützten Feldern mit Hilfe der logischen Steuerzeichen EPA, DAR, NUM und SPA (siehe Makro VTCSET).  Die Ausgabe des Textes erfolgt standardmäßig geschützt und halbhell. Die Nachricht kann durch logische Steuerzeichen strukturiert werden (siehe VTCSET). Bei 3270-Datenstationen ist zu beachten, daß die logischen Steuerzeichen Platz auf dem Bildschirm beanspruchen. Mehrere logische Steuerzeichen hintereinander benötigen aber nur einen Platz. Bereiche, in die der Anwender eingeben kann, werden mit EPA, DAR oder NUM begonnen und mit SPA beendet.  Bei Ein- und Ausgabe wird NIL als erlaubtes Zeichen behandelt, es wird vom Programm zur Datenstation und umgekehrt geschickt. Bei 3270-Datenstationen ist zu beachten, daß NIL-Zeichen nicht zur DVA übertragen werden. VTSU-B ergänzt Felder, die bei der Eingabe verkürzt zurückkommen, durch NIL-Zeichen auf ihre ursprüngliche Länge. Dadurch bekommt der Anwender die Felder immer in der Ausgabelänge zurück.



Der Anfang einer Ausgabenachricht wird an dem der Schreibmarke folgenden nächsten Zeilenanfang abgebildet. Vor dem 1. Textzeichen wird der Schirm ab Schreibmarke gelöscht, sofern die Nachricht nicht mit VPA beginnt.

Wird bei der Ausgabe das Bildschirmende erreicht, so wird am Schirmanfang fortgesetzt. Diese Fortsetzung ist in jedem Fall ungeschützt. Die Bildschirmüberlaufkontrolle ist unwirksam.

Die Tasten RU, EFZ, AFZ und LSP sind gesperrt.

Alle anderen Edit-Options außer OBELL, ILCASE, IGETFC werden ignoriert. (Programmierhinweise siehe Seite 99.)

=NO

Die Bedienerführung erfolgt durch das System. Geschützt gegen Überschreiben ist nur die Eingabeaufforderung durch das System bzw. das Anwendungsprogramm.

Ausgaben erfolgen ungeschützt und halbhell.

NIL-Zeichen im Ausgabebetext werden in das Ersatzzeichen umgewandelt, bei Eingabe entfernt.

Je nach Betriebsart wird am Anfang der Ausgabe der Bildschirm ab der Schreibmarke evtl. gelöscht.

OTRANS

=YES

Die Ausgabedaten sollen 'transparent' übertragen werden, d.h., sie bestehen aus beliebigen Binärzeichen (je nach Gerätecode aus 5, 7 oder 8 Bit pro Zeichen), die auf dem Übertragungsweg nicht umgewandelt werden. Ist der Übertragungsweg nicht 'potentiell transparent' generiert, wird die Ausgabe mit dem Returncode X'04' zurückgewiesen.

=NO

Die Ausgabedaten sollen 'normiert' übertragen werden, d.h., es findet eine Codeumsetzung statt.

OETB

=YES

Die Ausgabenachricht zur Datenstation wird mit dem Steuerzeichen ETB abgeschlossen.

=NO

Die Ausgabenachricht zur Datenstation wird mit dem Steuerzeichen ETX abgeschlossen.

RC	Dieser Operand ist nur für die 31-Bit-Schnittstelle zulässig.
=NEW	Der Returncode wird in Register 15 und im Standardheader (mit vollem 4-Byte-Returncode) abgelegt. Ein 4-Byte-Returncode wird nur zurückgeliefert, wenn SYSDTA von der Datensichtstation liest. In allen anderen Fällen wird nur ein 1-Byte-Returncode zurückgeliefert, unabhängig vom Wert des Returncodes.
= <u>OLD</u>	Der Returncode wird in Register 15 abgelegt. Die drei linksbündigen Byte werden ausgeblendet.
VTSUCBA	Dieser Operand wird nur ausgewertet, wenn SYSDTA der Datensichtstation zugeordnet ist. Zusätzlich ist er nur für die 31-Bit-Schnittstelle zulässig.
=addr	Adresse eines mit MF=L erzeugten VTSUCB. Bei Benutzung des Operanden VTSUCBA wird der Operand MODE und die folgenden EDIT-Optionen ignoriert. In der Parameterliste wird ihr Wert auf X'FF' gesetzt. Alle gewünschten EDIT-Optionen müssen im VTSUCB angegeben werden (siehe VTSU-Handbuch [1]). Standardwert: keine Verwendung des VTSUCB

## **Programmierhinweise für die Verwendung des Operanden MODE=EXTEND bzw. MODE=LINE mit EXTEND=YES**

(für 3270-Datenstationen siehe Anhang Seite 203)

Mit MODE=EXTEND bzw. EXTEND=YES im LINE-Mode kann mit Formaten gearbeitet werden, ohne daß hierbei eine Formatsteuerung benutzt werden muß.

- Soll der Anwender wie mit Formaten arbeiten, so muß die 1. Ausgabe mit NP beginnen, um den Bildschirm zu löschen und mit dem Text in Position (1.1) beginnen zu können.
- Mit NL wird auf den nächsten Zeilenanfang positioniert und der Rest des Bildschirmes gelöscht, mit VPAn auf den Anfang der Zeile n, wobei der Rest des Bildschirmes erhalten bleibt.
- Positionierung innerhalb einer Zeile kann nur durch Text, Zwischenräume oder NIL-Zeichen erfolgen.
- Der Text nach VPAn, NL und CHS wird geschützt und halbhell dargestellt.
- Ungeschützte Felder werden mit 'EPA Text SPA' erzeugt.  
Numerische Felder werden mit 'NUM text SPA' erzeugt.
- Mit VPAn am Ende der Nachricht kann die Schreibmarke auf den Anfang des ersten ungeschützten Feldes der Zeile n gesetzt werden, wobei der Bildschirminhalt erhalten bleibt. Beginnt in Zeile n kein ungeschütztes Feld, so wird auf den Anfang des ersten ungeschützten Feldes nach Zeile n positioniert.  
Wird am Ende der Nachricht kein VPAn angegeben, so wird die Schreibmarke in das erste ungeschützte Feld auf dem Bildschirm gesetzt.
- Folgeausgabe / Bild-Update  
Mit NP wird ein neues Bild erzeugt.

Mit VPAn am Anfang der Ausgabe wird Zeile n des Bildes geändert. Mit VPAn können eine oder mehrere Zeilen übersprungen werden. In der aktuellen Zeile wird ab der Schreibmarke ein dunkler Bereich erzeugt, der bis zum Zeilenende bzw. einem davor liegenden Feld reicht. Anschließend wird in die Zeile n positioniert. Am Ende der Änderung sollte die Schreibmarke wieder mit VPAn positioniert werden, da sonst der Bildschirm ab der Schreibmarke gelöscht würde (s.o.). Bei der Verwendung von NL innerhalb einer Änderung wird ebenfalls der Bildschirm ab Schreibmarke gelöscht. Soll dies verhindert werden, muß eine neue Zeile immer mit VPAn angesprungen werden.

**Format 2**

Operation	Operanden
WROUT	(1)

- (1) Register R1 enthält die Adresse der Operandenliste. Die Liste ist auf Wortgrenze auszurichten.

**Funktionsweise**

Beim Ablauf des Makros werden im Fall von Format 1 die spezifizierten Operanden in einer Operandentabelle abgespeichert und die Anfangsadresse dieser Tabelle in Register 1 geladen. Im Fall von Format 2 wird die im Anwendungsprogramm spezifizierte Tabelle verwendet.

## Aufbau der Operandenliste

Adressierungsmodus	Byte	Inhalt	
24-Bit-Modus	0	Output Edit Byte 1, gemäß Edit-Option-Tabelle für Ausgabe.	
	1 - 3	Adresse des auszugebenden Datensatzes (Operand satz).	
	4	Output Edit Byte 2, gemäß Edit-Option-Tabelle für Ausgabe.	
	5 - 7	Adresse, zu der im Fehlerfall verzweigt wird (Operand fehler).	
31-Bit-Modus	0 - 7	Standardheader (Aufbau siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen". Die Initialisierungswerte sind einer mit MF=L erzeugten Operandenliste zu entnehmen. Bei RC=OLD wird kein Returncode im Standardheader übergeben.	
	8 - 11	Adresse, zu der im Fehlerfall verzweigt wird (Operand fehler).	
	12 - 15	Adresse des auszugebenden Datensatzes (Operand satz).	
	16	Output Edit Byte 1	
	17	Output Edit Byte 2	} gemäß Equates im Makro, erzeugt mit MF=D
	18	reserviert (X' 00' )	
	19	Flag mit Anzeigen für Verwendung des VTSUCB und für Returncode-Verhalten; Equates im Makro erzeugt mit MF=D.	
	20 - 23	Adresse des VTSUCB	
24 - 27	reserviert (X' 00000000' )		

Die Werte für Output Edit Byte 1/2 sind der beim Makro CUPAB angegebenen Tabelle (24-Bit-Modus) oder einer mit MF=C/D erzeugten Datenliste (31-Bit-Modus) zu entnehmen.

### *Hinweis*

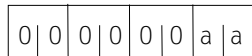
Wenn die Systemdatei SYSOUT eine katalogisierte Datei ist, so werden Sätze, die größer als 2044 byte sind, abgeschnitten (Register 15 = X'0C') und nur der erste Teil des Satzes in die Datei geschrieben. Bei EAM-Dateien werden Sätze, die größer als 2040 byte sind, ebenfalls abgeschnitten (Register 15 = X'0C').

## Rückinformation (Quittungen) und Fehleranzeigen

- Während der Makrobearbeitung enthält Register R1 die Adresse der Operandenliste.
- Über die Ausführung des Makros WROUT wird im Register R15 ein Returncode übergeben.

a) 24-Bit-Modus:

Register R15:



Bei korrekter Durchführung wird Register R15 nicht verändert.

X' aa'	Erläuterung
X' 04'	nicht behebbarer Fehler
X' 08'	Operandenfehler
X' 0C'	Abschneiden des Ausgabesatzes. Satzinhalt (ohne Satzlängengebiet) überschreitet die Größe des Ein/Ausgabepuffers: bei Datensichtstationen abhängig vom Gerätetyp und der Netzgenerierung, 2044 byte bei Ausgabe in eine katalogisierte Datei.
X' 10'	An der Datenstation wurde "BREAK" während der Ausführung des Makroaufrufs gegeben.
X' 20'	Ungültige Edit-Option; wurde vom System korrigiert.
X' 38'	Fehler im Zusammenhang mit POSIX.

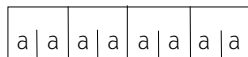
b) 31-Bit-Modus:

Bei Verwendung der 31-Bit-Schnittstelle kann man die Returncodes in

Register 15 (RC=OLD) oder in Register 15 und zusätzlich im Standardheader ablegen (RC=NEW).

### RC=OLD

Register R15:



Bei korrekter Durchführung wird Register R15 gelöscht.

Zusätzlich zu den unter a) beschriebenen Returncodes (rechtsbündig, links 0), können der Returncode X'00000000' (normale Beendigung), der Returncode X'24' (Fehler im VTSUCB) sowie die systemweit festgelegten Werte auftreten.

**RC=NEW**

Fehlermeldungen werden in Register 15 und in den Standard-Header eingetragen, siehe folgende Tabelle (Equates für MAINCODE und SUBCODE siehe Makroauflösung).

SUBCODE		MAINCODE		Bedeutung
2	1	2	1	
X' 00'	X' 00'	X' 00'	X' 00'	erfolgreiche Bearbeitung
			X' 20'	– keine Zusatzinformation
			X' 20'	– Operandenfehler, korrigiert durch TIAM/VTSU
X' 00'	X' 01'	X' 00'	X' 08'	Parameterfehler
X' 07'			X' 08'	– Operandenfehler, nicht näher spezifiziert
			X' 08'	– RESERVED-Felder nicht 0
X' 00'	X' 20'	X' 00'	X' 04'	interner Fehler
X' 02'			X' 04'	– nicht näher spezifiziert
X' 06'			X' 04'	– BCAM-Nachricht verloren
X' 00'			X' 28'	– negative Transportquittung
			X' 28'	– Probleme bei der Speicherzuordnung <sup>*)</sup>
X' 00'	X' 40'	X' 00'	X' 0C'	keine Klassenreaktion
			X' 10'	– Abschneiden des Ausgabesatzes
			X' 10'	– BREAK während der Ausführung
X' 01'	X' 80'	X' 00'	X' 04'	interner Engpaß
			X' 04'	– BCAM-Engpaß
X' 09'	X' 80'	X' 00'	X' 38'	Fehler im Zusammenhang mit POSIX
X' 0A'	X' 40'	X' 00'	X' 38'	– Ein/Ausgabe-Serialisierungsfehler
			X' 38'	– Wenn die LOGON-Task im System-Modus ist, sind keine Ein/Ausgaben von mit fork() erzeugten Prozessen möglich
aus VTSUCB	aus VTSUCB	aus VTSUCB	X' 24'	VTSU hat Fehler erkannt (für MAINCODE 1 siehe Fehlerinformation im VTSUCB-Header)

<sup>\*)</sup> SYSOUT wird einer Datei mit einer Sekundärzuweisung =0 zugeordnet und bei der Primärzuweisung wird der Speicher vollständig belegt.

## Makroauflösung

```

WROUTP  WROUT MF=C,PARMOD=31
*
*          WROUT PARAMETER BLOCK
*
WROUTP  FHDR MF=(C,*NO-NAMES)
*
WROUTP  DS    A      0  INTERFACE IDENTIFIER
        DS    A      4  GENERAL RETURN CODE
*
        MAIN RETURN CODES AS IN FHDR, ADDITIONAL VALUES:
CUWURE  EQU  4          UNRECOVERABLE ERROR
CUWOPE  EQU  8          OPERAND ERROR
CUWCUTE EQU 12          MESSAGE IS CUT
CUWBRE  EQU 16          BREAK
CUWOPEC EQU 32          OPERAND ERROR, CORRECTED
CUWVTE  EQU 36          VTSU ERROR
CUWSATE EQU 40          SPACE SATURATION ERROR
CUWPOSE EQU 56          ERROR IN POSIX CONTEXT
*
        SUB1 RETURN CODES AS IN FHDR
*
        SUB2 RETURN CODES:
CUWRCNS EQU  0          RC_S2_NOT_SPECIFIED
CUWBCSH EQU  1          BCAM SHORTAGE
CUWBCLL EQU  2          BCAM LETTER LOST
CUWNACK EQU  6          NEGATIVE TACK
CUWRFNZ EQU  7          RESERVED FIELDS NOT ZERO
CUWLOCE EQU  9          SERIALIZATION LOCK ERROR
CUWFTIO EQU 10          NO FORKED TASK IS POSSIBLE
*
CUWERRW DC   A(0)          ERROR RETURN ADDRESS
CUWMSGW DC   A(0)          OUTPUT MESSAGE ADDRESS
CUWEDIT1 DC  AL1(0)       EDIT OPTION 1
CUWEDIT2 DC  AL1(0)       EDIT OPTION 2
CUWRESO1 DC  AL1(0)       RESERVED
CUWFB1  DC   AL1(0)       FLAG BYTE 1
CUWVTSI EQU  X'80'        - VTSU INDICATOR
CUWRCIND EQU X'40'        - RC INDICATOR
CUWVTSU DC   A(0)          VTSUCB ADDRESS
CUWRTIA DC   F'0'         RES_FOR_TIAM
*
CUWL@WRO EQU *--CUWHEAD   WROUT PARAMETER BLOCK LENGTH
*
*          OUTPUT EDIT OPTION BYTE 1 - NAMES FOR BIT SETTINGS
*
CWR1CODE EQU  1          OTRSUP : CODE TRANSLATION
CWR1LNET EQU  2          OLINEND: LINE END TREATMENT
CWR1MBT1 EQU  4          MODE BIT 1
CWR1RSET EQU  8          OMANUAL: RESET FROM PAPER TAPE
CWR1HOM  EQU 16          OHOM  : HOMOGENEOUS OUTPUT
CWR1PTPE EQU 32          OPTAPE : PAPER TAPE CONTROL
CWR1MBT2 EQU 64          MODE BIT 2
CWR1HARD EQU 128         OHCOPY : HARDCOPY
CWR1MSK  EQU 68          MODE=  MODE MASK
CWR1COMP EQU  0          COMP  : COMPATIBLE MODE
CWR1LINE EQU  4          LINE  : LINE MODE
CWR1FORM EQU 64          FORM  : FORMAT MODE
CWR1PHYS EQU 68          PHYS  : PHYSICAL MODE
*
*          OUTPUT EDIT OPTION BYTE 2 - NAMES FOR BIT SETTINGS

```



```

*
CWR2HDR EQU 1          OHDR   : HEADER PRESENT
CWR2NOLC EQU 2         ONOLOGC: NO LOGIC CHARS INTERPRET.
CWR2EXT EQU 4          EXTEND  : EXTENDED LINE OUTPUT
CWR2INFO EQU 8         OINFO   : INFORMATIVE MESSAGE
CWR2POSN EQU 32        ONOPOSN: NO POSITIONING
CWR2TRAN EQU 32        OTRANS  : TRANSPARENT MODE
CWR2BEL EQU 64         OBELL   : ACCOUSTIC ALARM
CWR2ETB EQU 128        OETB   : ETB INSTEAD OF ETX
*
      END

```

*Beispiel 1*

```

WROUT1  START
        BALR 10,0
        USING *,10
        WROUT MELDUNG, FEHLER
        *,@DCEO      952      900503      53531004
        TERM
        *,VERSION 010
FEHLER  TERMD
        *,VERSION 010
MELDUNG DC Y(ENDE-MELDUNG)      Satzlaenge
        DS L2                    reserviert
        DC X'01'                 Druckvorschubzeichen (ignoriert)
        DC 'BEISPIEL WROUT'      Text
ENDE    EQU *
        END

```

```

/START-PROG $ASSEMBH
% BLS0500 PROGRAM 'ASSEMBH', VERSION '1.1A00' OF '1992-04-30' LOADED
% BLS0552 COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1990. ALL RIGHTS
% RESERVED
% ASS6010 V 1.1A00 OF BS2000 ASSEMBH- READY
//COMPILE SOURCE=*L(L=LIB.EX.MANUAL,E=WROUT1),MAC-LIB=$XXXX.LIB.M.SYSLIB.V11.0T10,
MOD-LIB=LIB.EX.MANUAL(E=WROUT1),LIST=PAR(OUT=*L(L=LIB.EX.MANUAL,E=WROUT1)),
TEST-SUP=YES
% ASS6011 ASSEMBLY TIME: 1275 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
% ASS6006 LISTING GENERATOR TIME: 801 MSEC
//END
% ASS6012 END OF ASSEMBH-
/START-PROG *MOD(L=LIB.EX.MANUAL,E=WROUT1)
% BLS0517 MODULE 'WROUT1' LOADED
BEISPIEL WROUT
/MOD-JOB-OPT LOG=P(L=NO)

```

*Beispiel 2a*

Bei diesem Beispiel wird der VTSUCB nicht verwendet.

```

WROUTT  START 0
*
        BALR 10,0
        USING *,10
*
        WROUT MELD1,FEHLER,PARMOD=31,MODE=LINE,OBELL=Y
*
        TERM
*
FEHLER  TERM
*
MELD1   DC      Y(EMELD1-MELD1)
        DS      3X
        DC      C' WROUT-Ausgabe'
EMELD1  EQU     *
*
        END

```

*Beispiel 2b*

Bei diesem Beispiel wird der VTSUCB verwendet.

```

WROUTT  START 0
*
        BALR 10,0
        USING *,10
*
        WROUT MELD1,FEHLER,PARMOD=31,VTSUCBA=VTSUPAR
*
        TERM
*
FEHLER  TERM
*
VTSUPAR VTSUCB MODE=LINE,BELL=YES
*
MELD1   DC      Y(EMELD1-MELD1)
        DS      3X
        DC      C' WROUT-Ausgabe'
EMELD1  EQU     *
*
        END

```

*Beispiel 3*

## Verwendung des Formats 2

```
WROUT2  START 0
*
        BALR 10,0
        USING *,10
*
        LA   1,PARAM
        WROUT (1),PARMOD=31
*
        TERM
*
FEHLER  TERM
*
PARAM   WROUT MELD1,FEHLER,MF=L,PARMOD=31,MODE=LINE,OBELL=Y
*
MELD1   DC   Y(EMELD1-MELD1)
        DS   3X
        DC   C'  WROUT-AUSGABE '
EMELD1  EQU   *
*
        END
```

## 3.7 Kombinierte Aus- und Eingabe: WRTRD

Der Makro WRTRD kann nur bei der Zugriffsmethode TIAM verwendet werden. WRTRD sendet eine Nachricht zur Datenstation und liest unmittelbar anschließend eine Nachricht von der Datenstation. Außer der zur Datenstation hin übertragenen Nachricht erscheint dabei kein weiteres Eingabeaufforderungszeichen.

Der Makro CUPAB generiert eine DSECT für die Operandentabelle des WRTRD für den Aufruf im 24-Bit-Adressierungsmodus.

Der Makroaufruf WRTRD akzeptiert den MF-Operanden (siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen").

24-Bit-Schnittstelle oder die 31-Bit-Schnittstelle generiert werden.

Für den 31-Bit-Adressierungsmodus ist folgendes zu beachten:

- Am Anfang der Operandenliste wird ein Standardheader (siehe Handbuch "Makroaufrufe an den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen") generiert. Der Anwender muß dafür sorgen, daß dieser Header korrekt initialisiert ist. Bei Verwendung von dynamisch erzeugten Operandenlisten (Zugriff über CSECT/DSECT) werden für den Standardheader keine symbolischen Namen und Equates erzeugt. Deshalb müssen bei dynamischer Versorgung der Operandenliste die Initialisierungswerte für den Standardheader aus einer mit MF=L erzeugten Operandenliste übernommen werden.
- Eine CSECT/DSECT wird mit MF=C/D bzw. MF=(C,p)/(D,p) generiert.  
p = Präfix (max. 3 Zeichen, ein längeres Präfix wird auf 3 Zeichen gekürzt);  
Voreinstellung p = CUB. Das Präfix verändert nur die Feldnamen, nicht die symbolischen Namen bei den Equates.

### *Hinweis zur Kompatibilität*

Der Makroaufruf WRTRD ab TIAM-Version V10.0A (gekoppelt an BS2000 V10.0A) erzeugt die Headerversion 2 im Standardheader. Damit ist er in BS2000-Versionen kleiner V10.0 nicht ablauffähig und verzweigt dort auf den Fehlerausgang. Soll ein Programm in mehreren BS2000-Versionen ausführbar sein, so muß das Programm in der niedrigsten Version erstellt sein (es wird nur Aufwärtskompatibilität garantiert).

**Format 1**

Operation	Operanden
WRTRD	<p>satz1,[edit1],satz2,[edit2],[länge],fehler</p> $\left[ \begin{array}{l} 24[\text{,MF}=\left\{ \begin{array}{l} \text{L} \\ \text{(E,...)} \end{array} \right\}] \\ \\ \left[ \text{,PARMOD}=\left\{ \begin{array}{l} \text{C} \\ \text{(C,p)} \\ \text{D} \\ \text{(D,p)} \\ \text{L} \\ \text{(E,...)} \end{array} \right\} \right] \\ 31[\text{,MF}=\left\{ \begin{array}{l} \text{L} \\ \text{(E,...)} \end{array} \right\}] \end{array} \right]$ $\left[ \begin{array}{l} \text{,MODE=COMP}[\text{,OTRSUP}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,OLINEND}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,OHCOPY}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]} \\ \\ [\text{,OHDR}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,ITRSUP}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,ILINEND}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]} \\ \\ [\text{,ILCASE}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,IHDR}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,IGETBS}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]} \\ \\ \left[ \text{,MODE=LINE}[\text{,OHCOPY}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,OHOM}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,OBELL}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]} \right] \\ \\ [\text{,ONOLOGC}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,ONOPSN}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,ILCASE}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]} \\ \\ [\text{,EXTEND}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,IGETFC}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,IGETBS}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]} \\ \\ [\text{,ICFD}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]}[\text{,IGETIC}=\left\{ \frac{\text{N[O]}}{\text{Y[ES]}} \right\}] \text{]} \end{array} \right]$

Operation	Operanden
WRTRD (Forts.)	$\left[ \begin{array}{l} \text{,MODE=FORM } [ \text{,IGETBS= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] [ \text{,ILCASE= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] \\ \text{,MODE=PHYS } [ \text{,OHDR= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] [ \text{,OTRANS= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] \\ [ \text{,OETB= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] [ \text{,ILCASE= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] \\ [ \text{,IHDR= } \left\{ \begin{array}{l} \text{Y[ES]} \\ \text{N[O]} \end{array} \right\} ] [ \text{,IGETBS= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] \\ [ \text{,ITRSUP= } \left\{ \begin{array}{l} \text{N[O]} \\ \text{Y[ES]} \end{array} \right\} ] \end{array} \right]$ $[ \text{,RC= } \left\{ \begin{array}{l} \text{OLD} \\ \text{NEW} \end{array} \right\} ]$ $[ \text{,VTSUCBA=addr} ]$ $[ \text{,TIMER=wert} ]$

Die Operanden RC, VTSUCBA und TIMER werden nur bei der 31-Bit-Schnittstelle unterstützt.

satz1

Symbolische Adresse des auszugebenden Datensatzes. Der Satz beginnt mit dem Satzlängenfeld, gefolgt von einem (beliebigen) Zeichen und der auszugebenden Nachricht.

Satzaufbau:

- |      |      |  |
|------|------|--|
| Byte | 0-1: | Länge des Satzes + 4 byte Satzlängenfeld, die Länge muß größer als 5 sein. |
|      | 2-3: | reserviert   |
|      | 4:   | beliebiges Zeichen; wird weder übertragen noch ausgewertet                 |
|      | 5-n: | Datensatz  |

*Beispiel*

```

satz      DC   Y (satzend-satz)   reservierte Byte
          DS   CL2
          DC   X'00'
          DC   C'datensatz'      zu übertragender Datensatz
satzend   EQU  *

```

edit1

Dieser Operand gibt die Aufbereitung für den zur Datenstation zu übertragenden Satz an. Durch Direktangabe (X'xx') kann hier nur das 1. Edit-Byte für Ausgabe auf die beim CUPAB-Makro beschriebene Bedeutung gesetzt werden. Dieser Operand ist nicht erforderlich, wenn Standardfunktionen (alle Edit-Bits=0) verwendet werden, bei einer MODE-Angabe oder bei Nutzung des VTSU-Control-Blocks.

satz2

Symbolische Adresse eines Feldes, in das der von der Datensichtstation eingelesene Datensatz übertragen wird. Der Datensatz wird als Satz variabler Länge eingelesen (die ersten vier Byte enthalten die Satzlänge).

## Satzaufbau:

```

Byte  0-1:  Länge des Satzes + 4 byte Satzlängenfeld
       2-3:  reserviert
       4-n:  Datensatz

```

*Beispiel*

```

SATZ2    DS   0CL74
LAENGE   DS   CL2
RESERV   DS   CL2
DATEN    DS   CL70

```

edit2

Gibt die Aufbereitung für den zum Anwendungsprogramm zu übertragenden Satz an. Durch Direktangabe (X'xx') kann hier nur das erste Edit-Byte für Eingabe auf die beim CUPAB-Makro beschriebene Bedeutung gesetzt werden. Dieser Operand ist nicht erforderlich, wenn Standardfunktionen (alle Edit-Bits=0) verwendet werden, bei einer MODE-Angabe oder bei Nutzung des VTSU-Control-Blocks.

*Hinweis*

Die Operanden edit1 und edit2 werden nur noch aus Gründen der Kompatibilität unterstützt. Die Edit-Optionen sollten über MODE-Angaben (siehe unten) oder den VTSU-Control-Block (VTSUCBA) gesteuert werden.

länge	Länge des unter satz2 angegebenen Feldes (einschließlich 4 byte für das Satzlängenfeld); $5 \leq \text{länge} \leq 32767$ . Bei fehlender Angabe wird das Längenattribut des angegebenen Feldes angenommen.
fehler	Symbolische Adresse einer Fehleroutine im Anwendungsprogramm, zu der im Fehlerfall verzweigt wird.  Im Fehlerfall enthält Register R14 die Adresse des Befehls, der dem WRTRD-Aufruf folgt. Der Fehlercode wird im Register 15 übergeben. 31-Bit-Schnittstelle: Bei Angabe fehler = 0 (Adresse X'00..0') wird das Programm mit dem Befehl fortgesetzt, der dem WRTRD-Aufruf folgt.
PARMOD	Steuert die Makroauflösung. Es wird entweder die 24-Bit-Schnittstelle oder die 31-Bit-Schnittstelle generiert. Wenn PARMOD nicht angegeben wird, erfolgt die Makroauflösung entsprechend der Angabe für den Makro GPARMOD (siehe Handbuch "Makroaufrufe an den Ablaufteil" [3]) oder der Voreinstellung für Assembler (= 24-Bit-Adressierung).
=24	Die 24-Bit-Schnittstelle wird generiert. Datenlisten und Befehle benutzen 24-Bit-Adressen (Adreßraum $\leq 16\text{MB}$ ).
=31	Die 31-Bit-Schnittstelle wird generiert. Datenlisten und Befehle benutzen 31-Bit-Adressen (Adreßraum $\leq 2\text{GB}$ ). Datenlisten beginnen mit dem Standardheader.
MF	
=L	Generiert eine Parameterliste, Felder werden gemäß Parameterangaben versorgt.
=(E,...)	Generiert den Befehlsteil.
=C	Generiert eine CSECT.
=(C,p)	Generiert eine CSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = CUB.
=D	Generiert eine DSECT.



=(D,p) Generiert eine DSECT, Vergabe eines maximal 3 byte langen Präfix p; Voreinstellung p = CUB.

### *Hinweis*

Die MODE-Angaben zusammen mit den Edit-Optionen werden nur noch aus Kompatibilitätsgründen unterstützt. Sie werden jetzt im VTSU-Control-Block (VTSUCB) zusammengefasst. Künftige Erweiterungen erfolgen nur noch im VTSUCB.

### MODE

=COMP Kompatibler Betriebsmodus. Steuerzeichen in der Ausgabenachricht sind zulässig, werden aber vom System nicht auf Richtigkeit geprüft. Diese Betriebsweise ist kompatibel zu früheren Versionen des Betriebssystems.  
Für die Geräte 8160, 8162, 9749, 975x, 9763, 3270 und X.29-Datenstationen wird dieser Modus wie MODE=LINE behandelt. Die Edit-Options OLINEND und ILINEND werden ignoriert. Die Edit-Options OTRSUP, OHDR, ITRSUP und IHDR werden abgewiesen (RC: X'08').

=LINE Die aktuelle Datenstation soll als logische Zeilen- bzw. Seiten-Datenstation behandelt werden. Die Nachricht kann durch logische Steuerzeichen strukturiert werden (siehe Makroaufruf VTCSET).  
Für die Ausgabe sind weitere Steuerzeichen unzulässig und werden in ein vom Anwender definiertes Ersatzzeichen umgewandelt (siehe Kommando MODIFY-TERMINAL-OPTIONS SUBSTITUTE-CHARACTER=).  
Ist SYSOUT keine Datenstation, so werden nur die logischen Steuerzeichen NL und NP ausgewertet z.B. bei Ausgaben auf Drucker im Stapelbetrieb.  
Bei der Eingabe wird der gerätespezifische Nachrichtenkopf nicht mitgeliefert.

=FORM Format-Modus. Das Anwendungsprogramm arbeitet mit der Software-Komponente FHS bzw. Formatsteuerung, die auch die datenstationsgerechte Aufbereitung der Ausgabenachricht vornimmt.

=PHYS Die Nachrichten sollen physikalisch, d.h. ohne Aufbereitung durch das System an die Datenstation ausgegeben bzw. von dort eingelesen werden. Damit können spezielle Gerätefunktionen angesprochen werden, für die der LINE- oder FORM-Modus nicht ausreicht.  
Wird keine der zulässigen Edit-Options angegeben, stellt das System der Ausgabenachricht einen gerätespezifischen Standard-Nachrichtenkopf voran; Aus der Eingabenachricht wird der geräte-

spezifische Nachrichtenkopf nicht entfernt. Kleinbuchstaben werden in Großbuchstaben umgewandelt und evtl. wird die Backspace-Funktion ausgeführt.

#### OTRSUP

=YES

Die Übersetzung der Nachricht wird unterdrückt. Das Programm muß in diesem Fall die Nachricht im Gerätecode liefern.

=NO

Die Übersetzung der Nachricht von EBCDIC in den Gerätecode wird nicht unterdrückt, d.h. das Programm liefert die Nachricht in EBCDI-Code. Im System wird die Nachricht in den Gerätecode übersetzt.

#### OLINEND

=YES

Die Ausgabe der Nachricht zur Datenstation erfolgt ohne vom System bereitgestellte Steuerzeichen für Wagenrücklauf/Zeilenvorschub. Die Steuerung muß vom Anwendungsprogramm vorgenommen werden.

=NO

Jede Ausgabenachricht beginnt bei einer Datenstation immer mit einer neuen Zeile. Die dafür notwendigen Steuerzeichen werden vom System der Nachricht vorangestellt bzw. bei Erreichen des physikalischen Zeilenendes in die Nachricht eingeschoben, wenn es der betreffende Datenstationstyp erforderlich macht.

#### OHCOPY

=YES

Die Ausgabenachricht für eine Datensichtstation wird gleichzeitig über ein dort angeschlossenes Hardcopy-Gerät (Drucker) ausgedruckt. Das Hardcopy-Gerät muß generiert werden oder mittels MODIFY-TERMINAL-OPTIONS zugewiesen werden.

##### **Bei 3270:**

Der Inhalt des gesamten Bildschirms wird über das Hardcopy-Gerät ausgedruckt. Dadurch werden eventuell auch frühere Aus- und Eingaben ausgedruckt. Bei mehreren direkt aufeinanderfolgenden Ausgaben wird die Hardcopy-Funktion nur bei der letzten Ausgabe ausgelöst. Der Hardcopy-Abdruck erfolgt nur dann, wenn für die Datensichtstation beim Verbindungsaufbau ein Hardcopy-Gerät generiert wurde.

Wurde EXTEND=YES/MODE=EXTEND angegeben, erfolgt kein Hardcopy-Abdruck.

Wird OHCOPY=YES verwendet und enthält die Nachricht das logische Steuerzeichen SPA, EPA, CHS oder NUM, so wird nicht die gesamte Nachricht, sondern nur der letzte ungeschützte Teil der Nachricht abgedruckt.

Wird gleichzeitig OVERFLOW-CONTROL=NO (MODIFY-TERMINAL-OPTIONS-Kommando) verwendet, kann es vorkommen, daß nur ein Teil der Ausgabe auf dem Hardcopy-Gerät wiedergegeben wird.

=NO

Es findet keine Druckausgabe über ein Hardcopy-Gerät statt.

OHDR

=YES

Die Nachricht enthält einen anwenderindividuellen Nachrichtenkopf (Nachrichtenkopf ist im US-ASCII-Code anzugeben), den das System dem Ausgabebetext voranstellt. Die Länge des Nachrichtenkopfs +1 muß im ersten Byte der Nachricht binär angegeben werden.

**Bei 3270:**

Der Nachrichtenkopf wird im EBCDI-Code angegeben und besteht aus dem CMD-Byte und dem WCC-Byte. Vor diesem Nachrichtenkopf muß ein Byte mit dem Inhalt X'01' gesetzt werden (Länge des TRANSDATA-Nachrichtenkopfes + 1).

*Hinweis*

Bei Ausgabe auf die Datensichtstationen 8160, 975x, 9763 und hieran lokal angeschlossene Drucker ist zu beachten, daß das System (MODE=LINE bzw. COMP) oder FHS mit MODE=FORM keinen Nachrichtenkopf verwendet (PARAMO, PARAM1), sondern mit Parameterangaben (PAG) arbeitet. Die Unterschiede zwischen diesen beiden Arbeitsweisen finden Sie beschrieben in den Handbüchern für Datensichtstationen bzw. Drucker.

=NO

Den Nachrichtenkopf (im US-ASCII-Code) stellt das System dem Ausgabebetext voran.

OHOM

=YES

Nur bei Datensichtstationen 816x, 9749, 975x, 9763 und 3270 anzuwenden.

Die Nachricht soll unstrukturiert, homogen ausgegeben werden, d.h. die gesamte Nachricht wird als eine Ausgabeeinheit betrachtet. Die Nachrichtenlänge ist durch die Größe des Ausgabepuffers im System beschränkt.

	<p>Wirkung bei Datenstationen 816x, 975x, 9763 und 3270 bei Betriebsart 1:</p> <p>Durch Modifikation eines Zeichens einer Ausgabenachricht kann die gesamte Nachricht wieder zurückübertragen werden, sofern diese nicht durch logische Anzeigesteuerzeichen explizit strukturiert wird.</p>
= <u>NO</u>	<p>Die Nachricht soll strukturiert, nicht homogen ausgegeben werden, d.h. als Ausgabeeinheit wird eine logische Zeile betrachtet. Die Nachrichtenlänge ist nicht beschränkt, falls die logischen Zeilen nicht länger als 255 Zeichen sind.</p> <p>Wirkung bei Datenstationen 816x, 975x, 9763 und 3270 bei Betriebsart 1:</p> <p>Einzelne logische Zeilen können getrennt modifiziert und damit gezielt zurückübertragen werden.</p>
ONOPSN	
=YES	Die Ausgabenachricht beginnt am Anfang der aktuellen Zeile (gilt nur für Schreibstationen).
= <u>NO</u>	Die Ausgabenachricht beginnt am Anfang der nächsten Zeile.
OBELL	
=YES	Bei der Ausgabe ertönt am Ende der Nachricht ein akustisches Signal (nur bei den Datenstationen 9749, 975x, 9763 816x und 3270 mit einem speziellen Gerätezusatz).
= <u>NO</u>	Bei der Ausgabe ertönt kein akustisches Signal.
ONOLOGC	
=YES	Logische Steuerzeichen werden nicht ausgewertet. Alle Zeichen, die im EBCDI-Code kleiner als X'40' sind, werden durch SUB ersetzt. Nur Zeichen größer oder gleich X'40' werden durchgelassen.
= <u>NO</u>	Alle logischen Steuerzeichen werden ausgewertet und spezielle physikalische Steuerzeichen werden durchgelassen (siehe VTCSET z.B. ESC, DC4). Andere Zeichen kleiner X'40' werden durch SUB ersetzt. Zeichen größer oder gleich X'40' werden durchgelassen.
EXTEND	
=YES	<p>(Nur für Datenstationen 9749, 975x, 9763, 816x und 3270)</p> <p>Diese Angabe unterstützt die Verwendung von geschützten und ungeschützten Feldern mit Hilfe der logischen Steuerzeichen EPA, DAR, NUM und SPA (siehe Makro VTCSET).</p>

Die Ausgabe des Textes erfolgt standardmäßig geschützt und halbhell. Die Nachricht kann durch logische Steuerzeichen strukturiert werden (siehe VTCSET). Bei 3270-Datenstationen ist zu beachten, daß die logischen Steuerzeichen Platz auf dem Bildschirm beanspruchen. Mehrere logische Steuerzeichen hintereinander benötigen aber nur einen Platz. Bereiche, in die der Anwender eingeben kann, werden mit EPA, DAR oder NUM begonnen und mit SPA beendet.

Bei Ein- und Ausgabe wird NIL als erlaubtes Zeichen behandelt, es wird vom Programm zur Datenstation und umgekehrt geschickt. Bei 3270-Datenstationen ist zu beachten, daß NIL-Zeichen nicht zur DVA übertragen werden. VTSU-B ergänzt Felder, die bei der Eingabe verkürzt zurückkommen, durch NIL-Zeichen auf ihre ursprüngliche Länge. Dadurch bekommt der Anwender die Felder immer in der Ausgabelänge zurück.

Der Anfang einer Ausgabenachricht wird an dem der Schreibmarke folgenden nächsten Zeilenanfang abgebildet. Vor dem 1. Textzeichen wird der Schirm ab Schreibmarke gelöscht, wenn die Nachricht nicht mit VPA beginnt.

Wird bei der Ausgabe das Bildschirmende erreicht, so wird am Schirmanfang fortgesetzt. Diese Fortsetzung ist in jedem Fall ungeschützt. Die Bildschirmüberlaufkontrolle ist unwirksam.

Die Tasten RU, EFZ, AFZ und LSP sind gesperrt.

Alle anderen Edit-Options außer OBELL, ILCASE, IGETFC werden ignoriert.

Wird in einer Eingabenachricht das Steuerzeichen NL erkannt, so wird die Bearbeitung fortgesetzt und der Returncode X'2C' geliefert (RC:X'2C').

=NO

Die Bedienerführung erfolgt durch das System. Geschützt gegen Überschreiben ist nur die Eingabeaufforderung durch das System bzw. das Anwendungsprogramm.

Ausgaben sind ungeschützt und halbhell.

NIL-Zeichen im Ausgabebetext werden in das Ersatzzeichen umgewandelt, bei Eingabe entfernt.

Je nach Betriebsart wird am Anfang der Ausgabe der Bildschirm ab der Schreibmarke evtl. gelöscht.

## OETB

=YES Die Ausgabenachricht zur Datenstation wird mit dem Steuerzeichen ETB abgeschlossen.

=NO Die Ausgabenachricht zur Datenstation wird mit dem Steuerzeichen ETX abgeschlossen.

## OTRANS

=YES Die Ausgabedaten sollen 'transparent' übertragen werden, d.h., sie bestehen aus beliebigen Binärzeichen (je nach Gerätecode aus 5, 7 oder 8 Bit pro Zeichen), die auf dem Übertragungsweg nicht umgewandelt werden. Ist der Übertragungsweg nicht 'potentiell transparent' generiert, wird die Ausgabe mit dem Returncode X'04' zurückgewiesen.

=NO Die Ausgabedaten sollen 'normiert' übertragen werden, d.h., es findet eine Codeumsetzung statt.

## ITRSUP

=YES Die Übersetzung von Gerätecode in EBCDIC wird unterdrückt. Das Anwendungsprogramm erhält also die Nachricht im Gerätecode.

=NO Die Übersetzung von Gerätecode in EBCDIC wird nicht unterdrückt. Das Anwendungsprogramm erhält die Nachricht im EBCDI-Code.

Ausnahme

Der Nachrichtenkopf bei den Datensichtstationen 816x, 9749, 975x und 9763 wird immer im Gerätecode geliefert.

## ILINEND

=YES Die Zeichen Wagenrücklauf/Zeilenvorschub werden in das Anwendungsprogramm übertragen.

=NO Die Wagenrücklauf-/Zeilenvorschubzeichen werden dem Anwendungsprogramm nicht übergeben.

## ILCASE

=YES Dem Anwendungsprogramm werden auch Kleinbuchstaben übergeben.

=NO Alle Kleinbuchstaben werden dem Anwendungsprogramm als Großbuchstaben übergeben.

## IHDR

- =YES Der gesamte Nachrichtenvorspann wird an das Anwendungsprogramm übergeben.  
Bei 3270-Datenstationen besteht der Nachrichtenvorspann aus dem Code der Sendetaste (AID-Byte) und der zwei byte langen Schreibmarkenposition.
- =NO Der Nachrichtenvorspann wird nicht an das Anwendungsprogramm übergeben.

## IGETBS

- =YES Die Zeichen "Underline" (X'6D') werden dem Anwendungsprogramm übergeben. Es findet keine Auswertung durch das System statt  
(nur für Schreibstationen).
- =NO "Underline" wird nicht in das Anwendungsprogramm übertragen. Stattdessen wird vom System die Korrekturfunktion durchgeführt (nur für Schreibstationen).

## IGETFC

- =YES Das 1. Byte des Datensatzes im Einlesebereich soll den normierten Funktionstastencode enthalten. Dieser identifiziert die zur Auslösung der Datenübertragung an der Datenstation betätigte Taste. Eine Tabelle der normierten Funktionstastencodes finden Sie im Anhang (siehe Seite 201). Sie enthält auch Funktionstastencodes für 3270-Datenstationen.
- =NO Es soll kein Funktionstastencode übergeben werden.

## IGETIC

- =YES Die Daten werden vom Ausweisleser eingegeben. Die Eingabedaten können aus der Ausweisinformation oder dem Kurztelegramm K14 bestehen. Diese Angabe ist nur bei Datenstationen 9749, 975x, 9763, 816x und 3270 mit definiertem Ausweisleser möglich (siehe Makroaufruf TSTAT TYPE=TCHAR).  
Im Unterschied zu den TRANSDATA-Geräten können bei 3270-Datenstationen jederzeit Daten von einem definierten Ausweisleser eingegeben werden. Wenn Eingaben vom Ausweisleser angefordert werden, wird jede andere Eingabe in K14 umgewandelt. Diese Edit-Option wird ignoriert (IGETIC auf NO zurückgesetzt), wenn kein Ausweisleser angeschlossen ist.

*Hinweis*

Der Operand IGETIC wird ignoriert, wenn gleichzeitig ICFD=YES angegeben wird, oder wenn kein Ausweisleser angeschlossen ist. Die Eingabequelle bleibt unverändert.

=NO

Die Eingabequelle soll nicht geändert werden.

ICFD

=YES

Die Eingabedaten sind vertraulich und sollen an der Datenstation unsichtbar bleiben. Dies erfolgt durch Dunkelsteuerung bzw. Löschen des Bildschirms (wodurch das Bildschirmformat auf 24x80 zurückgesetzt wird) oder durch Überschreiben der Eingabezeile an Schreibstationen.

=NO

Es sollen keine Vorkehrungen zum Schutz vertraulicher Daten getroffen werden.

RC

Dieser Operand ist nur für die 31-Bit Schnittstelle zulässig.

=NEW

Der Returncode wird in Register 15 und im Standardheader (mit vollem 4-Byte-Returncode) abgelegt. Ein 4-Byte-Returncode wird nur zurückgeliefert, wenn SYSDTA von der Datensichtstation liest. In allen anderen Fällen wird nur ein 1-Byte-Returncode zurückgeliefert, unabhängig vom Wert des Returncodes.

=OLD

Der Returncode wird in Register 15 abgelegt. Die drei linksbündigen Byte werden ausgeblendet.

VTSUCBA

=addr

Adresse eines mit MF=L erzeugten VTSUCB. Dieser Operand ist nur für die 31-Bit Schnittstelle zulässig. Bei Benutzung des Operanden VTSUCBA wird der Operand MODE und die folgenden EDIT-Optionen ignoriert. In der Parameterliste wird ihr Wert auf X'FF' gesetzt. Alle gewünschten EDIT-Optionen müssen im VTSUCB angegeben.

Standardwert: keine Verwendung des VTSUCB

TIMER

Dieser Operand ist nur für die 31-Bit Schnittstelle zulässig.

=wert

In diesem Feld kann der Benutzer angeben, ob ein TIMER benutzt werden soll oder nicht (UNLIMITED). Der Timer legt die maximale Wartezeit für die Eingabe fest. Erfolgt innerhalb der festgelegten Wartezeit keine Eingabe, wird ein Rückgabewert zurückgegeben. Der Wert wird in Sekunden angegeben.

Standardwert: UNLIMITED (es wird kein Timer benutzt).



Programmierhinweise für die Verwendung des Operanden MODE=EXTEND bzw. MODE=LINE mit EXTEND=YES finden Sie bei der Beschreibung der Makros WROUT (siehe Seite 99).

## Format 2

Operation	Operanden
WRTRD	(1)

(1) Register R1 enthält die Adresse der Operandenliste. Die Liste ist auf Wortgrenze auszurichten.

## Funktionsweise

Beim Ablauf des Makros werden im Fall von Format 1 die spezifizierten Operanden in einer Operandentabelle abgespeichert und die Anfangsadresse dieser Tabelle in Register 1 geladen. Im Fall von Format 2 wird die im Anwendungsprogramm spezifizierte Tabelle verwendet.

## Aufbau der Operandenliste

Adressierungsmodus	Byte	Inhalt
24-Bit-Modus	0	Output Edit Byte 1, gemäß Edit-Option-Tabelle für Ausgabe.
	1 - 3	Adresse des auszugebenden Datensatzes (Operand satz1).
	4	Input Edit Byte 1, gemäß Edit-Option-Tabelle für Eingabe.
	5 - 7	Adresse des Feldes, in das der einzulesende Datensatz übertragen wird (Operand satz2).
	8	Output Edit Byte 2, gemäß Edit-Option-Tabelle für Ausgabe.
	9	Input Edit Byte 2, gemäß Edit-Option-Tabelle für Eingabe.
	10 - 11	max. Länge des einzulesenden Datensatzes (Operand länge).
	12 - 15	Adresse, zu der im Fehlerfall verzweigt wird (Operand fehler).
31-Bit-Modus	0 - 7	Standardheader (Aufbau siehe Handbuch "Makroaufrufe für den Ablaufteil", [3], Abschnitt "Aufbau der generierten Anweisungen"). Die Initialisierungswerte sind einer mit MF=L erzeugten Operandenliste zu entnehmen. Bei RC=OLD wird kein Returncode im Standardheader übergeben.
	8 - 11	Adresse zu der im Fehlerfall verzweigt wird (Operand fehler).
	12 - 15	Adresse des auszugebenden Datensatzes (Operand satz1)
	16 - 19	Adresse des Feldes, in das der einzulesende Datensatz übertragen wird (Operand satz2).
	20	Output Edit Byte 1
	21	Output Edit Byte 2
	22	Input Edit Byte 1
	23	Input Edit Byte 2
	24 - 25	max. Länge des einzulesenden Datensatzes (Operand länge).
	26	reserviert (X' 00' )
	27	Flag mit Anzeigen für Verwendung des VTSUCB und für Returncode-Verhalten; Equates im Makro erzeugt mit MF=D.
	28 - 31	Adresse des VTSUCB
	32 - 33	Werte des Timers
34 - 35	reserviert (X' 0000' )	

Die Werte für Input/Output Edit Byte 1/2 sind der beim Makro CUPAB angegebenen Tabelle (24-Bit-Modus) oder einer mit MF=C/D erzeugten Liste (31-Bit-Modus) zu entnehmen.

*Hinweis*

- Wenn die Länge des zu schreibenden Satzes (weniger 4 byte für das Längenfeld und 1 byte reserviert) die Größe des Puffers für die Datenstation überschreitet, wird der Satz verkürzt. Es wird mit dem Rückkehrcode X'10' in Register 15 zur Fehleradresse des Benutzers verzweigt.
- Wenn die Länge des zu lesenden Satzes die angegebene Länge (minus 4 byte für das Längenfeld) überschreitet, wird der Satz verkürzt und es wird mit der Rückmeldung X'0C' in Register 15 zur Fehleradresse des Benutzerprogramms verzweigt.

**Rückinformation (Quittungen) und Fehleranzeigen**

- Edit-Options (Aufbereitungsparameter), die für ein Gerät oder für die gewählte MODE-Art nicht zulässig sind, werden vom System soweit möglich korrigiert (Rückmeldung X'20').
- Während der Makrobearbeitung enthält Register R1 die Adresse der Operandenliste.
- Über die Ausführung des Makros WRTRD wird im Register R15 ein Returncode übergeben.
  - a) 24-Bit-Modus:

Register R15:

0	0	0	0	0	0	a	a
---	---	---	---	---	---	---	---

Bei korrekter Durchführung wird Register R15 nicht verändert.

X' aa'	Erläuterung
X' 04'	Nicht behebbarer Fehler.
X' 08'	Operandenfehler (fehlerhafter Operand im Programm).
X' 0C'	Satz beim Einlesen abgeschnitten. Die Länge des Satzes überschreitet die angegebene Länge. Die Eingabenachricht wird von einem MSV-Terminal mit einem Header versehen. Hat die Nachricht bereits die Länge des Systempuffers, wird durch Vorstellen dieses Headers die Nachricht abgeschnitten.
X' 10'	Ausgabesatz abgeschnitten. Die Länge des auszugebenden Satzes überschreitet die Größe des Puffers für die Datenstation. Die überschüssigen Zeichen werden nicht ausgegeben.
X' 14'	WRTRD wurde in einem Stapelauftrag aufgerufen.
X' 18'	Eingabeende (ETX) erkannt.
X' 20'	Ungültiges Edit-Option-Byte, wurde vom System korrigiert.
X' 2C'	Nur bei EDIT-Option EXTEND=YES. Die Eingabe beginnt mit Steuerzeichen NL. Bei 3270: Eingabelänge verkürzt.
X' 38'	Fehler im Zusammenhang mit POSIX.

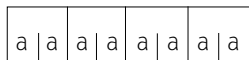
b) 31-Bit-Modus:

Bei Verwendung der 31-Bit-Schnittstelle kann man die Returncodes in

Register 15 (RC=OLD) oder in Register 15 und zusätzlich im Standardheader ablegen (RC=NEW).

**RC=OLD**

Register R15:



Bei korrekter Durchführung wird Register R15 gelöscht.

Zusätzlich, zu den unter a) angegebenen Returncodes (rechtsbündig, links 0), können der Returncode X'24' (Fehler im VTSUCB), X'34' (Timer-Ablauf) sowie die systemweit festgelegten Werte auftreten.

**RC=NEW**

Fehlermeldungen werden in Register 15 und in den Standard-Header eingetragen, siehe folgende Tabelle (Equates für MAINCODE und SUBCODE siehe Makroauflösung).

SUBCODE		MAINCODE		Bedeutung
2	1	2	1	
X' 00'	X' 00'	X' 00'	X' 00' X' 20'	erfolgreiche Bearbeitung – keine Zusatzinformation – Operandenfehler, korrigiert durch TIAM/VTSU
X' 00' X' 07' X' 08'	X' 01'	X' 00'	X' 08' X' 08' X' 08'	Parameterfehler – Operandenfehler, nicht näher spezifiziert – RESERVED-Felder nicht 0 – der Timerwert ist nicht im erlaubten Bereich von 10 bis 3600 Sekunden
X' 00' X' 02' X' 05' X' 06'	X' 20'	X' 00'	X' 04' X' 04' X' 04' X' 04'	interner Fehler – nicht näher spezifiziert – BCAM-Nachricht verloren – Eingabenachricht zu lang – negative Transportquittung
X' 00'	X' 40'	X' 00'	X' 0C' X' 10' X' 14' X' 18' X' 2C' X' 34'	keine Klassenreaktion – Abschneiden des Eingabesatzes – Abschneiden des Ausgabesatzes – BREAK im WRTRD *) – Ende der Eingabe – nur bei Edit-Optionen mit EXTEND=YES: NL erkannt – Timer-Ablauf (innerhalb der festgelegten Wartezeit erfolgte keine Eingabe)

SUBCODE		MAINCODE		Bedeutung
X' 00'	X' 00'	X' 00'	X' 14'	Fehler bei SYSFILE – WRTRD im Stapelbetrieb
X' 01'	X' 80'	X' 00'	X' 04'	interner Engpaß – BCAM-Engpaß
X' 09'	X' 80'	X' 00'	X' 38'	Fehler im Zusammenhang mit POSIX – Ein/Ausgabe-Serialisierungsfehler – Wenn die LOGON-Task im System-Modus ist, sind keine Ein/Ausgaben von mit fork() erzeugten Prozessen möglich
X' 0A'	X' 40'	X' 00'	X' 38'	
aus VTSUCB	aus VTSUCB	aus VTSUCB	X' 24'	VTSU hat Fehler erkannt (für MAINCODE 1 siehe Fehlerinformation im VTSUCB-Header)

- \*) Wird während einer Schreib-/Lese-Operation ein "BREAK" erkannt und ist RC=NEW gesetzt, wird der Returncode X'00400014' geliefert. Ist RC=OLD wird der P1-Befehlszähler auf den Anfang der Makroauflösung zurückgesetzt, so daß nach der Behandlung der Unterbrechung der Makroaufruf wiederholt wird.

## Makroauflösung

```

WRTRDP  WRTR  MF=C,PARMOD=31
*
*          WRTRD PARAMETER BLOCK
*
CUBHEAD  FHDR  MF=(C,*NO-NAMES)
CUBHEAD  DS    A          0  INTERFACE IDENTIFIER
          DS    A          4  GENERAL RETURN CODE
*
*          MAIN RETURN CODES AS IN FHDR, ADDITIONAL VALUES:
CUBURE   EQU  4          UNRECOVERABLE ERROR
CUBOPE   EQU  8          OPERAND ERROR
CUBCUTE  EQU 12          TRUNCATION DURING READ
CUBTRUN  EQU 16          TRUNCATION DURING WRITE
CUBATCH  EQU 20          WRTRD IN BATCH
CUBEOF   EQU 24          WRTRD EOF
CUBOPEC  EQU 32          OPERAND ERROR, CORRECTED
CUBVTE   EQU 36          VTSU ERROR
CUBNL    EQU 44          NL EXT INPUT
CUBTOUT  EQU 52          TIME OUT DURING READ
CUBPOSE  EQU 56          ERROR IN POSIX CONTEXT

*          SUB1 RETURN CODES AS IN FHDR
*          SUB2 RETURN CODES:
CUBRCNS  EQU  0          RC_S2_NOT_SPECIFIED
CUBBCSH  EQU  1          BCAM SHORTAGE
CUBBCLL  EQU  2          BCAM LETTER LOST
CUBMTL   EQU  5          MESSAGE TOO LONG
CUBNACK  EQU  6          NEGATIVE TACK
CUBRFNZ  EQU  7          RESERVED FIELDS NOT ZERO
CUBWT    EQU  8          WRONG TIMER
CUBLOCE  EQU  9          SERIALIZATION LOCK ERROR
CUBFTIO  EQU 10         NO FORKED TASK IS POSSIBLE
*
CUBERRW  DC    A(0)      ERROR RETURN ADDRESS
CUBMSGW  DC    A(0)      OUTPUT MESSAGE ADDRESS
CUBAREAW DC    A(0)      READ AREA ADDRESS
CUBOEDT1 DC  AL1(0)     OUTPUT EDIT OPTION 1
CUBOEDT2 DC  AL1(0)     OUTPUT EDIT OPTION 2
CUBIEDT1 DC  AL1(0)     INPUT EDIT OPTION 1
CUBIEDT2 DC  AL1(0)     INPUT EDIT OPTION 2
CUBALEN  DC    H'0'     READ AREA LENGTH
CUBRESO1 DC  AL1(0)     RESERVED
CUBFB1   DC    AL1(0)   FLAG BYTE 1
CUBVTSI  EQU  X'80'     - VTSU INDICATOR
CUBRCIND EQU  X'40'     - RC INDICATOR
CUBTIND  EQU  X'20'     - INPUT TIMER INDICATOR
CUBVTSU  DC    A(0)     VTSUCB ADDRESS
CUBTVAL  DC    H'0'     INPUT TIMER VALUE
CUBRTIA  DC    H'0'     RES_FOR_TIAM
*
CUBL@WRT EQU  *--CUBHEAD  WRTRD PARAMETER BLOCK LENGTH
*
*          OUTPUT EDIT OPTION BYTE 1 - NAMES FOR BIT SETTINGS
*
CWR1CODE EQU  1          OTRSUP : CODE TRANSLATION
CWR1LNET EQU  2          OLINEND: LINE END TREATMENT
CWR1MBT1 EQU  4          MODE BIT 1

```

```

CWR1RSET EQU 8          OMANUAL: RESET FROM PAPER TAPE
CWR1HOM EQU 16         OHOM : HOMOGENEOUS OUTPUT
CWR1PTPE EQU 32        OPTAPE : PAPER TAPE CONTROL
CWR1MBT2 EQU 64         MODE BIT 2
CWR1HARD EQU 128       OHCOPY : HARDCOPY
CWR1MMSK EQU 68        MODE=  MODE MASK
CWR1COMP EQU 0         COMP  : COMPATIBLE MODE
CWR1LINE EQU 4         LINE  : LINE MODE
CWR1FORM EQU 64        FORM  : FORMAT MODE
CWR1PHYS EQU 68        PHYS  : PHYSICAL MODE
*
*          OUTPUT EDIT OPTION BYTE 2 - NAMES FOR BIT SETTINGS
*
CWR2HDR EQU 1          OHDR  : HEADER PRESENT
CWR2NOLC EQU 2         ONOLOGC: NO LOGIC CHARS INTERPRET.
CWR2EXT EQU 4          EXTEND : EXTENDED LINE OUTPUT
CWR2INFO EQU 8         OINFO  : INFORMATIVE MESSAGE
CWR2POSN EQU 32        ONOPOSN: NO POSITIONING
CWR2TRAN EQU 32        OTRANS : TRANSPARENT MODE
CWR2BEL EQU 64         OBELL  : ACCOUSTIC ALARM
CWR2ETB EQU 128       OETB   : ETB INSTEAD OF ETX
*
*          INPUT EDIT OPTION BYTE 1 - NAMES FOR BIT SETTINGS
*
CRD1CODE EQU 1         ITRSUP : CODE TRANSLATION
CRD1LNET EQU 2         ILINEND: LINE END TREATMENT
CRD1BACK EQU 4         IGETBS : BACKSPACE
CRD1RSET EQU 8         IMANUAL: RESET TO MANUAL CONTROL
CRD1LCT EQU 16        ILCASE  : LOWER CASE TRANSLATION
CRD1MBT1 EQU 32        MODE BIT 1
CRD1MBT2 EQU 64        MODE BIT 2
CRD1HDR EQU 128       IHDR   : HEADER REQUIRED
CRD1MMSK EQU 96        MODE=  MODE MASK
CRD1COMP EQU 0         COMP  : COMPATIBLE MODE
CRD1LINE EQU 32        LINE  : LINE MODE
CRD1FORM EQU 64        FORM  : FORMAT MODE
CRD1PHYS EQU 96        PHYS  : PHYSICAL MODE
*
*          INPUT EDIT OPTION BYTE 2 - NAMES FOR BIT SETTINGS
*
CRD2GFC EQU 1         IGETFC : GET FUNCTION CODE
CRD2CFD EQU 4         ICFD  : CONFIDENTIAL INPUT DATA
CRD2GIC EQU 8         IGETIC : GET IDENTITY CARD
CRD2EXT EQU 32        EXTEND : EXTENDED LINE INPUT
*
          END

```



**Format 2**

Operation	Operanden
WRTRD	NA1 , , NA2 , , 60 , FEHL , MODE=LINE

NA1	Adresse der Nachricht, die zur Datenstation übertragen wird.
NA2	Adresse des Pufferbereichs für die Eingabenachricht, die von der Datenstation zum Anwendungsprogramm übertragen wird.
60	Länge des Bereiches für NA2 im Anwendungsprogramm.
FEHL	Adresse im Anwendungsprogramm, zu der im Fehlerfall verzweigt wird.

*Beispiel 1*

```

WRTRD  START
        BALR 10,0
        USING *,10
        PRINT NOGEN
FRAGE  WRTRD ABFRAG,,EINB,,5,ENDE
        *,@DCEO 999 921011 53531004
        *,@DCEI 999 921011 53531002
        CLI ANTW,'N'
        BE FRAGE
ENDE   TERM
        *,VERSION 100
ABFRAG DC Y(ABEND-ABFRAG)
        DS CL2
        DC X'01'
        DC 'PROGRAMM BEENDEN?(J/N)'
ABEND  EQU *
EINB   DS OCL5
        DS CL4
ANTW   DS CLI
        END

```

```

/START-PROG $ASSEMBH
% BLS0500 PROGRAM 'ASSEMBH', VERSION '1.1A00' OF '1992-04-30' LOADED
% BLS0552 COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1990. ALL RIGHTS
% RESERVED
% ASS6010 V 1.1A00 OF BS2000 ASSEMBH- READY
//COMPILE SOURCE=*L(L=LIB.EX.MANUAL,E=WRTRD),MAC-LIB=$XXX.LIB.M.SYSLIB.V11.OT10,
MOD-LIB=LIB.EX.MANUAL(E=WRTRD),LIST=PAR(OUT=*L(L=LIB.EX.MANUAL,E=WRTRD)),TEST-SUP=YES
% ASS6011 ASSEMBLY TIME: 1510 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
% ASS6006 LISTING GENERATOR TIME: 745 MSEC
//END
% ASS6012 END OF ASSEMBH-
/START-PROG *M(L=LIB.EX.MANUAL,E=WRTRD)
% BLS0517 MODULE 'WRTRD' LOADED
PROGRAMM BEENDEN? (J/N)
N
PROGRAMM BEENDEN? (J/N)
J
/MOD-JOB-OPT LOG=P(L=N)

```

*Beispiel 2*

## Verwendung des Formats 2

```

WRTRD2  START
        BALR  10,0
        USING *,10
*
LOOP    LA    1,PARAM
        WRTRD (1),PARMOD=31
*
        CLI  CHAR,'N'
        BE  LOOP
END     TERM
*
DS     OF
PARAM  WRTRD QUEST,,INPUT,,5,MF=L,PARMOD=31
*
QUEST  DC    Y(EQUEST-QUEST)
        DS    3X
        DC    'PROGRAMM BEENDEN?(J/N)'
EQUEST EQU   *
*
INPUT  DS    0CL5
LENGTH DS    CL2
UNUSED DS    CL2
CHAR   DS    CL1
END

```

*Beispiel 3*

## WRTRD mit Extended-Line-Mode

```

WRTRD3  START
ANF     BALR  10,0
        USING *,10
        PRINT NOGEN
FRAGE   WRTRD AUSGABE,,EINB,,69,ENDE,MODE=LINE,EXTEND=YES
        *,@DCEO      999   921011  53531004
        *,@DCEI      999   921011  53531002
        CLC  EINNAM(4),=C'ENDE'
        BNE  FRAGE
ENDE    TERM
        *,VERSION 100
VTCSET LOG
EINB    DS    0CL69
        DS    CL4
EINNAM  DS    CL12
EINVOR  DS    CL12
EINSTR  DS    CL25
EINPLZ  DS    CL4
EINORT  DS    CL12
*
*
AUSGABE DS    0H
        DC    Y(ABEND-AUSGABE)
        DS    CL2
        DC    X'01'

```

```

F1      DC      AL1(LOGNP)                      NP
F2      DC      AL1(LOGSPA)                     SPA
F3      DC      AL1(LOGEM3)                     EM3
F4      DC      C'BITTE GEBEN SIE NAMEN UND ANSCHRIFT EIN !'
        DC      AL1(LOGNL)                       2xNL
        DC      AL1(LOGNL)
F5      DC      AL1(LOGNOR)                      NOR
        DC      C'NAME: '
F6      DC      AL1(LOGEPA)                     EPA
NAME    DS      CL12
F7      DC      AL1(LOGSPA)                     SPA
        DC      C'      VORNAME: '
F8      DC      AL1(LOGEPA)                     EPA
VORNAME DS      CL12
F9      DC      AL1(LOGNL)                       NL
F10     DC      AL1(LOGSPA)                     SPA
        DC      C'STRASSE: '
F11     DC      AL1(LOGEPA)                     EPA
STRASSE DS      CL25
F12     DC      AL1(LOGNL)                       NL
F13     DC      AL1(LOGSPA)                     SPA
        DC      C'WOHNORT: '
F14     DC      AL1(LOGEPA)                     EPA
F15     DC      AL1(LOGNUM)                     NUM
PLZ     DS      CL4
F16     DC      AL1(LOGSPA)                     SPA
        DC      C'      '
F17     DC      AL1(LOGEPA)                     EPA
ORT     DS      CL12
F18     DC      AL1(LOGNL)                       NL
F19     DC      AL1(LOGNL)                       NL
F20     DC      AL1(LOGSPA)                     SPA
        DC      C'BEI PROGRAMMENDE BITTE DEN NAMEN "ENDE" EINGEBEN !'
ABEND   EQU     *
        END     ANF

```

```
/START-PROG $ASSEMBH
```

```
% BLS0500 PROGRAM 'ASSEMBH', VERSION '1.1A00' OF '1992-04-30' LOADED
```

```
% BLS0552 COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1990. ALL RIGHTS RESERVED
```

```
% ASS6010 V 1.1A00 OF BS2000 ASSEMBH- READY
```

```
//COMPILE SOURCE=*L(L=LIB.EX.MANUAL,E=WRTRD3),MAC-LIB=$XXXX.LIB.M.SYSLIB.V11.OT10,
MOD-LIB=LIB.EX.MANUAL(E=WRTRD3),
```

```
LIST=PAR(OUT=*L(L=LIB.EX.MANUAL,E=WRTRD3)),TEST-SUP=YES
```

```
% ASS6011 ASSEMBLY TIME: 1576 MSEC
```

```
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
```

```
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
```

```
% ASS6006 LISTING GENERATOR TIME: 833 MSEC
```

```
//END
```

```
% ASS6012 END OF ASSEMBH-
```

```
/START-PROG *M(L=LIB.EX.MANUAL,E=WRTRD3)
```

```
% BLS0517 MODULE 'WRTRD3' LOADED
```

BITTE GEBEN SIE NAMEN UND ANSCHRIFT EIN !

NAME: ..... VORNAME: .....

STRASSE: .....

WOHNORT: ....

BEI PROGRAMMENDE BITTE DEN NAMEN "ENDE" EINGEBEN !

/MOD-JOB-OPT LOG=P(L=NO)

### Beispiel 4a

In diesem Beispiel wird der VTSUCB nicht verwendet.

```

WRTRDT  START 0
*
      BALR 10,0
      USING *,10
*
      WRTRD MELD1,,INPUT1,,40,FEHL,PARMOD=31,MODE=LINE,      +
            OBELL=Y,IICASE=Y,ICFD=Y
*
FEHL    NOP    0
      TERM
*
MELD1   DC     Y(EMELD1-MELD1)
      DS     3X
      DC     C' WRTRD-Ausgabe'
EMELD1  EQU    *
*
*
INPUT1  DS     0CL40
LAENGE  DS     CL2
UNUSED  DS     CL2
DATEN   DS     CL36
*
      END

```

*Beispiel 4b*

In diesem Beispiel wird der VTSUCB verwendet.

```

WRTRDT  START  0
*
        BALR   10,0
        USING  *,10
*
        WRTRD MELD1,,INPUT1,,40,FEHL,PARMOD=31,VTSUCBA=VTSUPAR
*
FEHL    NOP    0
        TERM
*
VTSUPAR VTSUCB MODE=LINE,BELL=YES,LOW=YES,SPECIN=C
*
MELD1   DC     Y(EMELD1-MELD1)
        DS     3X
        DC     C' WRTRD-Ausgabe'
EMELD1  EQU    *
*
INPUT1  DS     0CL40
LAENGE  DS     CL2
UNUSED  DS     CL2
DATEN   DS     CL36
*
        END

```

---

## 4 TIAM-COBOL-Schnittstelle

Die TIAM-COBOL-Schnittstelle bietet dem COBOL-Programmierer folgende Funktionen:

<b>RDATA</b>	Daten lesen von SYSDTA
<b>WROUT</b>	Daten ausgeben nach SYSOUT
<b>WRTRD</b>	Kombinierte Aus- und Eingabe

Diese Funktionen sind in Form von Unterprogrammssprüngen (CALL ...) realisiert. Vor dem CALL-Aufruf zur Ein- bzw. Ausgabe müssen Sie lediglich bestimmte Datenstrukturen versorgen, denen TIAM Informationen über die gewünschte Ein/Ausgabe entnimmt und in denen TIAM einen Fehlercode hinterlegt.

Als Sonderfunktion ist dabei in den Aufrufen CALL "WROUT" und CALL "WRTRD" der Formateinsatz mit FHS integriert (siehe auch FHS-Handbücher [6] und [7]).

### *Hinweise*

- Wenn Sie die TIAM-COBOL-Schnittstelle benutzen, sollten Sie sich vergewissern, daß das TIAM-Schnittstellen-Modul DCCOBRSTS explizit mit dem COBOL-Programm über eine INCLUDE-Anweisung verbunden ist.
- Wenn Sie den integrierten FHS-Modus (Mode='F') benutzen, sollten Sie sich vergewissern, daß das FHS-Modul MFHSCALL explizit mit dem COBOL-Programm über eine INCLUDE-Anweisung verbunden ist. Fehlt das FHS-Modul, wird der Aufruf WROUT oder WRTRD mit einem Returncode abgelehnt.

### **Kurzbeschreibung der COBOL-Aufrufe**

<b>COBOL-Aufrufe</b>	<b>Kurzbeschreibung</b>
RDATA	liest einen Satz aus der Systemdatei SYSDTA.
WROUT	überträgt eine Nachricht in die Systemdatei SYSOUT.
WRTRD	überträgt im Dialogbetrieb eine Nachricht an die Datensichtstation und übernimmt anschließend eine Nachricht.

## 4.1 Datenstrukturen für die TIAM-COBOL-Aufrufe

Zur Verständigung zwischen TIAM und dem Anwendungsprogramm dienen Datenstrukturen, die als COPY-Elemente in einer Bibliothek stehen und in das Anwendungsprogramm kopiert werden.

Die folgenden Datenstrukturen stehen Ihnen zur Verfügung:

### VTSU-CONTROL-BLOCK

Mit Hilfe dieser Datenstruktur können spezielle VTSU-Parameter für die Ein- und Ausgabe eingestellt werden. Die Angaben im VTSUCB ersetzen die EDIT-Optionen in der Datenstruktur TIAM-CONTROL-INFO. Von den Standardwerten abweichende Parameter müssen mit MOVE in diesem VTSUCB gesetzt werden.

Die Datenstruktur VTSU-CONTROL-BLOCK wird mit folgendem Aufruf in das Anwendungsprogramm kopiert.

```
COPY VTSUCB
```

Dieses Copy-Element befindet sich in der Bibliothek \$TSOS.SYSLIB.VTSU-B.111. VTSU-CONTROL-BLOCK ist im VTSU-Handbuch beschrieben (siehe [1] im Literaturverzeichnis).

### LINE-MODE-CONTROL-CHARACTERS

Diese Datenstruktur enthält symbolische Namen für die Steuerzeichen bei Ausgabe im LINE-MODE, EXTENDED-LINE oder INFO im VTSUCB. Diese Steuerzeichen werden mit einer einfachen MOVE-Anweisung in den Ausgabebereich gebracht.

Die Datenstruktur LINE-MODE-CONTROL-CHARACTERS wird mit folgendem Aufruf in das Anwendungsprogramm kopiert.

```
COPY TIAMCTRC
```

Dieses Copy-Element befindet sich in der Bibliothek \$TSOS.SYSLIB.VTSU-B.111. TIAMCTRC ist im VTSU-Handbuch (siehe [1] im Literaturverzeichnis) beschrieben.



## Datenstruktur TIAM-CONTROL-INFO

Diese Datenstruktur steuert die Ein- bzw. Ausgabe:

- Das Anwendungsprogramm übergibt TIAM darin wichtige Informationen (z.B. den Ausgabemodus).
- TIAM wertet diese Datenstruktur aus. Wo keine Einträge vorliegen, nimmt TIAM ggf. Standardwerte an.
- TIAM hinterlegt hier Informationen (z.B. den Fehlercode) über den Ablauf des TIAM-COBOL-Aufrufs.

Die Datenstruktur TIAM-CONTROL-INFO wird mit folgendem Aufruf in das Anwendungsprogramm kopiert.

```
COPY TIAMINFO
```

Dieses Copy-Element befindet sich in der Bibliothek \$TSOS.SYSLIB.TIAM.112.

```

COPY TIAMINFO
*****
* TIAMINFO          002          920620          TIAM          U *
*****
*
* COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1991 *
*           ALL RIGHTS RESERVED                               *
*
*****
01 TIAM-CONTROL-INFO.
   41 EDIT-OUT.
       42 EDIT-MODE          PIC X.
           88 LINE-MODE          VALUE "L".
           88 PHYSICAL-MODE      VALUE "P".
           88 FHS-MODE          VALUE "F".
       42 FILLER          PIC X.
       42 EDIT-OPTIONS     PIC X(6).
           88 NO-OPTIONS        VALUE "NOOPTS".
           88 HARDCOPY          VALUE "HCOPY".
           88 HOMOGENEOUS-OUTPUT VALUE "HOMOUT".
           88 SYSTEM-HEADER     VALUE "HEADER".
           88 ETB-CLOSED        VALUE "ETB".
           88 INFORMATIVE-MESSAGE VALUE "INFO".
           88 TRANS-CODE        VALUE "TRNCD".
           88 EXTENDED-LINE-OUTPUT VALUE "EXTEND".
           88 NO-LOG-CONTROL     VALUE "NLOGC".
           88 BELL              VALUE "BELL".
           88 OVERWRITE         VALUE "OWRITE".
           88 BELL-HOMOGENEOUS   VALUE "BELHOM".
           88 BELL-EXTEND        VALUE "BELEXT".
           88 BELL-INFORMATIVE   VALUE "BELINF".
           88 BELL-NO-LOG-CONTROL VALUE "BELNLC".
           88 HARDCOPY-NO-LOG-CONTROL VALUE "HCNLC".
           88 VTSUCB-USED        VALUE "VTCBU".
       42 FILLER          PIC X(8).
   41 EDIT-IN.
```

```

42 EDIT-MODE PIC X.
    88 LINE-MODE VALUE "L".
    88 PHYSICAL-MODE VALUE "P".
42 FILLER PIC X.
42 EDIT-OPTIONS PIC X(6).
    88 NO-OPTIONS VALUE "NOOPTS".
    88 NO-CORRECTION VALUE "NOCORR".
    88 LOWER-CASE VALUE "LCASE".
    88 DELETE-DEVICE-HEADER VALUE "NOHDR".
    88 GET-FUNCTION-CODE VALUE "GETFC".
    88 CONFIDENTIAL-DATA VALUE "CFDATA".
    88 GET-ID-CARD VALUE "GETIC".
    88 EXTENDED-LINE-INPUT VALUE "EXTEND".
    88 LOWER-CASE-EXTEND VALUE "LOWEXT".
    88 CONF-DATA-LOW-CASE VALUE "CFDLOW".
    88 FCT-CODE-LOW-CASE VALUE "GFCLW".
    88 FCT-CODE-CONF-DATA VALUE "GFCCFD".
    88 FCT-CODE-CONF-DATA-LCASE VALUE "GFCCDL".
    88 FCT-CODE-EXTEND VALUE "GFCEXT".
    88 FCT-CODE-EXTEND-LCASE VALUE "GFCEXL".
    88 DELETE-DEVICE-HEADER-LCASE VALUE "NOHDRL".
    88 VTSUCB-USED VALUE "VTCBU".
    42 FILLER PIC X(8).
41 READLENGTH PIC 9(5) COMP SYNC.
41 ISAM-REQUEST PIC X.
    88 NO-KEY VALUE "N".
    88 GET-KEY VALUE "K".
    88 GET-KEY-LENGTH VALUE "L".
    88 GET-KEY-POSITION VALUE "P".
    88 GET-KEY-POSITION-LENGTH VALUE "B".
41 FILLER PIC X(3).
41 ASSIGNMENT-REQUEST PIC X.
    88 NO-ASSIGNMENT-CODE VALUE "N".
    88 GET-ASSIGNMENT-CODE VALUE "G".
41 FILLER PIC X(3).
41 COPYMEM-ID PIC 9(4) COMP SYNC.
41 FILLER PIC X(2).
41 TIAM-RETURN-INFO.
    42 TIAM-RC PIC 9(4) COMP SYNC.
    42 FILLER PIC X(2).
    42 ASSIGNMENT PIC 9(4) COMP SYNC.
    42 FILLER PIC X(2).
    42 KEY-POSITION PIC 9(4) COMP SYNC.
    42 KEY-LENGTH PIC 9(4) COMP SYNC.
    42 FILLER PIC X(4).

```

\*  
\*

\*

## Beschreibung der Datenfelder

<b>EDIT-OUT</b>	Dieser Teil von TIAM-CONTROL-INFO steuert die Ausgabe der Nachricht. EDIT-OUT wird ausgewertet bei den Aufrufen CALL "WROUT" und CALL "WRTRD".
<b>EDIT-MODE</b>	<p>Dieses Feld (1 byte) gibt den Ausgabe-Modus an. Bei Verwendung des VTSUCB werden die hier angegebenen Edit-Optionen ignoriert. Alle gewünschten Edit-Optionen müssen dann im VTSUCB angegeben werden.</p> <p>Folgende Angaben sind möglich:</p>
"L"	<p>LINE-MODE (Standardwert): Die Datenstation wird als logische Zeilen- bzw. Seitendatenstation behandelt. Die Nachricht kann durch logische Steuerzeichen strukturiert werden.</p> <p>Wollen Sie die zulässigen Line-Mode-Steuerzeichen verwenden, kopieren Sie die Datenstruktur LINE-MODE-CONTROL-CHARACTERS in die WORKING-STORAGE-SECTION und bringen Sie die gewünschten Steuerzeichen mit MOVE-Anweisung in den Ausgabebereich.</p> <p>Weitere Steuerzeichen sind unzulässig und werden vom System in ein vom Anwender definiertes Ersatzzeichen umgewandelt.</p>
"P"	PHYSICAL-MODE: Die Nachricht soll ohne Aufbereitung durch das System an die Datenstation ausgegeben werden.
"F"	FHS-MODE: Die Formatsteuerung FHS wird intern aufgerufen und bereitet die Nachricht auf. Danach gibt TIAM die aufbereitete Nachricht aus. Bei EDIT-MODE="F" wertet TIAM die übrigen Felder von TIAM-CONTROL-INFO nicht aus. Zum integrierten FHS-Anschluß siehe auch FHS-Handbücher [6] und [7].
<b>EDIT-OPTIONS</b>	<p>Dieses Feld (6 byte) bestimmt die Aufbereitungsoptionen für die Ausgabe.</p> <p>Folgende Angaben sind möglich:</p>
"NOOPTS"	NO-OPTIONS (Standardwert): Es wird keine der nachfolgenden Ausgabe-Optionen gewünscht. "NOOPTS" ist zulässig im Line- und Physical-Mode.
"HCOPY"	HARDCOPY: Die Nachricht wird zusätzlich auf einem Hardcopy-Drucker ausgegeben, falls dieser verfügbar ist. Diese Angabe ist nur im Line-Mode zulässig.
"HOMOUT"	HOMOGENEOUS-OUTPUT: Die Nachricht wird unstrukturiert ("homogen") ausgegeben. Die Option ist nur im Line-Mode zulässig.

"HEADER"	<p>SYSTEM-HEADER: Der Ausgabenachricht wird ein vom System erstellter Nachrichtenkopf vorangestellt. Die Option ist zulässig im Physical-Mode.</p>
"ETB"	<p>ETB-CLOSED: Die Ausgabenachricht wird mit ETB abgeschlossen. Die Option ist im Physical-Mode zulässig.</p>
"INFO"	<p>INFORMATIVE MESSAGE: Die Nachricht kann in einer speziellen Informationszeile abgebildet werden, ohne an der Datenstation wichtige Daten zu zerstören.</p> <p>Die Angabe ist vor allem für die Benutzerprogramme gedacht, die "asynchron" Nachrichten an Datenstationen senden, ohne die aktuelle Datenstationsanzeige zu kennen.</p> <p>Die Abbildung erfolgt</p> <ul style="list-style-type: none"> <li>– geschützt in einer Hardware-Anzeigezeile (z.B. DSS 9749, 9750, 9752) oder</li> <li>– geschützt in der letzten Bildschirmzeile</li> <li>– in allen anderen Fällen: wie eine normale Line-Mode-Nachricht.</li> </ul> <p>Ist die Nachricht länger als eine Bildschirmzeile, wird sie aufgeteilt und Zeile für Zeile ausgegeben. Das System berücksichtigt dabei die durch das Kommando /MODIFY-TERMINAL-OPTIONS OVERFLOW-CONTROL=TIME eingestellte Wartezeit.</p> <p>Bei OINFO=YES wird die Angabe OHCOPY=YES ignoriert, d.h. es werden weder die Informationszeile noch der Bildschirminhalt abgedruckt.</p> <p>Erst nach der nächsten Eingabe, auf die eine Ausgabe folgt, wird die Hardware-Anzeigezeile zurückgesetzt.</p>
"TRNCD"	<p>TRANS-CODE: Die Ausgabedaten sollen 'transparent' übertragen werden, d.h., sie bestehen aus beliebigen Binärzeichen (je nach Gerätecode aus 5, 7 oder 8 Bit pro Zeichen), die auf dem Übertragungsweg nicht umgewandelt werden.</p> <p>Ist der Übertragungsweg nicht 'potentiell transparent' generiert, wird die Ausgabe mit dem Fehlercode 4 zurückgewiesen.</p> <p>Option ist zulässig im Physical-Mode.</p>
"EXTEND"	<p>EXTENDED-LINE-OUTPUT: Die Nachricht unterstützt die erweiterte Verwendung von geschützten und ungeschützten Feldern. Die Option ist zulässig in Verbindung mit LINE-MODE.</p>

"NLOGC"	NO-LOG-CONTROL: Alle EBCDIC-Zeichen kleiner X'40' werden durch SUB ersetzt. Es erfolgt keine Auswertung logischer Steuerzeichen. Die Option ist zulässig in Verbindung mit LINE-MODE.
"BELL"	BELL: Bei einer Ausgabe an die Datenstation ertönt ein akustisches Signal (nur bei den Datensichtstationen 9749, 975x, 9763 und 816x mit einem speziellen Gerätezusatz). Die Option ist nur im Line-Mode zulässig.
"OWRITE"	OVERWRITE: Die Nachricht wird am Anfang der aktuellen Zeile ausgegeben. Die Option ist im Line-Mode und nur für Schreibstationen zulässig.
"BELHOM"	BELL-HOMOGENEOUS: Kombination von BELL und HOMOGENEOUS-OUTPUT im Line-Mode.
"BELEXT"	BELL-EXTEND: Kombination von BELL und EXTENDED-LINE-OUTPUT. Die Option ist zulässig in Verbindung mit LINE-MODE.
"BELINF"	BELL-INFORMATIVE: Kombination von BELL und INFORMATVE-MESSAGE im Line-Mode.
"BELNLC"	BELL-NO-LOG-CONTROL: Kombination von BELL und NO-LOG-CONTROL. Die Option ist zulässig in Verbindung mit LINE-MODE.
"HCNLC"	HARDCOPY-NO-LOG-CONTROL: Kombination von HCOPIY und NO-LOG-CONTROL. Die Option ist zulässig in Verbindung mit LINE-MODE.
"VTCBU"	VTSUCB-USED: Es wird ein VTSU-Kontrollblock verwendet. Der Bereich des VTSUCB muß definiert sein und im Funktionsaufruf angegeben werden. Soll der VTSUCB nicht verwendet werden, darf der Bereich des VTSUCB nicht im Funktionsaufruf angegeben werden.
<b>EDIT-IN</b>	Dieser Teil von TIAM-CONTROL-INFO steuert die Eingabe (das Lesen) von Nachrichten. EDIT-IN wird ausgewertet bei den Aufrufen CALL "RDATA" und CALL "WRTRD".
<b>EDIT-MODE</b>	Dieses Feld (1 byte) gibt den Eingabemodus an.  Bei Verwendung des VTSUCB werden die hier angegebenen Edit-Optionen ignoriert. Alle gewünschten Edit-Optionen müssen dann im VTSUCB angegeben werden.

	Folgende Angaben sind möglich:
"L"	LINE-MODE (Standardwert): Die Datenstation wird als logische Zeilendatenstation behandelt. Beim Aufruf CALL "RDATA" ist keine andere Angabe möglich.
"P"	Nur beim Aufruf CALL "WRTRD" zulässig: Die Nachricht soll ohne Aufbereitung durch das System an das Anwendungsprogramm übergeben werden.
<b>EDIT-OPTIONS</b>	Dieses Feld (6 byte) bestimmt die Aufbereitungsoptionen für die Eingabe. Folgende Angaben sind möglich:
"NOOPTS"	NO-OPTIONS (Standardwert): Es wird keine besondere Nachrichtenbehandlung, d.h., keine der nachfolgenden Edit-Optionen gewünscht. "NOOPTS" ist zulässig im Line- und Physical-Mode.
"NOCORR"	NO-CORRECTION: Korrekturzeichen werden wie Textzeichen behandelt (nur bei Schreibstationen).
"LCASE"	LOWER-CASE: Dem Anwendungsprogramm werden auch Kleinbuchstaben übergeben. Die Option ist im Line-Mode und im Physical-Mode zulässig.
"NOHDR"	DELETE-DEVICE-HEADER: Der gerätespezifische Nachrichtencode wird aus der Eingabenachricht entfernt. Diese Option ist nur im Physical-Mode zulässig.
"GETFC"	GET-FUNCTION-CODE: Das erste Byte der Eingabenachricht enthält den Funktionstastencode. Die Option ist nur im Line-Mode zulässig.
"CFDATA"	CONFIDENTIAL-DATA: Die Eingabedaten sind vertraulich und sollen an der Datenstation unsichtbar bleiben. Dies erfolgt je nach Datenstation durch Dunkelsteuerung bzw. Löschen des Bildschirms (wodurch das Bildschirmformat auf 24x80 zurückgesetzt wird) oder durch Überschreiben der Eingabezeile an Schreibstationen. Diese Option ist im Line-Mode zulässig.

"GETIC"	<p>GET-ID-CARD: Die Eingabe soll vom angeschlossenen Ausweisleser erfolgen. Die Eingabedaten können nur aus der Ausweisinformation oder aus dem Kurztelegramm K14 bestehen. Diese Angabe ist nur bei den Datenstationen 8160, 975x und 9763 mit einem definierten Ausweisleser möglich.</p> <p>Bei dieser Option werden dem Anwendungsprogramm auch Kleinbuchstaben (LOWER-CASE) und der Funktionstastencode (GET-FUNCTION-CODE) übergeben.</p> <p>Diese Option ist im Line-Mode zulässig.</p>
"EXTEND"	<p>EXTENDED-LINE-INPUT: Wird bei der Eingabe das Steuerzeichen NEW-LINE erkannt, so wird die Verarbeitung fortgesetzt und der Returncode 44 gesetzt.</p> <p>Die Option ist zulässig in Verbindung mit LINE-MODE und einer Output-Option mit 'Extended Output'.</p>
"LOWEXT"	<p>LOWER-CASE-EXTEND: Kombination von LOWER-CASE und EXTENDED-LINE-INPUT.</p> <p>Die Option ist zulässig in Verbindung mit LINE-MODE und einer Output-Option mit 'Extended Output'.</p>
"CFDLOW"	<p>CONF-DATA-LOW-CASE: Kombination von CONFIDENTIAL-DATA und LOWER-CASE.</p> <p>Diese Option ist zulässig im Line-Mode.</p>
"GFCLOW"	<p>FCT-CODE-LOW-CASE: Kombination von GET-FUNCTION-CODE und LOWER-CASE.</p> <p>Diese Option ist zulässig im Line-Mode.</p>
"GFCCFD"	<p>FCT-CODE-CONF-DATA: Kombination von GET-FUNCTION-CODE und CONFIDENTIAL-DATA.</p> <p>Diese Option ist zulässig im Line-Mode.</p>
"GFCCDL"	<p>FCT-CODE-CONF-DATA-LCASE: Kombination von GET-FUNCTION-CODE, CONFIDENTIAL-DATA und LOWER-CASE.</p> <p>Diese Option ist zulässig im Line-Mode.</p>
"GFCEXT"	<p>FCT-CODE-EXTEND: Kombination von GET-FUNCTION-CODE und EXTENDED-LINE-INPUT.</p> <p>Die Option ist zulässig in Verbindung mit LINE-MODE und einer Output-Option mit 'Extended Output'.</p>
"GFCEXL"	<p>FCT-CODE-EXTEND-LCASE: Kombination von GET-FUNCTION-CODE, EXTENDED-LINE-INPUT und LOWER-CASE.</p> <p>Die Option ist zulässig in Verbindung mit LINE-MODE und einer Output-Option mit 'Extended Output'.</p>

"NOHDRL"	DELETE-DEVICE-HEADER-LCASE: Kombination von DELETE-DEVICE-HEADER und LOWER-CASE. Diese Option ist nur im Physical-Mode zulässig.
"VTGBU"	VTSUCB-USED: Es wird ein VTSU-Kontrollblock verwendet. Der Bereich des VTSUCB muß definiert sein und im Funktionsaufruf angegeben werden. Soll der VTSUCB nicht verwendet werden, darf der Bereich des VTSUCB nicht im Funktionsaufruf angegeben werden.
<b>READLENGTH</b>	In dieses Feld (4 byte) schreibt das Anwenderprogramm die Länge des Eingabebereichs einschließlich 4 byte für das Satzlängenfeld. Die Angabe dieses Feldes ist zwingend, es gibt jedoch keinen Standardwert.
<b>ISAM-REQUEST</b>	In diesem Feld (1 byte) gibt das Anwenderprogramm die Aufbereitungsfunktion für den ISAM-Schlüssel des einzulesenden Satzes an.  Folgende Angaben sind möglich:
"N"	NO-KEY: Der ISAM-Schlüssel wird aus dem Satz entfernt.
"K"	GET-KEY (Standardwert): Keine Behandlung des ISAM-Schlüssels.
"L"	GET-KEY-LENGTH: Anforderung der ISAM-Schlüssellänge (minus 1), die in dem Feld KEY-LENGTH hinterlegt wird.
"P"	GET-KEY-POSITION: Anforderung der ISAM-Schlüsselposition (minus 1), die in dem Feld KEY-POSITION hinterlegt wird.
"B"	GET-KEY-POSITION-LENGTH: Anforderung von Position (minus 1) und Länge (minus 1) des ISAM-Schlüssels. Diese Informationen werden in den Feldern KEY-POSITION und KEY-LENGTH hinterlegt.
<b>ASSIGNMENT-REQUEST</b>	In diesem Feld (1 byte) legt das Anwenderprogramm fest, ob es über Zuweisungsänderungen von SYSDTA benachrichtigt wird. Folgende Angaben sind möglich:
"N"	NO-ASSIGNMENT-CODE (Standardwert): Keine Benachrichtigung über Zuweisungsänderungen
"G"	GET-ASSIGNMENT-CODE: Benachrichtigung über Zuweisungsänderungen. Diese werden im Feld ASSIGNMENT hinterlegt.



**COPYMEM-ID** Dieses Feld (2 byte) dient der Identifikation von Copy-Elementen. Um die Verwendung von TIAMINFO in FILE SECTION zu ermöglichen, muß der Benutzer das Feld mit dem Wert 1 initialisieren.

### TIAM-RETURN-INFO

Dieser Bereich von TIAM-CONTROL-INFO enthält nach dem Aufruf Informationen und Rückkehrcodes von TIAM für das Anwendungsprogramm.

**TIAM-RC** Dieses Feld (2 byte) enthält die TIAM-Fehlercodes. Die Bedeutung der Eintragungen in TIAM-RC finden Sie bei der Beschreibung der Aufrufe.

**ASSIGNMENT** Dieses Feld (2 byte) enthält, falls beim - RDATA-Aufruf verlangt, den Zuweisungsschlüssel.

Folgende Eintragungen sind in ASSIGNMENT möglich:

0	keine Änderung der Zuweisung
1	SYSDTA ist ein Diskettengerät
4	SYSDTA ist eine SAM-Datei
8	SYSDTA ist eine ISAM-Datei
20	SYSDTA ist eine Datenstation
24	SYSDTA ist eine SDF-P-Variable
32	SYSDTA ist ein Element einer PLAM-Bibliothek.

**KEY-POSITION** Dieses Feld (2 byte) enthält, falls beim - RDATA-Aufruf verlangt, nach Rückkehr zum COBOL-Programm den Wert:

ISAM-Schlüsselposition - 1

**KEY-LENGTH** Dieses Feld (2 byte) enthält, falls beim RDATA-Aufruf verlangt, nach Rückkehr zum Programm den Wert:

ISAM-Schlüssellänge - 1

Welche Felder bei welchem TIAM-COBOL-Aufruf verwendet werden, ist bei der Beschreibung der Aufrufe beschrieben.

## 4.2 TIAM-COBOL-Aufrufe

Dem COBOL-Programmierer stehen die folgenden COBOL-Aufrufe zur Verfügung:

CALL "RDATA" (siehe Seite 147)

CALL "WROUT" (siehe Seite 154)

CALL "WRTRD" (siehe Seite 160)

Für jeden Aufruf ist dabei ein Diagramm vorhanden, das zeigt,

- welche Datenstrukturen der Aufruf verwendet,
- welche Bereiche man vor dem Aufruf versorgen muß oder kann,
- welche Informationen TIAM zurückgibt.

Die unterschiedlichen Rasterungen in den Diagrammen haben folgende Bedeutung:



Diesen Bereich muß das Anwendungsprogramm vor dem Aufruf versorgen.



Diesen Bereich muß das Anwendungsprogramm nur unter bestimmten Bedingungen versorgen.

Bereiche ohne Rasterung sind wahlfrei. Angaben sind hier nur zu machen, wenn die entsprechende Funktion/Eigenschaft gewünscht wird.



Dieser Bereich enthält Informationen, die TIAM zurückgibt.

## 4.2.1 Daten lesen von SYSDTA: CALL "RDATA"

Mit dem Aufruf CALL "RDATA" liest der COBOL-Programmierer einen Datensatz von SYSDTA. SYSDTA kann ein Diskettengerät, ein Element einer PLAM-Bibliothek, eine SDF-P-Variable, eine katalogisierte SAM- oder ISAM-Datei oder die prozeßführende Datenstation sein.

### Format des Aufrufs:

```
CALL "RDATA" USING TIAM-CONTROL-INFO
                  benutzerbereich
                  [VTSUCB].
```

Die Angaben bei USING haben folgende Bedeutung:

### TIAM-CONTROL-INFO

steuert den TIAM-Aufruf. TIAM hinterlegt hier Rückkehrinformationen über den Ablauf des RDATA-Aufrufs.

### benutzerbereich

10 ist der Name des Eingabebereichs. TIAM schreibt in die ersten vier Byte dieses Bereichs die (tatsächliche) Länge der Eingabenachricht einschließlich des Satzlängenfeldes. Stellen Sie also bei der Definition dieses Bereichs dem eigentlichen Nachrichtenfeld ein vier byte langes Längenfeld voran. Dies erreichen Sie zum Beispiel so:

```
01  BENUTZERBEREICH.
    41  LAENGENFELD   PIC 9(5) COMP SYNC.
    41  TEXTFELD     PIC X(m).
```

Vor dem RDATA-Aufruf müssen Sie die Länge dieses Eingabebereichs (also den Wert m+4) in das Feld READLENGTH in TIAM-CONTROL-INFO eintragen. Ist die Nachricht länger als der Eingabebereich, wird sie abgeschnitten und der entsprechende Fehlercode geliefert.

### VTSUCB

Der VTSUCB soll zur Nachrichtenaufbereitung angegeben werden (siehe Abschnitt VTSU-Control-Block im VTSU-Handbuch [1]). Im Feld EDIT-Options müssen Sie dann "VTTCBU" angeben.

Vor dem TIAM-Aufruf **CALL RDATA** zum Lesen von SYSDTA versorgen Sie bitte die folgenden Felder:

- 1 In EDIT-IN (TIAM-CONTROL-INFO) **muß** das Feld **EDIT-MODE** den Wert "L" (für Line-Mode) haben (Standardwert).
- 3 In TIAM-CONTROL-INFO **muß** in dem Feld **READLENGTH** die Länge des Eingabebereichs stehen (einschließlich der 4 byte für das Längenfeld).

Außerdem können Sie in TIAM-CONTROL-INFO wahlweise folgende Angaben machen:

- 2 Wenn Sie bestimmte Eingabe-Optionen (siehe Seite 137) wünschen, versorgen Sie bitte das Feld **EDIT-OPTIONS** in EDIT-IN mit dem entsprechenden Wert ("NO-CORR", "LCASE", "GETFC", "CFDATA", "GETIC", "CFDLOW", "GFCLOW", "GFC-CFD" oder "GFCCDL"). Wenn Sie mit dem VTSUCB arbeiten, muß hier "VTCBU" angegeben werden.
- 4 Wenn Sie von TIAM Informationen über den ISAM-Schlüssel der Eingabenachricht haben wollen, versorgen Sie bitte das Feld **ISAM-REQUEST** mit einem der Werte "N", "K", "L", "P" oder "B".
- 5 Wenn Sie im Anwendungsprogramm über Zuweisungsänderungen von SYSDTA informiert werden wollen, versorgen Sie bitte das Feld **ASSIGNMENT-REQUEST** mit dem Wert "G".

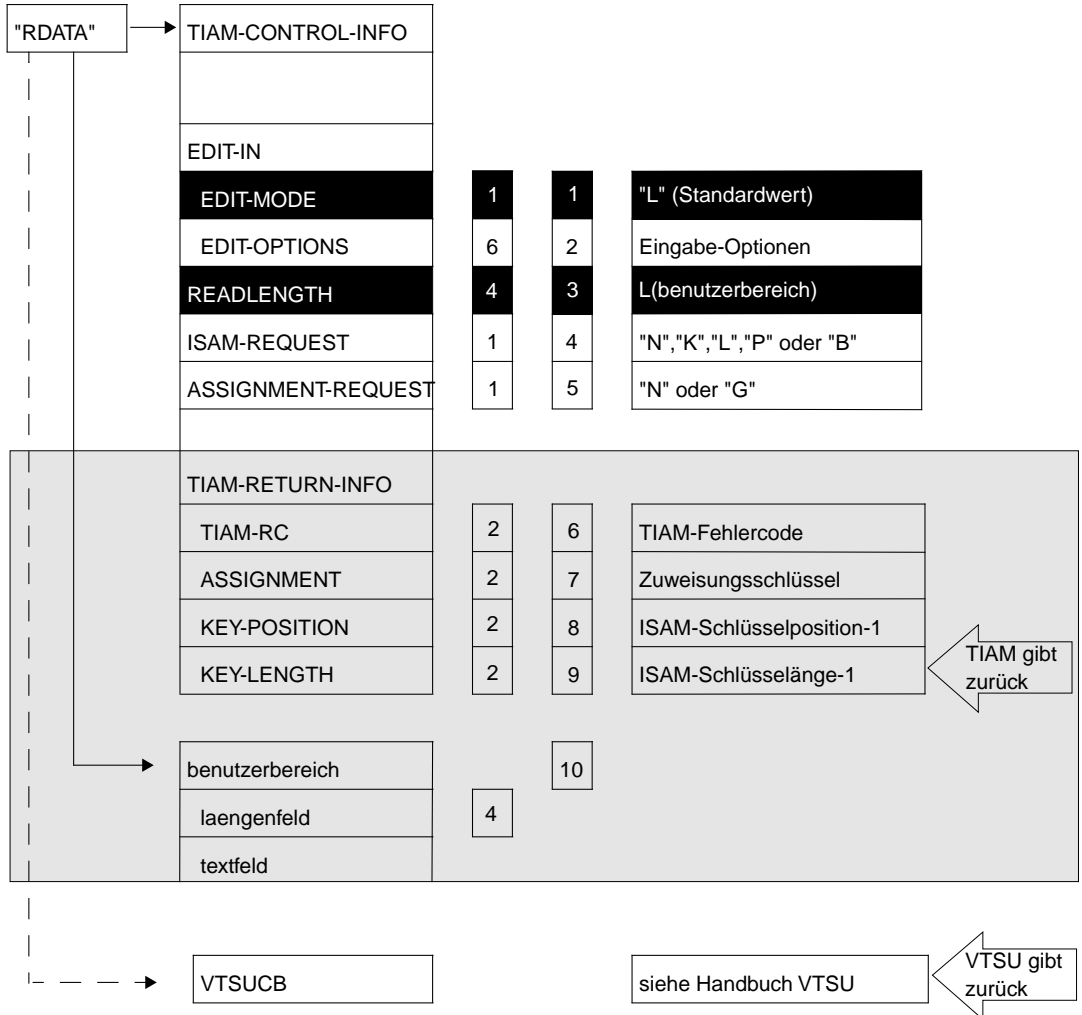
#### TIAM gibt zurück:

- 6 den TIAM-Fehlercode in dem Feld **TIAM-RC** (siehe unten). Hat er den Wert 0, ist der Aufruf fehlerfrei verlaufen.
- 7 den Zuweisungsschlüssel in **ASSIGNMENT**, wenn Sie dies im Feld ASSIGNMENT-REQUEST verlangt haben.
- 8 den Wert 'ISAM-Schlüsselposition - 1' in **KEY-POSITION**, wenn Sie dies im Feld ISAM-REQUEST verlangt haben.
- 9 den Wert 'ISAM-Schlüssellänge - 1' in **KEY-LENGTH**, wenn Sie dies im Feld ISAM-REQUEST angefordert haben.

Folgende Fehlercodes können beim Aufruf CALL "RDATA" auftreten:

0	normale Beendigung
4	nicht behebbarer Ein/Ausgabe-Fehler
8	Parameterfehler
12	Abschneiden beim Lesen
16	Dateiende
32	ein unzulässiger Parameter wurde vom System korrigiert
48	VTSU hat Fehler erkannt, siehe vollständigen RC im VTSUCB (nur bei Verwendung des VTSUCB)

CALL	USING	Länge	Beschreibung	Versorgung der Felder
------	-------	-------	--------------	-----------------------



*Programmbeispiel RDATA*

```

*****
* BEISPIEL FUER DEN   R D A T A           *
* MIT DER FUNKTION RDATA WERDEN SAETZE   *
* EINGELESEN. DIESE SAETZE WERDEN DANN  *
* MIT DER FUNKTION WROUT AUSGEGEBEN.    *
* ZUSAETZLICH WERDEN DIE LAENGE DES     *
* ISAM-SCHLUESSELS (MINUS 1) UND DIE    *
* POSITION DES ISAM-SCHLUESSELS (MINUS 1) *
* AUSGEGEBEN.                          *
* BEI DATEIENDE (EOF) ODER NACH EINGABE *
* VON "END" WIRD DAS PROGRAMM BEENDET.  *
*****
*
IDENTIFICATION DIVISION.
PROGRAM-ID. RTEST.
ENVIRONMENT DIVISION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY TIAMINFO.
COPY TIAMCTRC.
*
01 INAREA.
41 MSGLENGTH PIC 9(5) COMP SYNC.
41 TXT       PIC X(40).
*
01 OUTAREA-TXT.
41 MSGLEN    PIC 9(5) COMP SYNC VALUE 131.
41 FILLER    PIC X.
41 TXT-INPUT PIC X(40).
41 STZ1      PIC X.
41 TXT-KEYPOS PIC X(39) VALUE
        "POSITION DES ISAM-SCHLUESSELS MINUS 1: ".
41 KPOSITION PIC X(4).
41 STZ2      PIC X.
41 TXT-KEYLEN PIC X(37) VALUE
        "LAENGE DES ISAM-SCHLUESSELS MINUS 1: ".
41 KLENGTH   PIC X(4).
*
01 OUTAREA-ERROR.
41 MSGLENGTH PIC 9(5) COMP SYNC VALUE 25.
41 FILLER    PIC X.
41 TXT       PIC X(20).
*
*
PROCEDURE DIVISION.
ANF.
MOVE "L"      TO EDIT-MODE      IN EDIT-IN.
MOVE "LCASE"  TO EDIT-OPTIONS  IN EDIT-IN.
MOVE "B"      TO ISAM-REQUEST.
MOVE 44       TO READLENGTH.
MOVE NEW-LINE TO STZ1 IN OUTAREA-TXT.
MOVE NEW-LINE TO STZ2 IN OUTAREA-TXT.
MOVE 1        TO COPYMEM-ID.
*
READLOOP.
MOVE " "      TO TXT IN INAREA.

```

```
CALL "RDATA" USING TIAM-CONTROL-INFO
                    INAREA.
*
IF TIAM-RC IN TIAM-RETURN-INFO IS NOT EQUAL 16
AND TXT IN INAREA IS NOT EQUAL "END"
IF TIAM-RC IN TIAM-RETURN-INFO IS ZERO
  MOVE TXT          IN INAREA
    TO TXT-INPUT   IN OUTAREA-TXT
  MOVE KEY-POSITION IN TIAM-RETURN-INFO
    TO KPOSITION   IN OUTAREA-TXT
  MOVE KEY-LENGTH   IN TIAM-RETURN-INFO
    TO KLENGTH     IN OUTAREA-TXT
  CALL "WROUT" USING TIAM-CONTROL-INFO
                    OUTAREA-TXT
  GO TO READLOOP
ELSE
  MOVE "FEHLERHAFTE EINGABE!"
    TO TXT          IN OUTAREA-ERROR
  CALL "WROUT" USING TIAM-CONTROL-INFO
                    OUTAREA-ERROR
  GO TO READLOOP
ELSE
  MOVE "ENDE DER EINGABE!"
    TO TXT          IN OUTAREA-ERROR.
  CALL "WROUT" USING TIAM-CONTROL-INFO
                    OUTAREA-ERROR.
STOP RUN.
```

*Programmbeispiel RDATA mit Verwendung des VTSUCB*

```

*****
* BEISPIEL FUER DEN   R D A T A           *
* MIT DER FUNKTION RDATA WERDEN SAETZE   *
* EINGELESEN. DIESE SAETZE WERDEN DANN  *
* MIT DER FUNKTION WROUT AUSGEGEBEN.    *
* ZUSAETZLICH WERDEN DIE LAENGE DES     *
* ISAM-SCHLUESSELS (MINUS 1) UND DIE    *
* POSITION DES ISAM-SCHLUESSELS (MINUS 1) *
* AUSGEGEBEN.                           *
* BEI DATEIENDE (EOF) ODER NACH EINGABE *
* VON "END" WIRD DAS PROGRAMM BEENDET.  *
*****
*
IDENTIFICATION DIVISION.
PROGRAM-ID. RTEST.
ENVIRONMENT DIVISION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY TIAMINFO.
    COPY TIAMCTRC.
    COPY VTSUCBC.
*
01 INAREA.
   41 MSGLENGTH PIC 9(5) COMP SYNC.
   41 TXT       PIC X(40).
*
01 OUTAREA-TXT.
   41 MSGLEN    PIC 9(5) COMP SYNC VALUE 131.
   41 FILLER    PIC X.
   41 TXT-INPUT PIC X(40).
   41 STZ1      PIC X.
   41 TXT-KEYPOS PIC X(39) VALUE
      "POSITION DES ISAM-SCHLUESSELS MINUS 1: ".
   41 KPOSITION PIC X(4).
   41 STZ2      PIC X.
   41 TXT-KEYLEN PIC X(37) VALUE
      "LAENGE DES ISAM-SCHLUESSELS MINUS 1: ".
   41 KLENGTH   PIC X(4).
*
01 OUTAREA-ERROR.
   41 MSGLENGTH PIC 9(5) COMP SYNC VALUE 25.
   41 FILLER    PIC X.
   41 TXT       PIC X(20).
*
*
PROCEDURE DIVISION.
ANF.
    MOVE "L"      TO MODUS          IN VTSUCB.
    MOVE "Y"      TO LOWER-CHARS   IN VTSUCB.
    MOVE "B"      TO ISAM-REQUEST.
    MOVE 44       TO READLENGTH.
    MOVE NEW-LINE TO STZ1   IN OUTAREA-TXT.
    MOVE NEW-LINE TO STZ2   IN OUTAREA-TXT.
    MOVE 1        TO COPYMEM-ID.
*
READLOOP.

```



```
MOVE " " TO TXT IN INAREA.
MOVE "VTCBU" TO EDIT-OPTIONS IN EDIT-IN.
CALL "RDATA" USING TIAM-CONTROL-INFO
                INAREA
                VTSUCB.

*
IF TIAM-RC IN TIAM-RETURN-INFO IS NOT EQUAL 16
AND TXT IN INAREA IS NOT EQUAL "END"
IF TIAM-RC IN TIAM-RETURN-INFO IS ZERO
MOVE TXT IN INAREA
  TO TXT-INPUT IN OUTAREA-TXT
MOVE KEY-POSITION IN TIAM-RETURN-INFO
  TO KPOSITION IN OUTAREA-TXT
MOVE KEY-LENGTH IN TIAM-RETURN-INFO
  TO KLENGTH IN OUTAREA-TXT
CALL "WROUT" USING TIAM-CONTROL-INFO
                OUTAREA-TXT
GO TO READLOOP
ELSE
MOVE "FEHLERHAFTE EINGABE!"
  TO TXT IN OUTAREA-ERROR
CALL "WROUT" USING TIAM-CONTROL-INFO
                OUTAREA-ERROR
GO TO READLOOP
ELSE
MOVE "ENDE DER EINGABE!"
  TO TXT IN OUTAREA-ERROR.
CALL "WROUT" USING TIAM-CONTROL-INFO
                OUTAREA-ERROR.
STOP RUN.
```

## 4.2.2 Daten ausgeben nach SYSOUT: CALL "WROUT"

Mit dem Aufruf CALL "WROUT" gibt der COBOL-Programmierer eine Nachricht aus nach SYSOUT. SYSOUT kann ein Element einer PLAM-Bibliothek, eine SDF-P-Variable, eine katalogisierte SAM- oder ISAM-Datei oder die prozeßführende Datenstation sein. Als Sonderfunktion ist im Aufruf CALL "WROUT" der Formateinsatz mit FHS integriert. FHS wird dabei intern von TIAM aufgerufen.

### Format des Aufrufs:

```
CALL "WROUT" USING TIAM-CONTROL-INFO
                  [VTSUCB]
                  [FHS-MAIN-PAR].
```

Die Angaben bei USING haben folgende Bedeutung:

### TIAM-CONTROL-INFO

steuert den TIAM-Aufruf. TIAM hinterlegt hier Rückkehrinformationen über den Ablauf des WROUT-Aufrufs.

### benutzerbereich

- 4 ist der Name des Ausgabebereichs. Dieser muß wie folgt aufgebaut sein:
- Vor dem Ausgabebetext muß ein vier byte langes Längenfeld definiert sein.
  - Danach folgt im Line- und im Physical-Mode ein 1 byte langes Füllzeichen (entfällt bei integriertem FHS-Einsatz).
  - Dann erst kommt das eigentliche Textfeld, das den Ausgabebetext enthält.
- Dies erreichen Sie z.B. so:
- Im Line- oder Physical-Mode:

```
01  BENUTZERBEREICH.
    41  LAENGFELD      PIC 9(5) COMP SYNC.
    41  FILLER        PIC X.
    41  AUSGABETEXT   PIC X(m).
```

Vor dem WROUT-Aufruf müssen Sie die Länge dieses Ausgabebereichs (also den Wert m+5) in das Längenfeld eintragen (im Line- und im Physical-Mode). Das Füllzeichen kann einen beliebigen Wert haben, es hat keinen Einfluß auf die Ausgabe.

- Bei FHS-Einsatz (FHS-Mode):

```
01      BENUTZERBEREICH.
        41 LAENGENFELD          PIC 9(5) COMP.
        COPY adressierungshilfe.
```

wobei adressierungshilfe die COBOL-Adressierungshilfe ist, die FHS bzw. IFG bei der Formaterstellung liefern. In diesem Fall wird das Längenfeld von FHS versorgt.

## VTSUCB

Der VTSUCB soll zur Nachrichtenaufbereitung angegeben werden (siehe VTSU-Handbuch [1]). Im Feld EDIT-Options müssen Sie dann "VTCBU" angeben.

## FHS-MAIN-PAR (nur bei FHS-Einsatz)

steuert die Formatierung. Welche Felder hier zu versorgen sind, entnehmen Sie bitte dem FHS-Benutzerhandbuch.

Vor dem TIAM-Aufruf **CALL WROUT** versorgen Sie bitte die folgenden Felder:

- 1 In EDIT-OUT (TIAM-CONTROL-INFO) **muß** das Feld **EDIT-MODE** einen der Werte "L" (für Line-Mode), "P" (für Physical-Mode) oder "F" (für FHS-Mode) haben. Ist EDIT-MODE="F" (FHS-Einsatz) wertet TIAM die restlichen Felder von TIAM-CONTROL-INFO nicht aus. Ist EDIT-MODE="L" (Line-Mode), können Sie in die Nachricht Steuerzeichen einfügen. TIAM stellt diese Steuerzeichen mit dem COPY-Element TIAMCTRC zur Verfügung (siehe Seite 136).
- 5 Bei EDIT-MODE="L" oder "P" (Line- oder Physical-Mode) **muß** in den ersten vier Byte des Ausgabebereichs die Länge der Nachricht stehen (einschließlich Längenfeld und Füllzeichen).

Außerdem können Sie in TIAM-CONTROL-INFO wahlweise folgende Angaben machen:

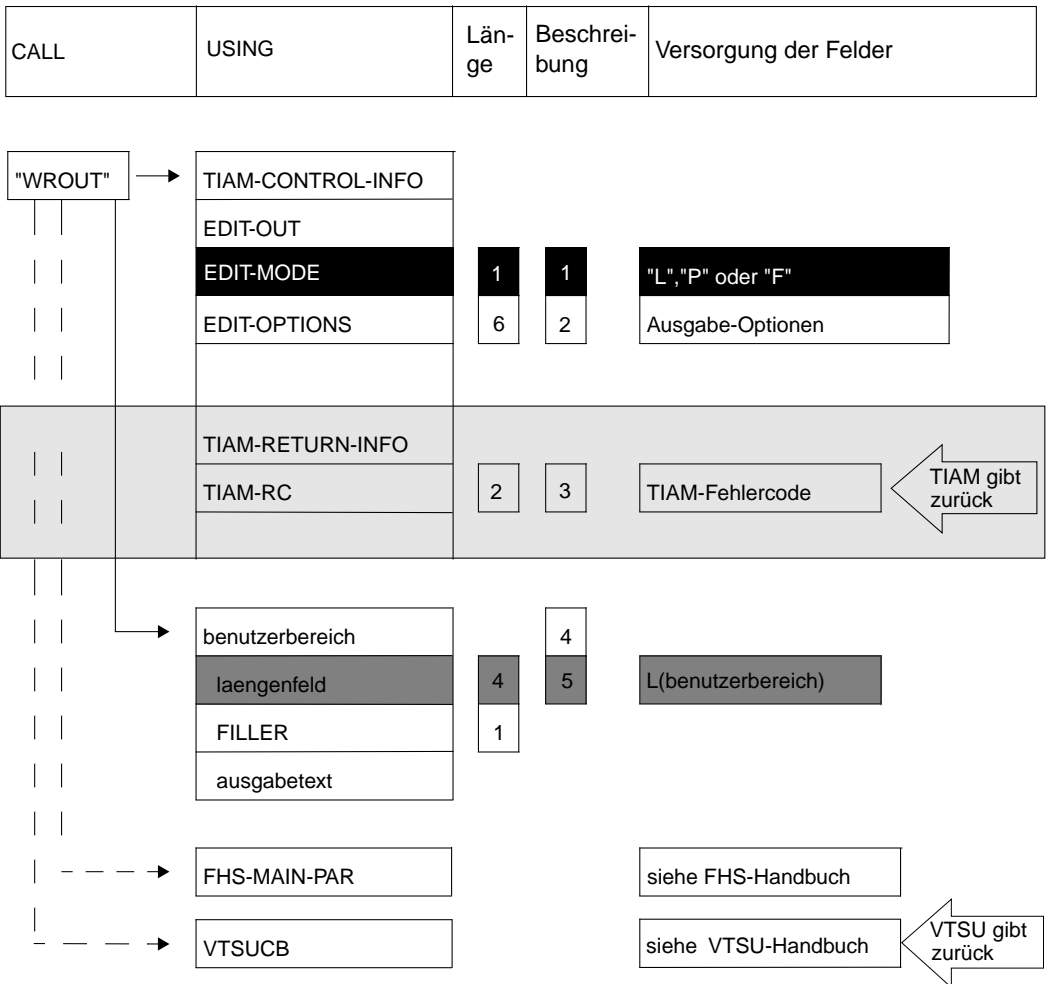
- 2 Wenn Sie bestimmte Ausgabe-Optionen (siehe Seite 137) wünschen, versorgen Sie bitte das Feld **EDIT-OPTIONS** in EDIT-OUT mit dem entsprechenden Wert ("HCOPY", "HOMOUT", "HEADER", "ETB", "INFO", "TRNCD", "EXTEND", "NLOGC", "BELL", "OWRITE", "BELHOM", "BELEXT", "BELINF", "BELNLC", "HCNLC"). Dieses Feld wird im FHS-Mode (EDIT-MODE="F") nicht ausgewertet. Wenn Sie mit dem VTSUCB arbeiten, muß hier "VTCBU" angegeben werden.

**TIAM gibt zurück:**

- 3 den TIAM-Fehlercode in dem Feld **TIAM-RC** (siehe unten). Hat er den Wert 0, ist der Aufruf fehlerfrei verlaufen.

Folgende Fehlercodes können beim Aufruf CALL "WROUT" auftreten:

- 0 normale Beendigung
- 4 nicht behebbarer Ein/Ausgabe-Fehler
- 8 Parameterfehler
- 12 Abschneiden beim Schreiben
- 32 ein unzulässiger Parameter wurde vom System korrigiert
- 36 Formatierung durch FHS nicht möglich, da die FHS-Module fehlen
- 40 FHS-Rückkehrcode ungleich Null (siehe FHS-Rückkehrcode)
- 48 VTSU hat Fehler erkannt, siehe vollständigen RC im VTSUCB) (nur bei Verwendung des VTSUCB)



*Hinweis*

Bei integriertem FHS-Einsatz (MODE="F") beachten Sie bitte beim Binden Ihres Anwendungsprogramms, daß der FHS-Modul **MFHSCALL** explizit mit einer INCLUDE-Anweisung zum COBOL-Programm dazugebunden werden muß. Fehlt beim WROUT-Aufruf mit MODE="F" der FHS-Modul, weist TIAM den Aufruf mit einem Returncode zurück.

*Programmbeispiel WROUT im Line-Mode*

```

*****
* BEISPIEL FUER DEN   W R O U T.           *
* ES WIRD AUF DER DATENSTATION EIN SATZ   *
* MIT DER FUNKTION W R O U T IM LINE-MODE *
* AUSGEGEBEN.                                           *
*****

IDENTIFICATION DIVISION.
PROGRAM-ID.   WTEST.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*
*   COPY TIAMINFO.
*
*   COPY TIAMCTRC.
*
01  OUTPUT-AREA.
    41  MSGLENGTH PIC 9(5)  COMP SYNC  VALUE 85.
    41  FILLER    PIC X.
    41  TXT1      PIC X(36) VALUE
           "DAS IST EIN BEISPIEL FUER DEN WROUT.".
    41  STZ1     PIC X.
    41  TXT2     PIC X(20) VALUE "DIESER TEXT WIRD IM ".
    41  STZ2     PIC X.
    41  TXT3     PIC X(9)  VALUE "LINE-MODE".
    41  STZ3     PIC X.
    41  TXT4     PIC X(12) VALUE " AUSGEGEBEN.".
*
*
PROCEDURE DIVISION.
HP.
    MOVE NEW-LINE      TO STZ1  IN OUTPUT-AREA.
    MOVE EMPH-LAYOUT1 TO STZ2  IN OUTPUT-AREA.
    MOVE NORMAL-LAYOUT TO STZ3  IN OUTPUT-AREA.
*
    MOVE "L"          TO EDIT-MODE    IN EDIT-OUT.
    MOVE "NOOPTS"     TO EDIT-OPTIONS IN EDIT-OUT.
    MOVE 1            TO COPYMEM-ID.
    CALL "WROUT"      USING  TIAM-CONTROL-INFO OUTPUT-AREA.
*
    STOP RUN.

```

*Programmbeispiel WROUT im Line-Mode mit Verwendung des VTSUCB*

```

*****
* BEISPIEL FUER DEN   W R O U T.           *
* ES WIRD AUF DER DATENSTATION EIN SATZ   *
* MIT DER FUNKTION W R O U T IM LINE-MODE *
* AUSGEGEBEN.                                           *
*****

IDENTIFICATION DIVISION.
PROGRAM-ID.   WTEST.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*
*   COPY TIAMINFO.
*
*   COPY TIAMCTRC.
*
*   COPY VTSUCBC.
*
01  OUTPUT-AREA.
    41  MSGLNGTH PIC 9(5)  COMP SYNC  VALUE 85.
    41  FILLER   PIC X.
    41  TXT1     PIC X(36) VALUE
           "DAS IST EIN BEISPIEL FUER DEN WROUT.".
    41  STZ1     PIC X.
    41  TXT2     PIC X(20) VALUE "DIESER TEXT WIRD IM ".
    41  STZ2     PIC X.
    41  TXT3     PIC X(9)  VALUE "LINE-MODE".
    41  STZ3     PIC X.
    41  TXT4     PIC X(12) VALUE " AUSGEGEBEN.".
*
*
PROCEDURE DIVISION.
HP.
    MOVE NEW-LINE      TO STZ1  IN OUTPUT-AREA.
    MOVE EMPH-LAYOUT1 TO STZ2  IN OUTPUT-AREA.
    MOVE NORMAL-LAYOUT TO STZ3  IN OUTPUT-AREA.
*
    MOVE 1             TO COPYMEM-ID.
    MOVE "L"           TO MODUS   IN VTSUCB.
    MOVE "VTCBU"       TO EDIT-OPTIONS IN EDIT-OUT.
    CALL "WROUT" USING TIAM-CONTROL-INFO OUTPUT-AREA
                       VTSUCB.
*
    STOP RUN.

```

### 4.2.3 Kombinierte Aus- und Eingabe: CALL "WRTRD"

Mit dem Aufruf CALL "WRTRD" gibt der COBOL-Programmierer eine Nachricht an eine Datenstation aus und liest anschließend eine Nachricht von der Datenstation. Außer der zur Datenstation übertragenen Nachricht erscheint dabei kein weiteres Eingabeaufforderungszeichen.

CALL "WRTRD" kann nur bei der Zugriffsmethode TIAM verwendet werden.

Als Sonderfunktion ist im Aufruf CALL "WRTRD" der Formateinsatz mit FHS integriert. FHS wird dabei intern von TIAM aufgerufen.

Nach einer Eingabe CALL "WRTRD" wird für die Datensichtstationen 8160, 9749, 975x und 9763 die Tastatur gesperrt, so daß bis zur nächsten Ausgabe keine weitere Eingabe möglich ist. Lediglich Kurztelegramme sind zugelassen.

#### Format des Aufrufs:

```
CALL "WRTRD" USING TIAM-CONTROL-INFO
                   benutzerbereich1
                   benutzerbereich2
                   [VTSUCB]
                   [FHS-MAIN-PAR].
```

Die Angaben bei USING haben folgende Bedeutung:

#### TIAM-CONTROL-INFO

steuert den TIAM-Aufruf. TIAM hinterlegt hier Rückkehrinformationen über den Ablauf des WRTRD-Aufrufs.

#### benutzerbereich1

- 7 ist der Name des Ausgabebereichs. Dieser muß wie folgt aufgebaut sein:
- Vor dem Ausgabebetext muß ein vier byte langes Längenfeld definiert sein.
  - Danach folgt im Line- und im Physical-Mode ein 1 byte langes Füllzeichen (entfällt bei integriertem FHS-Einsatz).
  - Dann erst kommt das eigentliche Textfeld, das den Ausgabebetext enthält.

Dies erreichen Sie z.B. so:

- Im Line- oder Physical-Mode:

```
01  BENUTZERBEREICH1.
    41  LAENGENFELD      PIC 9(5) COMP SYNC.
    41  FILLER           PIC X.
    41  AUSGABETEXT     PIC X(m).
```



Vor dem WRTRD-Aufruf müssen Sie die Länge dieses Ausgabebereichs (also den Wert m+5) in das Längenfeld eintragen (im Line- und im Physical-Mode). Das Füllzeichen kann einen beliebigen Wert haben, es hat keinen Einfluß auf die Ausgabe.

- Bei FHS-Einsatz (FHS-Mode):

```
01  BENUTZERBEREICH1.
    41 LAENGENFELD      PIC 9(5) COMP.
    COPY adressierungshilfe.
```

wobei adressierungshilfe die COBOL-Adressierungshilfe ist, die FHS bzw. IFG bei der Formaterstellung liefern.

In diesem Fall wird das Längenfeld von FHS versorgt.

## benutzerbereich2

- 9 ist der Name des Eingabebereichs. Dieser muß wie folgt aufgebaut sein:

- Vor dem Eingabefeld muß ein vier byte langes Längenfeld definiert sein.
- Dann erst kommt das eigentliche Textfeld, in das TIAM die Eingabenachricht schreibt.

Dies erreichen Sie z.B. so:

- Im Line- oder Physical-Mode:

```
01  BENUTZERBEREICH2.
    41 LAENGENFELD      PIC 9(5) COMP SYNC.
    41 EINGABEFELD     PIC X(m).
```

Vor dem WRTRD-Aufruf müssen Sie die Länge dieses Eingabebereichs (also den Wert m+4) in das Feld READLENGTH eintragen (im Line- und im Physical-Mode).

- Bei FHS-Einsatz (FHS-Mode):

```
01  BENUTZERBEREICH2.
    41 LAENGENFELD      PIC 9(5) COMP.
    COPY adressierungshilfe.
```

wobei adressierungshilfe die COBOL-Adressierungshilfe ist, die FHS bzw. IFG bei der Formaterstellung liefern.

In diesem Fall wird das Längenfeld von FHS versorgt.

## VTSUCB

Der VTSUCB soll zur Nachrichtenaufbereitung angegeben werden (siehe Abschnitt VTSU-Control-Block im VTSU-Handbuch [1]). Im Feld EDIT-Options müssen Sie dann "VTVCBU" angeben.

**FHS-MAIN-PAR (nur bei FHS-Einsatz)**

steuert die Formatierung. Welche Felder hier zu versorgen sind, entnehmen Sie bitte dem FHS-Benutzerhandbuch.

Vor dem TIAM-Aufruf **CALL WRTRD** versorgen Sie bitte die folgenden Felder:

- 1 In EDIT-OUT (TIAM-CONTROL-INFO) **muß** das Feld **EDIT-MODE** einen der Werte "L" (für Line-Mode), "P" (für Physical-Mode) oder "F" (für FHS-Mode) haben. Ist EDIT-MODE="F" (FHS-Einsatz) wertet TIAM die restlichen Felder von TIAM-CONTROL-INFO nicht aus. Ist EDIT-MODE="L" (Line-Mode), können Sie in die Nachricht Steuerzeichen einfügen. TIAM stellt diese Steuerzeichen mit dem COPY-Element TIAMCTRC zur Verfügung (siehe Seite 136).

Bei EDIT-MODE= "L" oder "P" (Line- oder Physical-Mode) **müssen** zusätzlich versorgt werden:

- 3 - in EDIT-IN das Feld **EDIT-MODE** mit einem der Werte "L" (Line-Mode) oder "P" (Physical-Mode),
- 5 - das Feld **READLENGTH** mit der Länge des Eingabebereichs (einschließlich Längensfeld),
- 8 - die ersten vier Byte des Ausgabebereichs (Längensfeld) mit der Länge der Nachricht (einschließlich Längensfeld und Füllzeichen).

Außerdem können Sie in TIAM-CONTROL-INFO wahlweise folgende Angaben machen:

- 2 Wenn Sie bestimmte Ausgabe-Optionen (siehe Seite 137) wünschen, versorgen Sie bitte das Feld **EDIT-OPTIONS** in EDIT-OUT mit dem entsprechenden Wert ("HCOPY", "HOMOUT", "HEADER", "ETB", "TRNCD", "EXTEND", "NLOGC", "BELL", "OWRITE", "BELHOM", "BELEXT", "BELNLC" oder "HCNLC"). Dieses Feld wird im FHS-Mode (EDIT-MODE="F") nicht ausgewertet. Wenn Sie mit dem VTSUCB arbeiten muß hier "VTTCBU" angegeben werden.
- 4 Wenn Sie bestimmte Eingabe-Optionen (siehe Seite 137) wünschen, versorgen Sie bitte das Feld **EDIT-OPTIONS** in EDIT-IN mit dem entsprechenden Wert ("NOCORR", "LCASE", "NOHDR", "GETFC", "CFDATA", "GETIC", "EXTEND", "LOWEXT", "CFDLOW", "GFCLOW", "GFCCFD", "GFCCDL", "GFCEXT", "GFCEXL" oder "NOHDRL"). Dieses Feld wird im FHS-Mode (EDIT-MODE="F") nicht ausgewertet. Wenn Sie mit dem VTSUCB arbeiten muß hier "VTTCBU" angegeben werden.

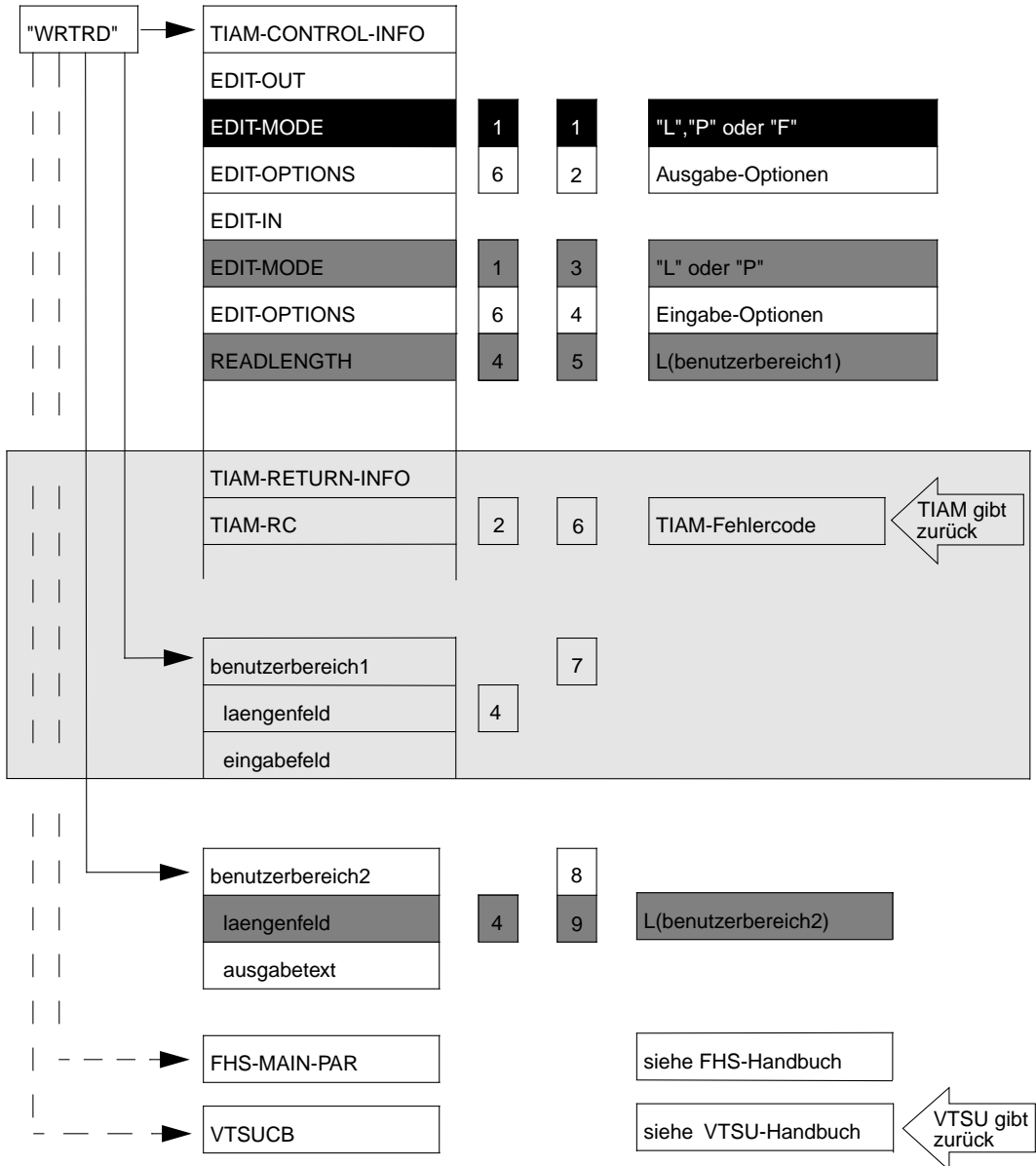
**TIAM gibt zurück:**

- 6 den TIAM-Fehlercode in dem Feld **TIAM-RC** (siehe unten). Hat er den Wert 0, ist der Aufruf fehlerfrei verlaufen.

Folgende Fehlercodes können beim Aufruf CALL "WRTRD" auftreten:

- |    |   |
|----|---|
| 0  | normale Beendigung  |
| 4  | nicht behebbarer Ein/Ausgabe-Fehler   |
| 8  | Parameterfehler   |
| 12 | Abschneiden beim Lesen  |
| 16 | Abschneiden beim Schreiben  |
| 20 | WRTRD wurde in einem Stapelprozeß aufgerufen  |
| 24 | Ende der Eingabe  |
| 32 | ein unzulässiger Parameter wurde vom System korrigiert                                    |
| 36 | Formatierung durch FHS nicht möglich, da FHS-Module fehlen                                |
| 40 | FHS-Rückkehrcode ungleich Null (siehe FHS-Rückkehrcode)                                   |
| 44 | Bei der Eingabe wurde das Zeichen NEW-LINE erkannt (nur bei 'extended line input')        |
| 48 | VTSU hat Fehler erkannt, siehe vollständigen RC im VTSUCB (nur bei Verwendung des VTSUCB) |

CALL	USING	Länge	Beschreibung	Versorgung der Felder
------	-------	-------	--------------	-----------------------



*Hinweis*

Bei intergriertem FHS-Einsatz (MODE="F") beachten Sie bitte beim Binden Ihres Anwendungsprogramms, daß der FHS-Modul **MFHSCALL** explizit mit einer INCLUDE-Anweisung zum COBOL-Programm dazugebunden werden muß. Fehlt beim WRTRD-Aufruf mit MODE="F" der FHS-Modul, weist TIAM den Aufruf mit einem Returncode zurück.

*Programmbeispiel: WRTRD im Line-Mode*

```

*****
* BEISPIEL FUER DEN  W R T R D          *
* MIT DEM AUFRUF  W R T R D  WIRD      *
* VON DER DATENSTATION EINE EINGABE   *
* ANGEFORDERT, DIE DANN MIT DER FUNKTION *
* WROUT WIEDER AUSGEGEBEN WIRD.       *
*****

IDENTIFICATION DIVISION.
PROGRAM-ID.    WRTEST.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*
*      COPY TIAMINFO.
*
*      COPY TIAMCTRC.
*
01  INAREA.
    41  MSGLENGTH PIC 9(5) COMP SYNC.
    41  TXT        PIC X(40) VALUE " ".
*
01  OUTAREA.
    41  MSGLENGTH PIC 9(5) COMP SYNC  VALUE 46.
    41  FILLER    PIC X.
    41  TXT        PIC X(40) VALUE
           "BITTE MAX. 40 BYTE LANGEN TEXT EINGEBEN.".
    41  STZ        PIC X.
*
01  OUTAREA-INPUT-TXT.
    41  MSGLENGTH PIC 9(5) COMP SYNC  VALUE 45.
    41  FILLER    PIC X.
    41  TXT        PIC X(40).
*
01  OUTAREA-ERROR.
    41  MSGLENGTH PIC 9(5) COMP SYNC  VALUE 25.
    41  FILLER    PIC X.
    41  TXT        PIC X(20) VALUE
           "FEHLERHAFTE EINGABE!".
*
PROCEDURE DIVISION.
ANF.
*
    MOVE "L"      TO EDIT-MODE    IN EDIT-OUT.
    MOVE "NOOPTS" TO EDIT-OPTIONS IN EDIT-OUT.
    MOVE "L"      TO EDIT-MODE    IN EDIT-IN.
    MOVE "LCASE"  TO EDIT-OPTIONS IN EDIT-IN.

```

```

MOVE 44          TO READLENGTH.
MOVE NEW-LINE   TO STZ  IN OUTAREA.
MOVE 1          TO COPYMEM-ID.
CALL "WRTRD"    USING   TIAM-CONTROL-INFO
                   OUTAREA INAREA.

*
IF  TIAM-RC IN TIAM-RETURN-INFO IS ZERO
  MOVE TXT IN INAREA TO TXT IN OUTAREA-INPUT-TXT
  CALL "WROUT" USING   TIAM-CONTROL-INFO
                   OUTAREA-INPUT-TXT
  ELSE
  CALL "WROUT" USING   TIAM-CONTROL-INFO
                   OUTAREA-ERROR.
STOP RUN.

```

*Programmbeispiel: WRTRD im Line-Mode mit Verwendung des VTSUCB*

```

*****
* BEISPIEL FUER DEN W R T R D *
* MIT DEM AUFRUF W R T R D WIRD *
* VON DER DATENSTATION EINE EINGABE *
* ANGEFORDERT, DIE DANN MIT DER FUNKTION *
* WROUT WIEDER AUSGEGEBEN WIRD. *
*****

```

```

IDENTIFICATION DIVISION.
PROGRAM-ID. WRTEST.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

```

```

*
*   COPY TIAMINFO.
*
*   COPY TIAMCTRC.
*
*   COPY VTSUCBC.
*
01 INAREA.
41 MSGLENGTH PIC 9(5) COMP SYNC.
41 TXT       PIC X(40) VALUE " ".
*
01 OUTAREA.
41 MSGLENGTH PIC 9(5) COMP SYNC VALUE 46.
41 FILLER    PIC X.
41 TXT       PIC X(40) VALUE
              "BITTE MAX. 40 BYTE LANGEN TEXT EINGEBEN.".
41 STZ       PIC X.
*
01 OUTAREA-INPUT-TXT.
41 MSGLENGTH PIC 9(5) COMP SYNC VALUE 45.
41 FILLER    PIC X.
41 TXT       PIC X(40).
*
01 OUTAREA-ERROR.
41 MSGLENGTH PIC 9(5) COMP SYNC VALUE 25.
41 FILLER    PIC X.
41 TXT       PIC X(20) VALUE
              "FEHLERHAFTE EINGABE!".

```

```

*
PROCEDURE DIVISION.
ANF.
*
MOVE "L"      TO MODUS      IN VTSUCB.
MOVE "Y"      TO LOWER-CHARS IN VTSUCB.
MOVE 44       TO READLENGTH.
MOVE NEW-LINE TO STZ      IN OUTAREA.
MOVE 1        TO COPYMEM-ID.
MOVE "VTCBU"  TO EDIT-OPTIONS IN EDIT-OUT.
CALL "WRTRD"  USING      TIAM-CONTROL-INFO
                        OUTAREA
                        INAREA
                        VTSUCB.
*
IF TIAM-RC IN TIAM-RETURN-INFO IS ZERO
MOVE TXT IN INAREA TO TXT IN OUTAREA-INPUT-TXT
MOVE "NOOPTS" TO EDIT-OPTIONS IN EDIT-OUT
CALL "WROUT" USING      TIAM-CONTROL-INFO
                        OUTAREA-INPUT-TXT
ELSE
MOVE "NOOPTS" TO EDIT-OPTIONS IN EDIT-OUT.
CALL "WROUT" USING      TIAM-CONTROL-INFO
                        OUTAREA-ERROR.
STOP RUN.

```

*Programmbeispiel: WRTRD mit FHS-Einsatz*

```

IDENTIFICATION DIVISION.
PROGRAM-ID. FHSTIAM.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SPECIAL-NAMES. TERMINAL IS DSS.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 USER-AREA.
   41 U-A-LAENGE          PIC 9(5) COMP.
   COPY BEISPIEL.
   COPY TIAMINFO.
   COPY FHSMAINP.
PROCEDURE DIVISION.
*
FHS-MODE
MOVE "F" TO EDIT-MODE IN EDIT-OUT.
MOVE 1 TO COPYMEM-ID.
MARK1.
*
FHS-BEREICHE AUSFUELLEN
MOVE "BEISPIEL" TO FHS-MAP-NAME
MOVE "Y" TO FHS-MAP-LIB-OPT
MOVE "TIAM.MAPLIB" TO FHS-MAP-LIB-NAME.
MARK2.
CALL "WRTRD" USING TIAM-CONTROL-INFO
                  USER-AREA
                  USER-AREA
                  FHS-MAIN-PAR.
IF FHS-MAIN-RC NOT = 0 GO TO FHSFEHLER.
IF TIAM-RC NOT = 0 GO TO TIAMFEHLER.
IF EINGABE = "ENDE" GO TO SCHLUSS.

```

```

*****
*
*   V E R A R B E I T U N G   D E R   E I N G A B E   *
*
*****
      GO TO MARK2.
FHSFEHLER.
      DISPLAY "FORMATIERUNGSFEHLER! RETURNCODE: "
              ERROR-CATEGORY " " ERROR-REASON UPON DSS.
      GO TO SCHLUSS.
TIAMFEHLER.
      DISPLAY "EIN/AUSGABEFehler! RETURNCODE: "
              TIAM-RC UPON DSS.
SCHLUSS.
      STOP RUN.

```

*Programmbeispiel: WRTRD im Extended-Line-Mode*

```

IDENTIFICATION DIVISION.
PROGRAM-ID.    WRTRDEX.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
      COPY TIAMINFO.
      COPY TIAMCTRC.
01  EINGABE.
      41 LAENGE      PIC 9(5) COMP SYNC.
      41 EINNAM      PIC X(12).
      41 EINVOR      PIC X(12).
      41 EINSTR      PIC X(25).
      41 EINPLZ      PIC X(4).
      41 EINORT      PIC X(12).
01  AUSGABE.
      41 LAENGE      PIC 9(5) COMP SYNC VALUE 223.
      41 FILLER      PIC X.
      41 F1NP        PIC X.
      41 F2SPA       PIC X.
      41 F3EM3       PIC X.
      41 TEXT1       PIC X(25) VALUE "BITTE GEBEN SIE NAMEN UND".
      41 TEXT1A      PIC X(16) VALUE " ANSCHRIFT EIN !".
      41 F4NL        PIC X.
      41 F5NL        PIC X.
      41 F6NOR       PIC X.
      41 TEXT2       PIC X(6) VALUE "NAME: ".
      41 F7EPA       PIC X.
      41 NAME        PIC X(12) VALUE SPACES.
      41 F8SPA       PIC X.
      41 TEXT3       PIC X(15) VALUE "          VORNAME: ".
      41 F9EPA       PIC X.
      41 VORNAME     PIC X(12) VALUE SPACES.
      41 F10NL       PIC X.
      41 F11SPA      PIC X.
      41 TEXT4       PIC X(9) VALUE "STRASSE: ".
      41 F12EPA      PIC X.
      41 STRASSE     PIC X(25) VALUE SPACES.
      41 F13NL       PIC X.
      41 F14SPA      PIC X.
      41 TEXT5       PIC X(9) VALUE "WOHNORT: ".
      41 F15EPA      PIC X.
      41 F16NUM      PIC X.

```



```
41 PLZ          PIC X(4) VALUE "0000".
41 F17SPA       PIC X.
41 BLNK         PIC X(2) VALUE "  ".
41 F18EPA       PIC X.
41 ORT          PIC X(12) VALUE SPACES.
41 F19NL        PIC X.
41 F20NL        PIC X.
41 F21SPA       PIC X.
41 TEXT6        PIC X(27) VALUE "BEI PROGRAMMENDE BITTE DEN ".
41 TEXT6A       PIC X(23) VALUE "NAMEN 'ENDE' EINGEBEN !".
```

PROCEDURE DIVISION.

ANF.

```
MOVE "L"        TO EDIT-MODE      IN EDIT-OUT.
MOVE "EXTEND"   TO EDIT-OPTIONS  IN EDIT-OUT.
MOVE "L"        TO EDIT-MODE      IN EDIT-IN.
MOVE "EXTEND"   TO EDIT-OPTIONS  IN EDIT-IN.
MOVE 223        TO READLENGTH.
MOVE 1          TO COPYMEM-ID.
MOVE NEW-LINE   TO F4NL F5NL F10NL F13NL F19NL F20NL.
MOVE NEW-PAGE   TO F1NP.
MOVE START-PROT-AREA TO F2SPA F8SPA F11SPA F14SPA F17SPA.
MOVE START-PROT-AREA TO F21SPA.
MOVE EMPH-LAYOUT3 TO F3EM3.
MOVE NORMAL-LAYOUT TO F6NOR.
MOVE END-PROT-AREA TO F7EPA F9EPA F12EPA F15EPA F18EPA.
MOVE START-NUM-DATA TO F16NUM.
```

AUSGEBEN.

```
CALL "WRTRD" USING TIAM-CONTROL-INFO AUSGABE EINGABE.
IF TIAM-RC IN TIAM-RETURN-INFO IS NOT ZERO GO TO ENDE.
IF EINNAM = "ENDE          " GO TO ENDE.
GO TO AUSGEBEN.
```

ENDE.

```
STOP RUN.
```



---

## 5 TIAM-FORTRAN-Schnittstelle

Die Definitionen der FORTRAN-Datenstruktur und der FORTRAN-Datenfelder entsprechen den Definitionen der COBOL-Datenstruktur und der COBOL-Datenfelder. In den folgenden Abschnitten werden somit nur Einzelheiten über die FORTRAN-Sprache und Syntax gegeben, die diese Datenstrukturen betreffen. Die vollständigen Informationen zu den Datenstrukturen und Datenfeldern finden Sie in der COBOL-Schnittstelle.

### *Hinweise*

- Wie Sie die Ein- und Ausgabeschnittstellen nutzen können, hängt von der Version des FORTRAN-Compilers und den eingegebenen COMOPT-Anweisungen ab.

Benutzen Sie den **FORTRAN-Compiler V2.1** mit `COMOPT EXTENDED=SYSTEM=NO` generiert der FORTRAN-Compiler eine Parameterliste in der der letzte Parameter dieser Liste durch ein auf binär 1 gesetztes Flag-Bit gekennzeichnet ist. Diese Art von Parameter-Übertragung wird durch das TIAM-Schnittstellen-Modul DCCOBRSTS unterstützt.

Benutzen Sie den **FORTRAN-Compiler V2.1** mit `COMOPT EXTENDED=SYSTEM=YES` generiert der FORTRAN-Compiler eine Parameterliste in der das Ende der Liste nicht durch ein auf binär 1 gesetztes Flag-Bit gekennzeichnet ist. Die Anzahl der Parameter wird durch das Register 0 angezeigt. Diese Art von Parameter-Übertragung wird durch eine ASSEMBLER-Schnittstelle definiert und nicht durch das TIAM-Schnittstellen-Modul DCCOBRSTS unterstützt.

```
EXTERNAL WRTRD
CALL OLDASS (WRTRD, TIAMCONTROLINFO, benutzerbereich1,
             benutzerbereich2 [,VTSUCB] [,FHSMAINPAR])
```

Benutzen Sie den **FORTRAN-Compiler V2.2**, wird eine Parameterliste generiert, wobei das Register 0 die Anzahl der Parameter und das Register 1 eine Liste der Parameter-Adressen enthält. Diese Art der Parameter-Übertragung wird ab TIAM V11.0 durch das TIAM-Schnittstellen-Modul DCCOBRSTS unterstützt.

- Die TIAM-Schnittstelle für FORTRAN ist ab TIAM V11.0A verfügbar.  
Die FHS-Schnittstelle für FORTRAN ist ab FHS V7.0 verfügbar.  
Die VTSU-Schnittstelle für FORTRAN ist ab VTSU V10.1A verfügbar.

## 5.1 Datenstrukturen für die TIAM-FORTRAN-Aufrufe

Zur Verständigung zwischen TIAM und dem Anwendungsprogramm dienen Datenstrukturen, die als Include-Elemente in einer Bibliothek stehen und in das Anwendungsprogramm kopiert werden.

### Datenstruktur VTSU-CONTROL-BLOCK

Die Datenstruktur VTSU-CONTROL-BLOCK wird mit folgender %INCLUDE Anweisung in das Anwendungsprogramm kopiert.

```
%INCLUDE $TSOS.SYSLIB.VTSU-B111(VTSUCBF)
```

VTSU-CONTROL-BLOCK ist im VTSU-Handbuch beschrieben (siehe [1] im Literaturverzeichnis).

### Datenstruktur LINE-MODE-CONTROL-CHARACTERS

Die Datenstruktur LINE-MODE-CONTROL-CHARACTER wird mit folgender %INCLUDE Anweisung in das Anwendungsprogramm kopiert.

```
%INCLUDE $TSOS.SYSLIB.VTSU-B.111(FORCTRC)
```

LINE-MODE-CONTROL-CHARACTER ist im VTSU-Handbuch beschrieben (siehe [1] im Literaturverzeichnis).

## Datenstruktur TIAM-CONTROL-INFO

Die Datenstruktur TIAM-CONTROL-INFO wird mit folgender %INCLUDE Anweisung in das Anwendungsprogramm kopiert:

```
%INCLUDE $TSOS.SYSLIB.TIAM.112(FORINFO)
```

Nachfolgend wird die Struktur des TIAM-CONTROL-INFO dargestellt. Eine vollständige Information zu den einzelnen Parametern finden Sie ab Seite 137.

```
*****
*          FORINFO          001          920320          TIAM          U *
*****
*
*          COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1991          *
*          ALL RIGHTS RESERVED          *
*
*****
*
*          TIAM CONTROL INFO
*
*          CHARACTER*64  TIAMCONTROLINFO
*
*          CHARACTER*16  EDITOUT          EDIT OUT
*
*          CHARACTER*1   EDITMODEOUT          EDIT MODE
*
*          'L'          - LINE MODE
*          'P'          - PHYSICAL MODE
*          'F'          - FHS MODE
*
*          CHARACTER*6   EDITOPTIONSOUT          EDIT OPTIONS
*
*          'NOOPTS'     - NO OPTIONS
*          'HCOPY'      - HARDCOPY
*          'HOMOUT'     - MOMOGENEOUS OUTPUT
*          'HEADER'     - SYSTEM HEADER
*          'ETB'        - ETB CLOSED
*          'INFO'       - INFORMATIVE MESSAGE
*          'TRNCD'      - TRANS CODE
*          'EXTEND'     - EXTENDED LINE OUTPUT
*          'NLOGC'      - NO LOG CONTROL
*          'BELL'       - BELL
*          'OWRITE'     - OVERWRITE
*          'BELHOM'     - BELL HOMOGENEOUS
*          'BELEXT'     - BELL EXTEND
*          'BELINF'     - BELL INFORMATIVE
*          'BELNLC'     - BELL NO LOG CONTROL
*          'HCNLC'     - HARDCOPY NO LOG CONTROL
*
*          CHARACTER*16  EDITIN          EDITIN
*
*          CHARACTER*1   EDITMODEIN          EDIT MODE
*
*          'L'          - LINE MODE
*          'P'          - PHYSICAL MODE
*
*          CHARACTER*6   EDITOPTIONSIN          EDIT OPTIONS
```

```

*          'NOOPTS' - NO-OPTIONS
*          'NOCORR' - NO-CORRECTION
*          'LCASE' - LOWER-CASE
*          'NOHDR' - DELETE-DEVICE-HEADER
*          'GETFC' - GET-FUNCTION-CODE
*          'CFDATA' - CONFIDENTIAL-DATA
*          'GETIC' - GET-ID-CARD
*          'EXTEND' - EXTENDED-LINE-INPUT
*          'LOWEXT' - LOWER-CASE-EXTEND
*          'CFDLOW' - CONF-DATA-LOW-CASE
*          'GFCCLOW' - FCT-CODE-LOW-CASE
*          'GFCCFD' - FCT-CODE-CONF-DATA
*          'GFCCDL' - FCT-CODE-CONF-DATA-LCASE
*          'GFCEXT' - FCT-CODE-EXTEND
*          'GFCEXL' - FCT-CODE-EXTEND-LCASE
*          'NOHDRL' - DELETE-DEVICE-HEADER-LCASE
*
*          READLENGTH
*
*          INTEGER*4 READLENGTH
*
*          CHARACTER*1 ISAMREQUEST
*          CHARACTER*1 ISAMREQUEST
*          'N' - NO KEY
*          'K' - GET KEY
*          'L' - GET KEY LENGTH
*          'P' - GET KEY POSITION
*          'B' - GET KEY POSITION LENGTH
*
*          ASSIGNMENT REQUEST
*
*          CHARACTER*1 ASSIGNREQUEST
*          'N' - NO ASSIGNMENT CODE
*          'G' - GET ASSIGNMENT CODE
*
*          COPYMEM ID
*
*          INTEGER*2 COPYMEMID / 1 /
*          TIAM RETURN INFO
*
*          CHARACTER*16 TIAMRETURNINFO
*          TIAM RC
*
*          INTEGER*2 TIAMRC
*          ASSIGNMENT
*
*          INTEGER*2 ASSIGNMENT
*          KEY POSITION
*
*          INTEGER*2 KEYPOSITION
*          KEY LENGTH
*
*          INTEGER*2 KEYLENGTH
*
*          *****
*
*          CHARACTER*1 LINEMODE / 'L' /
*          CHARACTER*1 PHYISCALMODE / 'P' /
*          CHARACTER*1 FHSMODE / 'F' /
*
*          CHARACTER*6 NOOPTIONS / 'NOOPTS' /
*          CHARACTER*6 HARDCOPY / 'HCOPY' /
*          CHARACTER*6 HOMOGENEOUSOUT / 'HOMOUT' /
*          CHARACTER*6 SYSTEMHEADER / 'HEADER' /
*          CHARACTER*6 ETBCLOSED / 'ETB' /
*          CHARACTER*6 INFORMATIVEMESS / 'INFO' /
*          CHARACTER*6 TRANSCODE / 'TRNCD' /
*          CHARACTER*6 EXTENDEDLINEOUT / 'EXTEND' /
*          CHARACTER*6 NOLOGCONTROL / 'NLOGC' /
*          CHARACTER*6 BELL / 'BELL' /

```

```

CHARACTER*6  OVERWRITE           / 'OWRITE' /
CHARACTER*6  BELLHOMOGENEOUS     / 'BELHOM' /
CHARACTER*6  BELLEXTEND          / 'BELEXT' /
CHARACTER*6  BELLINFORMATIVE     / 'BELINF' /
CHARACTER*6  BELLNOLOGCONTRL     / 'BELNLC' /
CHARACTER*6  HARDCOPYNOLOGCT     / 'HCNLC ' /
*
CHARACTER*6  NOCORRECTION        / 'NOCORR' /
CHARACTER*6  LOWERCASE           / 'LCASE ' /
CHARACTER*6  DELETEDEVICEHDR     / 'NOHDR ' /
CHARACTER*6  GETFUNCTIONCODE     / 'GETFC ' /
CHARACTER*6  CONFIDENTIALDTA     / 'CFDATA' /
CHARACTER*6  GETIDCARD           / 'GETIC ' /
CHARACTER*6  EXTENDEDLINEINP     / 'EXTEND' /
CHARACTER*6  LOWERCASEEXTEND     / 'LOWEXT' /
CHARACTER*6  CONFDATALOWCASE     / 'CFDLOW' /
CHARACTER*6  FCTCODELOWCASE      / 'GFCLOW' /
CHARACTER*6  FCTCODECONFDATA     / 'GFCCFD' /
CHARACTER*6  FCTCODECONFDTLC     / 'GFCCDL' /
CHARACTER*6  FCTCODEEEXTEND      / 'GFCEXT' /
CHARACTER*6  FCTCODEEEXTENDLC    / 'GFCEXL' /
CHARACTER*6  DELETEDEVHDRCL     / 'NOHDRL' /
*
CHARACTER*1  NOKEY               / 'N' /
CHARACTER*1  GETKEY              / 'K' /
CHARACTER*1  GETKEYLENGTH        / 'L' /
CHARACTER*1  GETKEYPOSITION      / 'P' /
CHARACTER*1  GETKEYPOSLENGTH     / 'B' /
*
CHARACTER*1  NOASSIGNCODE        / 'N' /
CHARACTER*1  GETASSIGNCODE       / 'G' /
*****
*
EQUIVALENCE (TIAMCONTROLINFO ( 1: 16), EDITOUT)
EQUIVALENCE (EDITOUT          ( 1: 1), EDITMODEOUT)
EQUIVALENCE (EDITOUT          ( 3: 8), EDITOPTIONSOUT)
EQUIVALENCE (TIAMCONTROLINFO (17: 32), EDITIN)
EQUIVALENCE (EDITIN          ( 1: 1), EDITMODEIN)
EQUIVALENCE (EDITIN          ( 3: 8), EDITOPTIONSIN)
EQUIVALENCE (TIAMCONTROLINFO (33: 36), READLENGTH)
EQUIVALENCE (TIAMCONTROLINFO (37: 37), ISAMREQUEST)
EQUIVALENCE (TIAMCONTROLINFO (41: 41), ASSIGNREQUEST)
EQUIVALENCE (TIAMCONTROLINFO (45: 46), COPYMEMID)
EQUIVALENCE (TIAMCONTROLINFO (49: 64), TIAMRETURNINFO)
EQUIVALENCE (TIAMRETURNINFO  ( 1: 2), TIAMRC)
EQUIVALENCE (TIAMRETURNINFO  ( 5: 6), ASSIGNMENT)
EQUIVALENCE (TIAMRETURNINFO  ( 9: 10), KEYPOSITION)
EQUIVALENCE (TIAMRETURNINFO  (11: 12), KEYLENGTH)
*****

```

## 5.2 TIAM-FORTRAN-Aufrufe

### 5.2.1 Daten lesen von SYSDTA: CALL RDATA

Mit dem Aufruf CALL RDATA lesen Sie einen Datensatz von SYSDTA. Die Datei, die als SYSDTA erklärt ist, kann eine katalogisierte SAM- oder ISAM-Datei, ein Diskettengerät, ein Element einer PLAM-Bibliothek, eine SDF-P-Variable oder die prozeßführende Datensichtstation sein.

#### Format des Aufrufs:

```
EXTERNAL RDATA
CALL RDATA (TIAMCONTROLINFO, benutzerbereich [,VTSUCB])
```

#### Programmbeispiel RDATA

```
*****
* Beispiel für den R D A T A in FORTRAN-Programmen. *
* Sie werden zu einer Eingabe aufgefordert, die anschließend *
* ausgegeben wird. *
* *
*****

PROGRAM FORRDATA
*
CHARACTER*24 IOAREA
INTEGER*4 DTLNGT
CHARACTER*20 TEXT
EQUIVALENCE (IOAREA (1:4), DTLNGT)
EQUIVALENCE (IOAREA (5:24), TEXT)
%INCLUDE $TSOS.SYSLIB.TIAM.112(FORINFO)
%INCLUDE $TSOS.SYSLIB.VTSU-B.111(VTSUCBF)
*
EXTERNAL RDATA
*
READLENGTH = 24
EDITOPTIONSIN = 'VTCBU'
CALL RDATA(TIAMCONTROLINFO, IOAREA,VTSUCB)
WRITE(*,*) 'TIAMRC : ',TIAMRC
WRITE(*,*) 'VTSU MRC : ',MAINRETCODE
WRITE(*,*) 'TEXTE : ',TEXT
WRITE(*,*) 'DATA LEN. : ',DTLNGT
END
```



## 5.2.2 Daten ausgeben nach SYSOUT: CALL WROUT

Mit dem Aufruf CALL WROUT geben Sie eine Nachricht nach SYSOUT aus. Die Datei, die als SYSOUT erklärt ist, kann eine katalogisierte SAM- oder ISAM-Datei, ein Element einer PLAM-Bibliothek, eine SDF-P-Variable oder die prozeßführende Datenstation sein.

Als Sonderfunktion ist im Aufruf CALL WROUT der Formateinsatz mit FHS integriert. FHS wird dabei intern von TIAM aufgerufen.

### Format des Aufrufs:

```
EXTERNAL WROUT
CALL WROUT (TIAMCONTROLINFO, benutzerbereich [,VTSUCB] [,FHSMAINPAR])
```

### Programmbeispiel WROUT

```
*****
* Beispiel für den W R O U T in FORTRAN-Programmen. *
* Es werden ein formatierter Bildschirm (FHS-Mode), die TIAM und*
* FHS Returncodes und die ausgefüllten Felder (Line-Mode) *
* ausgegeben. *
*****
```

```
PROGRAM FORWROUT
*
  INTEGER * 4 DTLNGT
  %INCLUDE FORLIB(FWROUT)
  %INCLUDE FORLIB(FHSMAINP)
  %INCLUDE $TSOS.SYSLIB.TIAM.112(FORINFO)
*
  EXTERNAL WROUT
*
  FHSMAPNAME      = 'FWROUT  '
  FHSRESTARTOPT1 = 'Y'
  FHSMAPLIBOPT   = 'Y'
  FHSMAPLIBNAME  = 'FORLIB'
  EDITMODEOUT    = 'F'
  CALL WROUT(TIAMCONTROLINFO, IOAREA, FHSMAINPAR)
  WRITE(*,*) 'RCCATEGO      : ',FORRCCATEGO
  WRITE(*,*) 'RCREASON       : ',FORRCREASON
  WRITE(*,*) 'TIAMRC          : ',TIAMRC
  WRITE(*,*) 'FHSMAINRC      : ',FHSMAINRC
  WRITE(*,*) 'ERCATEGO       : ',ERRORCATEGORY
  WRITE(*,*) 'ERREASON        : ',ERRORREASON
  END
```

### 5.2.3 Kombinierte Aus- und Eingabe: CALL WRTRD

Mit dem Aufruf CALL WRTRD geben Sie eine Nachricht an einer Datensichtstation aus und lesen anschließend eine Nachricht von der Datensichtstation. Außer der zur Datensichtstation übertragenen Nachricht erscheint dabei kein weiteres Eingabeaufforderungszeichen. Als Sonderfunktion ist im Aufruf CALL WRTRD der Formateinsatz mit FHS integriert. FHS wird dabei intern von TIAM aufgerufen.

#### Format des Aufrufs:

```
EXTERNAL WRTRD
CALL WRTRD (TIAMCONTROLINFO, benutzerbereich1, benutzerbereich2
           [,VTSUCB] [,FHSMAINPAR])
```

#### Programmbeispiel WRTRD

```
*****
* Beispiel für den W R T R D in FORTRAN-Programmen. *
* Zuerst wird ein formatierter Bildschirm (FHS-Mode) ausgegeben,*
* danach eine Eingabe von der Datensichtstation angefordert und *
* anschließend werden die TIAM und *
* FHS Returncodes und die ausgefüllten Felder (Line-Mode) *
* ausgegeben. *
*****

PROGRAM FORWRTRD
*
INTEGER * 4 DTLNGT
%INCLUDE FORLIB(FWRTRD)
%INCLUDE FORLIB(FHSMAINP)
%INCLUDE $TSOS.SYSLIB.TIAM.112(FORINFO)
*
EXTERNAL WRTRD
*
FHSMAPNAME = 'FWRTRD '
FHSRESTARTOPT1 = 'Y'
FHSMAPLIBOPT = 'Y'
FHSMAPLIBNAME = 'FORLIB'
EDITMODEOUT = 'F'
CALL WRTRD(TIAMCONTROLINFO, IOAREA, IOAREA, FHSMAINPAR)
WRITE(*,*) 'RCCATEGO : ',FORRCCATEGO
WRITE(*,*) 'RCREASON : ',FORRCREASON
WRITE(*,*) 'TIAMRC : ',TIAMRC
WRITE(*,*) 'FHSMAINRC : ',FHSMAINRC
WRITE(*,*) 'ERCATEGO : ',ERRORCATEGORY
WRITE(*,*) 'ERREASON : ',ERRORREASON
WRITE(*,*) 'FIELD : ',FIELD
END
```

---

## 6 TIAM-PL/I-Schnittstelle

Die Definitionen der PL/I-Datenstruktur und der PL/I-Datenfelder entsprechen den Definitionen der COBOL-Datenstruktur und der COBOL-Datenfelder. In den folgenden Abschnitten werden somit nur Einzelheiten über die PL/I-Sprache und Syntax gegeben, die diese Datenstrukturen betreffen. Die vollständigen Informationen zu den Datenstrukturen und Datenfeldern finden Sie in der COBOL-Schnittstelle.

### *Hinweis*

- Die TIAM-Schnittstelle für PL/I ist ab TIAM V11.0A verfügbar.  
Die FHS-Schnittstelle für PL/I ist ab FHS V7.0 verfügbar.  
Die VTSU-Schnittstelle für PL/I ist ab VTSU V10.1A verfügbar.

## 6.1 Datenstrukturen für die TIAM-PL/I-Aufrufe

Zur Verständigung zwischen TIAM und dem Anwendungsprogramm dienen Datenstrukturen, die als Include-Elemente in einer Bibliothek stehen und in das Anwendungsprogramm kopiert werden.

### Datenstruktur VTSU-CONTROL-BLOCK

Die Datenstruktur VTSU-CONTROL-BLOCK wird mit folgender %INCLUDE Anweisung in das Anwendungsprogramm kopiert.

```
%INCLUDE $TSOS.SYSLIB.VTSU-B.111(VTSUCBP)
```

Die Datenstruktur VTSU-CONTROL-BLOCK ist im VTSU-Handbuch beschrieben (siehe [1] im Literaturverzeichnis).

### Datenstruktur LINE-MODE-CONTROL-CHARACTERS

Die Datenstruktur LINE-MODE-CONTROL-CHARACTER wird mit folgender %INCLUDE Anweisung in das Anwendungsprogramm kopiert.

```
%INCLUDE $TSOS.SYSLIB.VTSU-B.111(PL1CTRC)
```

Eine vollständige Information zu den einzelnen Parametern finden Sie im VTSU-Handbuch (siehe [1] im Literaturverzeichnis).

## Datenstruktur TIAM-CONTROL-INFO

Die Datenstruktur TIAM-CONTROL-INFO wird mit folgender %INCLUDE Anweisung in das Anwendungsprogramm kopiert:

```
%INCLUDE $TSOS.SYSLIB.TIAM.111(PL1INFO)
```

Nachfolgend wird die Struktur des TIAM-CONTROL-INFO dargestellt. Eine vollständige Information zu den einzelnen Parametern finden Sie ab Seite 137.

```

/*****
/*      PL1INFO          001          920320          TIAM          U */
/*****
/*
/*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1991 */
/*                  ALL RIGHTS RESERVED                          */
/*                                                                */
/*****
/* ***** Edit mode *****
/* Possible values for EDIT_OUT.EDIT_MODE / EDIT_IN.EDIT_MODE */
DCL LINE_MODE      CHAR (1) STATIC INIT ('L');
DCL PHYSICAL_MODE  CHAR (1) STATIC INIT ('P');

/* Possible values for EDIT_OUT.EDIT_MODE */
DCL FHS_MODE      CHAR (1) STATIC INIT ('F');

/* ***** Edit options *****
/* Possible values for EDIT_OUT.EDIT_OPTIONS / EDIT_IN.EDIT_OPTIONS */
DCL NO_OPTIONS     CHAR (6) STATIC INIT ('NOOPTS');
DCL VTSUCB_USED   CHAR (6) STATIC INIT ('VTCBU ');

/* Possible values for EDIT_OUT.EDIT_OPTIONS */
DCL HARDCOPY      CHAR (6) STATIC INIT ('HCOPY ');
DCL HOMOGENEOUS_OUTPUT CHAR (6) STATIC INIT ('HOMOUT');
DCL SYSTEM_HEADER CHAR (6) STATIC INIT ('HEADER ');
DCL ETB_CLOSED    CHAR (6) STATIC INIT ('ETB  ');
DCL INFORMATIVE_MESSAGE CHAR (6) STATIC INIT ('INFO ');
DCL TRANS_CODE    CHAR (6) STATIC INIT ('TRNCD ');
DCL EXTENDED_LINE_OUTPUT CHAR (6) STATIC INIT ('EXTEND ');
DCL NO_LOG_CONTROL CHAR (6) STATIC INIT ('NLOGC ');
DCL BELL          CHAR (6) STATIC INIT ('BELL ');
DCL OVERWRITE     CHAR (6) STATIC INIT ('OWRITE');
DCL BELL_HOMOGENEOUS CHAR (6) STATIC INIT ('BELHOM');
DCL BELL_EXTEND   CHAR (6) STATIC INIT ('BELEXT');
DCL BELL_INFORMATIVE CHAR (6) STATIC INIT ('BELINF');
DCL BELL_NO_LOG_CONTROL CHAR (6) STATIC INIT ('BELNLC');
DCL HARDCOPY_NO_LOG_CONTROL CHAR (6) STATIC INIT ('HCNLC ');

/* Possible values for EDIT_IN.EDIT_OPTIONS */
DCL NO_CORRECTION CHAR (6) STATIC INIT ('NOCORR');
DCL LOWER_CASE    CHAR (6) STATIC INIT ('LCASE ');
DCL DELETE_DEVICE_HEADER CHAR (6) STATIC INIT ('NOHDR ');
DCL GET_FUNCTION_CODE CHAR (6) STATIC INIT ('GETFC ');
DCL CONFIDENTIAL_DATA CHAR (6) STATIC INIT ('CFDATA');
DCL GET_ID_CARD   CHAR (6) STATIC INIT ('GETIC ');
DCL EXTENDED_LINE_INPUT CHAR (6) STATIC INIT ('EXTEND ');
DCL LOWER_CASE_EXTEND CHAR (6) STATIC INIT ('LOWEXT ');
DCL CONF_DATA_LOW_CASE CHAR (6) STATIC INIT ('CFDLOW');
DCL FCT_CODE_LOW_CASE CHAR (6) STATIC INIT ('GFLOW');
DCL FCT_CODE_CONF_DATA CHAR (6) STATIC INIT ('GFCCFD');
DCL FCT_CODE_CONF_DATA_LCASE CHAR (6) STATIC INIT ('GFCCDL');
DCL FCT_CODE_EXTEND CHAR (6) STATIC INIT ('GFCEXT');

```

```

DCL FCT_CODE_EXTEND_LCASE      CHAR (6) STATIC INIT ('GFCEXL');
DCL DELETE_DEVICE_HEADER_LCASE CHAR (6) STATIC INIT ('NOHDRL');

/* ***** ISAM informations ***** */
/* Possible values for ISAM_REQUEST */
DCL NO_KEY                      CHAR (1) STATIC INIT ('N');
DCL GET_KEY                     CHAR (1) STATIC INIT ('K');
DCL GET_KEY_LENGTH              CHAR (1) STATIC INIT ('L');
DCL GET_KEY_POSITION            CHAR (1) STATIC INIT ('P');
DCL GET_KEY_POSITION_LENGTH     CHAR (1) STATIC INIT ('B');

/* ***** ASSIGNMENT informations ***** */
/* ASSIGNMENT informations */
/* Possible values for ASSIGNMENT_REQUEST */
DCL NO_ASSIGNMENT_CODE          CHAR (1) STATIC INIT ('N');
DCL GET_ASSIGNMENT_CODE         CHAR (1) STATIC INIT ('G');

/* ***** Call parameter list ***** */
DECLARE
01 TIAM_CONTROL_INFO,
   02 EDIT_OUT,
      03 EDIT_MODE              CHAR,
      03 FILL1                  CHAR,
      03 EDIT_OPTIONS           CHAR(6),
      03 FIL8                    CHAR(8),
   02 EDIT_IN,
      03 EDIT_MODE              CHAR,
      03 FILL1                  CHAR,
      03 EDIT_OPTIONS           CHAR(6),
      03 FIL8                    CHAR(8),
   02 READLENGTH                BIN FIXED(31) ALIGNED,
   02 ISAM_REQUEST              CHAR,
   02 FIL3                       CHAR(3),
   02 ASSIGNMENT_REQUEST        CHAR,
   02 FILL3                      CHAR(3),
   02 COPYMEM_ID                BIN FIXED(15) ALIGNED INIT(1),
   02 FIL2                       CHAR(2),
   02 TIAM_RETURN_INFO,
      03 TIAM_RC                 BIN FIXED(15) ALIGNED INIT(0),
      03 FIL2                    CHAR(2),
      03 ASSIGNMENT              BIN FIXED(15) ALIGNED,
      03 FILL2                   CHAR(2),
      03 KEY_POSITION            BIN FIXED(15) ALIGNED,
      03 KEY_LENGTH              BIN FIXED(15) ALIGNED,
      03 FIL4                    CHAR(4);

```

## 6.2 TIAM-PL/I-Aufrufe

### 6.2.1 Daten lesen von SYSDTA: CALL RDATA

Mit dem Aufruf CALL RDATA lesen Sie einen Datensatz von SYSDTA. Die Datei, die als SYSDTA erklärt ist, kann eine katalogisierte SAM- oder ISAM-Datei, ein Diskettengerät, ein Element einer PLAM-Bibliothek, eine SDF-P-Variable oder die prozeßführende Datensichtstation sein.

#### Format des Aufrufs:

```
DCL RDATA ENTRY EXTERNAL OPTIONS (ASSEMBLER);
CALL RDATA (TIAM_CONTROL_INFO, benutzerbereich [,VTSUCB]);
```

#### Programmbeispiel RDATA

```

/*****
/* Beispiel für den R D A T A in PL/I-Programmen.          */
/* Sie werden zu einer Eingabe aufgefordert, die anschließend */
/* ausgegeben wird.                                       */
/*                                                         */
/*****
PLIRDATA: PROC OPTIONS(MAIN);

    %INCLUDE $TSOS.SYSLIB.TIAM.112(PLIINFO)
    %INCLUDE $TSOS.SYSLIB.VTSU-B.111(VTSUCBP)

    DCL RDATA ENTRY EXTERNAL OPTIONS (ASSEMBLER);
    DCL      1 IOAREA,
              19 DTLNGT    BIN FIXED(31),
              19 TEXT      CHAR (20);

    IOAREA.TEXT                = ' ';
    TIAM_CONTROL_INFO.READLENGTH = 25;
    TIAM_CONTROL_INFO.EDIT_IN.EDIT_OPTIONS = VTSUCB_USED;
    CALL RDATA(TIAM_CONTROL_INFO,IOAREA,VTSUCB);
    DISPLAY ('TIAMRC      : ' || CHAR(TIAM_RC));
    DISPLAY ('VTSUCB MRC : ' || CHAR(MAINCODE));
    DISPLAY ('VTSUCB SRC : ' || CHAR(SUBCODES));
    DISPLAY ('TEXT LEN.  : ' || CHAR(DTLNGT));
    DISPLAY ('TEXT      : ' || TEXT);
END;
```

## 6.2.2 Daten ausgeben nach SYSOUT: CALL WROUT

Mit dem Aufruf CALL WROUT geben Sie eine Nachricht nach SYSOUT aus. Die Datei, die als SYSOUT erklärt ist, kann eine katalogisierte SAM- oder ISAM-Datei, ein Element einer PLAM-Bibliothek, eine SDF-P-Variable oder die prozeßführende Datenstation sein. Als Sonderfunktion ist im Aufruf CALL WROUT der Formateinsatz mit FHS integriert. FHS wird dabei intern von TIAM aufgerufen.

### Format des Aufrufs:

```
DCL WROUT ENTRY EXTERNAL OPTIONS (ASSEMBLER);
CALL WROUT (TIAM_CONTROL_INFO, benutzerbereich [,VTSUCB][,FHSMAINPAR]);
```

### Programmbeispiel WROUT

```

/*****
/* Beispiel für den W R O U T in PL/I-Programmen.          */
/* Es wird ein formatierter Bildschirm (FHS-Mode), die TIAM und */
/* FHS Returncodes und die ausgefüllten Felder (Line-Mode)    */
/* ausgegeben.                                               */
/*****
PL1WROUT: PROC OPTIONS(MAIN);

    DCL WROUT ENTRY EXTERNAL OPTIONS (ASSEMBLER);
    DCL 1 IOAREA,
        19 DTLNGT BIN FIXED(31),
        %INCLUDE PL1LIB(PWROUT);
    DCL IOPR BIT(528) DEFINED IOAREA;
        %INCLUDE PL1LIB(FHSMAINP);
        %INCLUDE $TSOS.SYSLIB.TIAM.112(PL1INFO);

    IOPR          = (528)'0'B;
    FHS_MAP_NAME  = 'PWROUT';
    FHS_RESTART_OPT1 = 'Y';
    FHS_MAP_LIB_OPT = 'Y';
    FHS_MAP_LIB_NAME = 'PL1LIB';
    EDIT_OUT.EDIT_MODE = 'F';
    CALL WROUT(TIAM_CONTROL_INFO, IOAREA, FHS_MAIN_PAR);
    DISPLAY ('TIAMRC : ' || CHAR(TIAM_RC));
    DISPLAY ('FHSMAINRC : ' || CHAR(FHS_MAIN_RC));
    DISPLAY ('ERCATEGO : ' || CHAR(ERROR_CATEGORY));
    DISPLAY ('ERREASON : ' || CHAR(ERROR_REASON));
    DISPLAY ('FIELD : ' || FIELD);
END;
```



## 6.2.3 Kombinierte Aus- und Eingabe: CALL WRTRD

Mit dem Aufruf CALL WRTRD geben Sie eine Nachricht an einer Datensichtstation aus und lesen anschließend eine Nachricht von der Datensichtstation. Außer der zur Datensichtstation übertragenen Nachricht erscheint dabei kein weiteres Eingabeaufforderungszeichen. Als Sonderfunktion ist im Aufruf CALL WRTRD der Formateinsatz mit FHS integriert. FHS wird dabei intern von TIAM aufgerufen.

### Format des Aufrufs:

```
DCL WRTRD ENTRY EXTERNAL OPTIONS (ASSEMBLER);
CALL WRTRD (TIAM_CONTROL_INFO, benutzerbereich1, benutzerbereich2
           [,VTSUCB] [,FHS-MAIN-PAR]);
```

### Programmbeispiel WRTRD

```

/*****
/* Beispiel für den W R T R D in PL/I-Programmen.
/* Zuerst wird ein formatierter Bildschirm (FHS-Mode) ausgegeben,*/
/* danach eine Eingabe von der Datensichtstation angefordert und */
/* anschließend werden die TIAM und
/* FHS Returncodes und die ausgefüllten Felder (Line-Mode)
/* ausgegeben.
/*****
PL1WRTRD: PROC OPTIONS(MAIN);

DCL WRTRD ENTRY EXTERNAL OPTIONS (ASSEMBLER);
DCL 1 IOAREA,
    19 DTLNGT BIN FIXED(31),
    %INCLUDE PL1LIB(PWRTRD);
DCL IOPR BIT(528) DEFINED IOAREA;
    %INCLUDE PL1LIB(FHSMAINP);
    %INCLUDE $TSOS.SYSLIB.TIAM.112(PL1INFO);

IOPR          = (528)'0'B;
FHS_MAP_NAME  = 'PWRTRD';
FHS_RESTART_OPT1 = 'Y';
FHS_MAP_LIB_OPT = 'Y';
FHS_MAP_LIB_NAME = 'PL1LIB';
EDIT_OUT.EDIT_MODE = 'F';
CALL WRTRD(TIAM_CONTROL_INFO, IOAREA, IOAREA, FHS_MAIN_PAR);
DISPLAY ('TIAMRC : ' || CHAR(TIAM_RC));
DISPLAY ('FHSMAINRC : ' || CHAR(FHS_MAIN_RC));
DISPLAY ('ERCATEGO : ' || CHAR(ERROR_CATEGORY));
DISPLAY ('ERREASON : ' || CHAR(ERROR_REASON));
DISPLAY ('FIELD : ' || FIELD);
END;
```



---

## 7 TIAM-C-Schnittstelle

Die Definitionen der C-Datenstruktur und der C-Datenfelder entsprechen den Definitionen der COBOL-Datenstruktur und der COBOL-Datenfelder. In den folgenden Abschnitten werden somit nur Einzelheiten über die C-Sprache und Syntax gegeben, die diese Datenstrukturen betreffen. Die vollständigen Informationen zu den Datenstrukturen und Datenfeldern finden Sie bei der Beschreibung der COBOL-Schnittstelle.

### *Hinweise*

- Bei der Benutzung von Ein-/Ausgabeschnittstellen müssen Sie die folgenden Punkte berücksichtigen:
  1. Wenn die C-Compiler-Version, die Sie verwenden, nicht die ILCS-Konventionen unterstützt, müssen Sie mit einem kleinen ASSEMBLER-Schnittstellenmodul das Parameterlisten-Format erzeugen, das von TIAM unterstützt wird (siehe nachfolgende Beispiele). Diese ASSEMBLER-Schnittstelle setzt das höherwertige Bit des letzten Parameters in der Parameterliste auf binär 1.

Beispiel der ASSEMBLER-Schnittstelle für WROUT mit VTSUCB :

#### a) C Programm

```
extern cwrou3 ();
cwrou3 (&tiam_control_info, &output_area, &vtstu_cb);
```

#### b) ASSEMBLER Modul

```
AWROUT   START
AWROUT   AMODE ANY
AWROUT   RMODE ANY
CWROUT3  C$ENT ((TIAMCI,A),(DTA,A),(VTSUA,A))
          NI   TIAMCI,X'7F'   TIAM-CONTROL-INFO  adresse
          NI   DTA,X'7F'     DATA  adresse
          OI   VTSUA,X'80'   VTSUCB  adresse
          L    R15,WROUT@
          BASR R14,R15
          C$EX
WROUT@   DC   V(WROUT)
          END
```

2. Wenn die C-Compiler-Version, die Sie verwenden, die ILCS-Konventionen unterstützt, müssen Sie das oben beschriebene ASSMBLER-Modul nicht verwenden. Sie können dann direkt Aufrufe an TIAM übergeben, wie in den Beispielen im Abschnitt „TIAM-C-Aufrufe“ (siehe ab Seite 192) dargestellt, .
- Die TIAMINFO-Schnittstelle für C ist ab TIAM V11.0A verfügbar.
  - Die VTSUCB- und TIAMCTRC-Schnittstellen für C sind ab VTSU V10.1A verfügbar.
  - Es gibt derzeit keine FHS-Schnittstelle für C.

## 7.1 Datenstrukturen für die TIAM-C-Aufrufe

Zur Verständigung zwischen TIAM und dem Anwendungsprogramm dienen Datenstrukturen, die als Include-Elemente in einer Bibliothek stehen und in das Anwendungsprogramm kopiert werden.

### Datenstruktur VTSU-CONTROL-BLOCK

Die Datenstruktur LINE-MODE-CONTROL-CHARACTERS wird mit folgendem Aufruf in das Anwendungsprogramm kopiert.

```
#include <vtsucb.h>
```

Dieses Include-Element befindet sich in der Bibliothek \$TSOS.SYSLIB.VTSU-B.111.

Die Datenstruktur VTSU-CONTROL-BLOCK ist im VTSU-Handbuch beschrieben (siehe [1] im Literaturverzeichnis).

### Datenstruktur LINE-MODE-CONTROL-CHARACTERS

Die Datenstruktur LINE-MODE-CONTROL-CHARACTERS wird mit folgendem Aufruf in das Anwendungsprogramm kopiert.

```
#include <cctrc.h>
```

Dieses Include-Element befindet sich in der Bibliothek \$TSOS.SYSLIB.VTSU-B.111.

Eine vollständige Information zu den einzelnen Parametern finden Sie im VTSU-Handbuch (siehe [1] im Literaturverzeichnis).

## Datenstruktur TIAM-CONTROL-INFO

Die Datenstruktur TIAM-CONTROL-INFO wird mit folgendem Aufruf in das Anwendungsprogramm kopiert:

```
#include <info.h>
```

Dieses Include-Element befindet sich in der Bibliothek \$TSOS.SYSLIB.TIAM.112.

Nachfolgend wird die Struktur des TIAM-CONTROL-INFO dargestellt. Eine vollständige Information zu den einzelnen Parametern finden Sie ab Seite 137.

```

/*****
/* TIAMINFO.H      211      950320      TIAM      U */
/*****
/*
/*      COPYRIGHT (C) SIEMENS NIXDORF INFORMATIONSSYSTEME AG 1991      */
/*      ALL RIGHTS RESERVED      */
/*      */
/*****

#ifndef WAS_TIAMINFO
#else
#define WAS_TIAMINFO
/***** TIAM RC values      *****/
/* Possible values for TIAM_RETURN_INFO.TIAM_RC      */
#define TIAM_NO_ERROR      0
#define TIAM_IO_ERROR      4
#define TIAM_PARAMETER_ERROR      8
#define TIAM_RECORD_TRUNC_READ      12
#define TIAM_RECORD_TRUNC_WRITE      16
#define TIAM_WRTRD_CALLED_BATCH      20
#define TIAM_END_OF_INPUT      24
#define TIAM_PARAMETER_CORRECTED      32
#define TIAM_FHS_MISSING      36
#define TIAM_FHS_ERROR      40
#define TIAM_NEWLINE_DETECTED      44
#define TIAM_VTSU_ERROR      48

/***** Edit mode      *****/
/* Possible values for EDIT_OUT.EDIT_MODE / EDIT_IN.EDIT_MODE      */
#define LINE_MODE      'L'
#define PHYSICAL_MODE      'P'

/* Possible values for EDIT_OUT.EDIT_MODE      */
#define FHS_MODE      'F'

/***** Edit options      *****/
/* The edit options must be moved in corresponding parameter fields      */
/* (edit_in.edit_options, edit_out.edit_options) thanks to STRNCPY      */
/* function.      */
/* EX: strncpy (edit_in.edit_options, "LCASE ",      */
/*      sizeof (edit_in.edit_options));      */

/* Possible values for EDIT_OUT.EDIT_OPTIONS / EDIT_IN.EDIT_OPTIONS      */

```

```

/* NO EDIT OPTIONS SPECIFIED                "NOOPTS"
   VTSUCB_USED                              "VTCBU "                                */

/* Possible values for EDIT_OUT.EDIT_OPTIONS
   HARDCOPY                                 "HCOPY "
   HOMOGENEOUS_OUTPUT                      "HOMOUT"
   SYSTEM_HEADER                           "HEADER"
   ETB_CLOSED                              "ETB "
   INFORMATIVE_MESSAGE                     "INFO "
   TRANS_CODE                              "TRNCD "
   EXTENDED_LINE_OUTPUT                    "EXTEND"
   NO_LOG_CONTROL                          "NLOGC "
   BELL                                     "BELL "
   OVERWRITE                               "OVERRIDE"
   BELL_HOMOGENEOUS                        "BELHOM"
   BELL_EXTEND                             "BELEXT"
   BELL_INFORMATIVE                        "BELINF"
   BELL_NO_LOG_CONTROL                     "BELNLC"
   HARDCOPY_NO_LOG_CONTROL                  "HCNLC "                                */

/* Possible values for EDIT_IN.EDIT_OPTIONS
   NO_CORRECTION                           "NOCORR"
   LOWER_CASE                              "LCASE "
   DELETE_DEVICE_HEADER                    "NOHDR "
   GET_FUNCTION_CODE                       "GETFC "
   CONFIDENTIAL_DATA                       "CFDATA"
   GET_ID_CARD                             "GETIC "
   EXTENDED_LINE_INPUT                     "EXTEND"
   LOWER_CASE_EXTEND                       "LOWEXT"
   CONF_DATA_LOW_CASE                      "CFDLW"
   FCT_CODE_LOW_CASE                       "GFCLW"
   FCT_CODE_CONF_DATA                      "GFCCFD"
   FCT_CODE_CONF_DATA_LCASE                "GFCCDL"
   FCT_CODE_EXTEND                         "GFCEXT"
   FCT_CODE_EXTEND_LCASE                   "GFCEXL"
   DELETE_DEVICE_HEADER_LCASE              "NOHDRL"                                */

/***** ISAM informations *****/
/* Possible values for ISAM_REQUEST          */
#define NO_KEY                             'N'
#define GET_KEY                             'K'
#define GET_KEY_LENGTH                       'L'
#define GET_KEY_POSITION                     'P'
#define GET_KEY_POSITION_LENGTH              'B'

/***** ASSIGNMENT informations *****/
/* ASSIGNMENT informations                  */
/* Possible values for ASSIGNMENT_REQUEST   */
#define NO_ASSIGNMENT_CODE                   'N'
#define GET_ASSIGNMENT_CODE                  'G'

/***** Call parameter list *****/
typedef struct {
    char    edit_mode;
    char    filler_1;
    char    edit_options[6];
    char    filler_2[8];
} edit_out_type;

```

```
typedef struct {
    char        edit_mode;
    char        filler_1;
    char        edit_options[6];
    char        filler_2[8];
} edit_in_type;

typedef struct {
    short       tiam_rc;
    char        filler_1[2];
    short       assignment;
    char        filler_2[2];
    short       key_position;
    short       key_length;
    char        filler_3[4];
} tiam_return_info_type;

typedef struct {
    edit_out_type    edit_out;
    edit_in_type     edit_in;
    unsigned         readlength;
    char             isam_request;
    char             filler_1[3];
    char             assignment_request;
    char             filler_2[3];
    short            copymem_id;
    char             filler_3[2];
    tiam_return_info_type tiam_return_info;
} tiam_control_info_type;
#endif
```

## 7.2 TIAM-C-Aufrufe

### 7.2.1 Daten lesen von SYSDDTA

Der Aufruf RDATA liest einen Datensatz von SYSDDTA. Die Datei, die als SYSDDTA erklärt ist, kann eine katalogisierte SAM- oder ISAM-Datei, ein Diskettengerät, ein Element einer PLAM-Bibliothek, eine SDF-P-Variable oder die prozeßführende Datensichtstation sein.

#### Format des Aufrufs:

```
extern RDATA(tiam-control-info_type *ptr1,
             struct input_area *ptr2,
             vtsucb_type *ptr3);
```

#### Programmbeispiel RDATA

```

/*****
* Beispiel für den R D A T A in C-Programmen.
* Sie werden zu einer Eingabe aufgefordert, die anschließend
* ausgegeben wird. Danach werden die TIAM-Returncodes angezeigt.*
*****/
#include <stdio.h>
#include <cinio.h>
#include <vtsucb.h>
#include <cctrc.h>
#include <string.h>

struct input_area {
    long    dtlngt;
    char    text[20];
} input_area;
tiam_control_info_type tiam_control_info;
vtsucb_type vtsucb1 = VTSUCB_DEFAULT;

extern RDATA(tiam_control_info_type *ptr1,
             struct input_area *ptr2,
             vtsucb_type *ptr3);

/**/
int main () {
    printf ("Geben Sie Ihren Text ein (max. 20 Zeichen)\n");
    strncpy (tiam_control_info.edit_in.edit_options, "VTCBU ",
            sizeof (tiam_control_info.edit_in.edit_options));
    tiam_control_info.readlength = sizeof(input_area);
    tiam_control_info.copymem_id = 1;
    RDATA(&tiam_control_info, &input_area, &vtsucb1);
    printf ("TIAM RC : %d\n",
            tiam_control_info.tiam_return_info.tiam_rc);
    printf ("VTSU MRC : %d\n",
            vtsucb1.vtsucb_std_hdr.std_vtsucb_rc.maincode);
    printf ("Länge des Eingabetextes : %ld\n", input_area.dtlngt);
    printf ("Eingabetext : %s\n", input_area.text);
    return(0);
}

```



## 7.2.2 Daten ausgeben nach SYSOUT

Der Aufruf WROUT gibt eine Nachricht nach SYSOUT aus. Die Datei, die als SYSOUT erklärt ist, kann eine katalogisierte SAM- oder ISAM-Datei oder die prozeßführende Datenstation sein.

### Format des Aufrufs:

```
extern WROUT(tiam_control_info_type *ptr1,
             out_area *ptr2);
```

### Programmbeispiel WROUT

```

/*****
* Beispiel für den W R O U T in C-Programmen.
* Es wird eine Meldung und anschließend der TIAM-Returncode
* ausgegeben.
*****/
#include <stdio.h>
#include <cinio.h>
#include <string.h>

typedef struct {
    long    dtlngt;
    char    pcc;
    char    text[22];
} out_area;
tiam_control_info_type tiam_control_info;
out_area output_area;

extern WROUT(tiam_control_info_type *ptr1,
             out_area *ptr2);
/**/
int main () {
    tiam_control_info.edit_out.edit_mode = LINE_MODE;
    tiam_control_info.copymem_id = 1;
    strncpy (tiam_control_info.edit_out.edit_options, "BELL ",
            sizeof (tiam_control_info.edit_out.edit_options));
    output_area.dtlngt = sizeof(output_area.dtlngt) +
                        sizeof(output_area.pcc) +
                        sizeof(output_area.text);
    strncpy (output_area.text, "Test des WROUT-Aufrufs", 22);

    WROUT(&tiam_control_info, &output_area);
    printf ("TIAM RC : %d\n",
            tiam_control_info.tiam_return_info.tiam_rc);
    return(0);
}

```

## 7.2.3 Kombinierte Aus- und Eingabe

Der Aufruf WRTRD gibt eine Nachricht an eine Datensichtstation aus und liest anschließend eine Nachricht von der Datensichtstation. Außer der zur Datensichtstation übertragenen Nachricht erscheint dabei kein weiteres Eingabeaufforderungszeichen.

### Format des Aufrufs:

```
extern WRTRD (tiam_control_info_type *,
             out_area *,
             in_area *);
```

### Programmbeispiel WRTRD

```
/* *****
 * Beispiel für den W R T R D in C-Programmen. *
 * ***** */
#include <stdio.h>
#include <cinio.h>
#include <string.h>

typedef struct {
    long    dtlngt;
    char    pcc;
    char    text[24];
} out_area;
typedef struct {
    long    dtlngt;
    char    text[24];
} in_area;
tiam_control_info_type tiam_control_info;
out_area output_area;
in_area input_area;

extern WRTRD (tiam_control_info_type *,
             out_area *,
             in_area *);

int main ()
{
    tiam_control_info.edit_in.edit_mode = LINE_MODE;
    tiam_control_info.edit_out.edit_mode = LINE_MODE;
    tiam_control_info.copymem_id = 1;

    output_area.dtlngt = sizeof(output_area.dtlngt) +
                        sizeof(output_area.pcc) +
                        sizeof(output_area.text);
    strncpy (tiam_control_info.edit_out.edit_options, "BELL ",
            sizeof (tiam_control_info.edit_out.edit_options));
    strncpy (output_area.text, "Geben Sie Ihren Text ein", 24);

    tiam_control_info.readlength = sizeof(input_area);
    strncpy (tiam_control_info.edit_in.edit_options, "LCASE ",
            sizeof (tiam_control_info.edit_in.edit_options));
```

---

```
WRTRD(&tiam_control_info, &output_area, &input_area);
printf ("TIAM RC : %d\n",
        tiam_control_info.tiam_return_info.tiam_rc);
printf ("Eingabetext : %s\n", input_area.text);
return(0);
}
```



---

## 8 POSIX-Schnittstelle

Ab TIAM V11.2A werden auch die POSIX Ein-/Ausgabeschnittstellen unterstützt. Dies gilt jedoch nur für NDS-Datensichtstationen (wie z.B. 9750, 9755, 9763...), nicht für 3270 oder über X29 angeschlossene Datensichtstationen. POSIX-Ein-/Ausgabeschnittstellen sind im wesentlichen die POSIX-Funktionen `read()` und `write()`.

Dieser Abschnitt soll die wichtigsten Eigenschaften von POSIX Ein-/Ausgabeschnittstellen aus der Sicht von TIAM beschreiben. Eine vollständige Beschreibung der POSIX-Schnittstellen finden Sie in den Handbüchern "POSIX Grundlagen" [16] und "POSIX-Schnittstellen" [17].

Besonderheiten bzw. Einschränkungen der BS2000-POSIX-Ein-/Ausgabeschnittstellen, sind ebenfalls beschrieben.

### 8.1 Parallele Ein-/Ausgabeoperationen unter BS2000 und POSIX

Ein BS2000-Dialogprozeß kann die TIAM-Ein-/Ausgabeanforderungen RDATA, WROUT und WRTRD und die POSIX-Ein-/Ausgabeanforderungen `read()` und `write()` in beliebiger Reihenfolge an die gleiche BS2000-Datenstation ausgeben. Dies gilt für eine von LOGON aufgerufene Task und alle dazugehörigen mit der POSIX-Funktion `fork()` erzeugten Tasks.

Die Synchronisierung der Ein-/Ausgaben von verschiedenen zu einer Datenstation gehörigen Tasks unterliegt der Verantwortung der Anwendung. Das System benutzt eine FIFO-Methode, um die Ein-/Ausgabeanforderungen von POSIX-Prozessen zu bedienen. Der erste POSIX-Prozeß, der eine Eingabe erkennt, erhält die erste Benutzereingabe. Die Entscheidung, welche Schnittstellen verwendet werden, die POSIX-Schnittstellen `write()` und `read()` oder die BS2000-Schnittstellen RDATA, WROUT und WRTRD, ist ebenfalls die Sache der Anwendung.

### 8.1.1 Einschränkungen

- Es gibt einige Einschränkungen bei der Benutzung von BS2000-Schnittstellen innerhalb von mit `fork()` erzeugten Tasks, wenn die Systemdateien nicht der Datenstation zugeordnet sind (siehe Beschreibung der SYSFILE-Umgebung für mit `fork()` erzeugte Tasks Seite 200).
- Wenn die LOGON-Task in den Systemmodus wechselt, werden Ein-/Ausgaben der zugehörigen mit `fork()` erzeugten Tasks zurückgestellt, wenn ein Programm für die LOGON-Task geladen bleibt. Dies ist zum Beispiel der Fall, wenn eine LOGON-Task mit der K2-Taste unterbrochen wird. Die Ein-/Ausgabe der mit `fork()` erzeugten Tasks werden erst weiterbearbeitet, wenn die LOGON-Task in den Programm-Modus zurückgekehrt ist. Eine POSIX-Shell läßt sich nicht mit K2 unterbrechen.
- Ein-/Ausgaben von mit `fork()` erzeugten Tasks werden zurückgewiesen, wenn die LOGON-Task das Ende des Programms erreicht hat. (Siehe Returncode X'38' bei RDATA, WROUT und WRTRD). Ein- /Ausgaben, die durch eine mit `fork()` erzeugte Task veranlaßt wurden, werden auch dann mit Returncode X'38' zurückgewiesen, wenn die betreffende Task nicht im Rahmen des laufenden Programms erzeugt wurde.

### 8.1.2 Behandlung von Sonderzeichen

Die Funktion der K2-Taste (zurück zum Systemmodus) kann nur zur Unterbrechung der LOGON-Task benutzt werden; nur diese Task kann von SYSCMD lesen.

Die POSIX-Steuerzeichen (INTR, QUIT) werden unterstützt und können über die P-Tastenfunktionen eingesetzt werden. Der Benutzer von POSIX-Anwendungen ist dafür verantwortlich, daß den P-Tasten die korrekten Werte zugewiesen werden. Eine Beschreibung der POSIX-Steuerzeichen und der entsprechenden Werte finden Sie im Handbuch "POSIX Kommandos" [14]).

### 8.1.3 BS2000-Ein-/Ausgabe und POSIX-Ein-/Ausgabe im Vergleich

BS2000 Ein-/Ausgabe	POSIX Ein-/Ausgabe
Nur eine Dialog-Task pro aktiver Datensichtstation	Mehrere POSIX-Prozesse (BS2000-Tasks) an einer Datensichtstation
Ein-/Ausgabe immer synchron <ul style="list-style-type: none"> <li>– Halbduplexbetrieb</li> <li>– immer blockiertes Lesen (Task, die die Ein-/Ausgabe realisiert, wartet immer auf Benutzereingabe)</li> </ul>	Asynchrone Ein-/Ausgabe <ul style="list-style-type: none"> <li>– Vollduplex-Betrieb</li> <li>– nicht-blockiertes Lesen möglich (Es wird ein Returncode geliefert, falls Daten nicht sofort verfügbar)</li> </ul>
Dialog-Task wird während der BCAM CONR Bearbeitung erstellt	POSIX-Prozeß wird durch <code>fork()</code> -Operation erstellt
Unterstützung der F- und K-Tasten.	Unterstützung spezieller Steuerzeichen (QUIT, INTR)
zeilenorientiertes Lesen (Line-Mode)	Lesen einer bestimmten Anzahl von Zeichen (siehe Definition der Schnittstelle <code>read()</code> )
Drei Funktionen : RDATA, WROUT und WRTRD	Zwei Funktionen : <code>read()</code> und <code>write()</code>
Eingabeaufforderungszeichen für RDATA (* für TU, / für TPR)	kein Eingabeaufforderungszeichen für READ

## 8.2 TCHNG und TSTAT in Verbindung mit POSIX

Datenstationseigenschaften, die innerhalb eines TU-Programms mit dem Makro TCHNG modifiziert wurden, bleiben bis zur Rückkehr in den Systemmodus gültig. Sie sind für alle Tasks gültig, die in diesem Programm aufgerufen wurden. Dabei spielt es keine Rolle, von welcher Task das Makro TCHNG aufgerufen wird, das heißt, ob die Task von LOGON oder mit der POSIX-Operation `fork()` erzeugt wird.

Wenn beispielsweise eine Task, die von der Operation `fork()` erzeugt wurde, eine Datenstationseigenschaft ändert, bleibt diese Änderung auch für die Task, die von LOGON erzeugt wird und für andere Tasks, die von `fork()` erzeugt werden, gültig. Die Änderung bleibt bis zur Rückkehr zum Systemmodus aktiv, selbst wenn die Task, die TCHNG aufgerufen hat, beendet wird.

## 8.3 SYSFILE-Umgebung für mit fork() erzeugte Tasks

Die SYSFILE-Umgebung ist zu Beginn für mit `fork()` erzeugte Tasks folgendermaßen definiert :

Systemdatei	Makro	Primärzuweisung im Dialogbetrieb	Primärzuweisung im Stapelbetrieb
SYSOUT	WROUT	Datenstation	Temporäre Spoolout-Datei S.OUT.
SYSLST	WRLST	Temporäre Spoolout-Datei S.LST...	Temporäre Spoolout-Datei S.LST...
SYSLSTn	WRLST	SYSLST	SYSLST
SYSDTA	RDATA	Datenstation	*DUMMY
SYSCMD	Priv.	*NONE	*NONE
	WRTRD	Datenstation	Abgelehnt (Returncode)
SYSOPT	WRTOT	Temporäre Spoolout-Datei S.OPT...	Temporäre Spoolout-Datei S.OPT...
SYSIPT	RDCRD	*DUMMY	*DUMMY

Die LOGON-Task erhält eine Standard-Dialogumgebung SYSFILE; jede mit `fork()` erzeugte Task hat ihre eigene SYSFILE-Umgebung. Während einer `fork()`-Operation erfolgt keine Übertragung der zugewiesenen Systemdatei vom Vaterprozeß.

Die Primärzuweisung der Systemdateien SYSDTA, SYSOUT und SYSLST kann selbst bei mit `fork()` erzeugten Tasks geändert werden. Beachten Sie, daß bei einer solchen Operation gegebenenfalls SDF aktiviert werden muß.

Nur die LOGON-Task darf von SYSCMD lesen. Bei mit `fork()` erzeugten Tasks wird die Ausführung eines Kommandos (z.B. das Makro CMD mit /CALL-PROCEDURE), das Zugriff auf SYSCMD erfordert, abgelehnt.

Im Gegensatz zu Ein-/Ausgabeoperationen unter BS2000 werden POSIX-Ein-/Ausgabeoperationen nicht in SYSOUT/SYSLST protokolliert.



# 9 Anhang

## 9.1 Tabelle des normierten Funktionstastencodes

Code		Bedeutung	8110	8151	8152	816x	9749 975x	9763	3270 <sup>3)</sup>
normal	mit löschen <sup>1)</sup>								
00	10	Datenübertragung (normal)	DÜ	DÜZ, DÜM	DÜZ, DÜM, DÜB	DÜ, DÜ1, DÜ2	DÜ, DÜ1, DÜ2	SEND	ENTER
01	11	Kurztelegramm 1		FT1	F1	K1	K1	K1	PA 1
02	12	" 2		FT2 <sup>2)</sup>	F2 <sup>2)</sup>	K2 <sup>2)</sup>	K2 <sup>2)</sup>	K2 <sup>2)</sup>	PA 2 <sup>2)</sup>
03	13	" 3		DVA	F3	K3	K3	K3	PA 3 PF6
04	14	" 4			F1+FZ	K4	K4	K4	PF7
05	15	" 5			F2+FZ <sup>2)</sup>	K5	K5	K5	PF8
06	16	" 6			F3+FZ	K6	K6	K6	PF9
07	17	" 7				K7	K7	K7	PF10
08	18	" 8				K8	K8	K8	PF11
09	19	" 9				K9	K9	K9	PF12
0A	1A	" 10				K10	K10	K10	PF13
0B	1B	" 11				K11	K11	K11	PF14
0C	1C	" 12				K12	K12	K12	PF15
0D	1D	" 13				K13	K13	K13	PF16
0E	1E	" 14				K14	K14	K14	PF17
20	30	Datenübertragung (markiert) 0			DÜZ, DÜM, DÜB +FZ		DÜZ, DÜM, DÜB, +FZ		
21	31	" 1				F1	F1	F1	PF 1
22	32	" 2				F2	F2	F2	PF 2
23	33	" 3				F3	F3	F3	PF 3
24	34	" 4				F4	F4	F4	PF 4
25	35	" 5				F5	F5	F5	PF 5

Code									
normal	mit löschen <sup>1)</sup>	Bedeutung	8110	8151	8152	816x	9749 975x	9763	3270 <sup>3)</sup>
26	36	" 6						F6	PF 18
27	37	" 7						F7	PF 19
28	38	" 8						F8	PF 20
29	39	" 9						F9	PF 21
2A	3A	" 10						F10	PF22
2B	3B	" 11						F11	PF23
2C	3C	" 12						F12	PF24
2D	3D	" 13						F13	
2E	3E	" 14						F14	
2F	3F	" 15			PÜ			F15	
30		Datenübertragung (Positionsdaten)							
40	50	Datenübertragung (speziell) 0					Bypass- Eingabe		
41	51	" 1					positive Rück- meldung		
42	52	" 2					negative Rück- meldung		
43	53	" 3							
45	55	Magn. Ausweisleser					x	x	x
46	56	Datenübertragung (markiert) 16						F16	
47	57	" 17						F17	
48	58	" 18						F18	
49	59	" 19						F19	
4A	5A	" 20						F20	
4B	5B	" 21						F21	
4C	5C	" 22						F22	
4D	5D	" 23						F23	
4E	5E	" 24						F24	
10		Löschen Speicher							CLEAR
60		Chipkartenterminal						x	
80		Ausweis steckt					x	x	x

1) Bildschirm wurde vor der Eingabe gelöscht

2) bei Verwendung der Zugriffsmethode TIAM reserviert für ESCAPE/BREAK Funktion

3) Standard-Werte siehe VTSU-Betriebsparameter COMPKEYS (siehe VTSU-Handbuch [1], Kapitel „Betriebsparameter“.

## 9.2 Besonderheiten von 3270-Datenstationen

1. Es gibt keinen Roll-up-Betrieb. Nach Erreichen der letzten Bildschirmzeile wird die Ausgabe in der ersten Zeile fortgesetzt.
2. Am Ende einer Ausgabe wird nicht der Bildschirmrest gelöscht, sondern nur der Zeilenrest und die gesamte folgende Zeile (das kann, falls die Ausgabe in der letzten Zeile endet, auch die erste Bildschirmzeile sein). An den Beginn der übernächsten Zeile wird das Anzeigesteuerzeichen NOR gesetzt.
3. Wirkungsweise der logischen Steuerzeichen:
  - Jedes logische Steuerzeichen benötigt den Platz eines Zeichens auf dem Bildschirm. Die Steuerzeichen NL und VPA benötigen zwei Plätze (einen am Ende der Zeile, in der sie angegeben werden und einen in der Zeile, in der die Ausgabe fortgesetzt werden soll). Mehrere Steuerzeichen hintereinander werden zu einem Zeichen auf dem Bildschirm zusammengezogen.
  - Durch jedes logische Steuerzeichen wird ein neues Feld eingerichtet, sodaß bei Verwendung der logischen Steuerzeichen EM1 bis EM4, DAR und NOR (durch diese wird bei TD-Datenstationen kein neues Feld eingerichtet) mit EXTEND=NO die Länge einer nach Modifikation wieder eingegebenen Ausgabe kürzer als bei TD-Datenstationen sein kann.
  - Das logische Steuerzeichen NUM läßt andere Angaben zu als bei TD-Datenstationen.
4. Wirkung der EDIT-Options:
 

HCOPY	Es wird immer der gesamte Bildschirm abgedruckt, sodaß es zu mehrmaligem Ausdruck von einzelnen Ausgaben und Eingaben kommen kann. Genauso werden auch Ausgaben ausgedruckt, für die zwar kein Abdruck verlangt ist, die aber zur Zeit des angeforderten Hardcopy-Drucks noch auf dem Bildschirm stehen.
EXTEND	NIL-Zeichen können in der Eingabe nur bedingt verwendet werden, da sie nicht mit zur DVA übertragen werden. VTSU-B ergänzt aber alle Felder, die verkürzt zurückkommen durch NIL-Zeichen, sodaß der Anwender die erwartete Länge erhält. Kommen allerdings einzelne Felder gar nicht zurück (z.B. wenn die Taste "ERASE INPUT" gedrückt wurde), so liefert VTSU-B einen Returncode.
GETFC	Die Abbildung der 3270-Funktionstastencodes können Sie der entsprechenden Tabelle auf Seite 201 entnehmen.

Weitere Einzelheiten für die Verwendung des VTSUCB entnehmen Sie dem VTSU-Handbuch (siehe [1]).



---

# Literatur

- [1] **VTSU**  
Virtual Terminal Support  
Benutzerhandbuch
- Zielgruppe*  
Anwender der Zugriffsmethoden DCAM, TIAM und UTM sowie System- und Netzverwalter.
- Inhalt*  
VTSU (Virtual Terminal Support) ist ein Softwareprodukt des Betriebssystems BS2000. Es realisiert eine logische Zeilendatenstation. Eine logische Datenstation erlaubt eine von den physikalischen Eigenschaften der Datenstation unabhängige Programmierung.
- [2] **BS2000/OSD-BC**  
Kommandos  
Benutzerhandbuch
- Zielgruppe*  
Das Handbuch wendet sich sowohl an den nichtprivilegierten BS2000/OSD-Anwender als auch an die Systembetreuung.
- Inhalt*  
Es enthält die BS2000/OSD-Kommandos ADD-... bis LOGOFF (Grundausbau und ausgewählte Produkte). Die Einleitung gibt Hinweise zur Kommandoeingabe. Die Kommandobeschreibung zeigt u.a. die Privilegien zur Nutzung der Kommandos.
- [3] **BS2000/OSD-BC**  
Makroaufrufe an den Ablaufteil  
Benutzerhandbuch
- Zielgruppe*  
Das Handbuch wendet sich an alle BS2000/OSD-Assembler-Programmierer.
- Inhalt*  
Das Handbuch enthält eine Zusammenstellung der Makroaufrufe an den Ablaufteil, die ausführliche Beschreibung jedes Makroaufrufs mit Hinweisen und Beispielen, einschließlich der Jobvariablen-Makros, sowie einen ausführlichen allgemeinen Lernteil.

- [4] **BS2000/OSD-BC**  
Systeminstallation  
Benutzerhandbuch

*Zielgruppe*

BS2000/OSD-Systemverwaltung

*Inhalt*

Das Handbuch beschreibt

- die Generierung der Hardware- und Software-Konfiguration mit UGEN
- die Installationsdienste
  - Plattenorganisation mit MPVS
  - Programmsystem SIR
  - Datenträgerinstallation mit SIR
  - Configuration Update (CONFUPD)
  - Dienstprogramm IOFCOPY.

- [5] **SDF V4.0A (BS2000/OSD)**  
Einführung in die Dialogschnittstelle SDF  
Benutzerhandbuch

*Zielgruppe*

BS2000/OSD-Anwender

*Inhalt*

Das Handbuch beschreibt die Dialog-Eingabe von Kommandos und Anweisungen im SDF-Format. Ein Schnelleinstieg mit leicht nachvollziehbaren Beispielen und weitere umfangreiche Beispiele erleichtern die Anwendung. SDF-Syntaxdateien werden erklärt.

–

- [6] **FHS (TRANSDATA)**  
Benutzerhandbuch

*Zielgruppe*

Programmierer

*Inhalt*

Programmschnittstellen von FHS für TIAM-, DCAM- und UTM-Anwendungen. Erstellen, Einsatz und Verwalten von Formaten.

- [7] **FHS (BS2000/OSD, TRANSDATA)**  
Dialogerweiterung für TIAM und SDF-P  
Benutzerhandbuch

*Zielgruppe*

Anwendungsentwickler

*Inhalt*

Das Handbuch beschreibt die Programmschnittstelle für den Einsatz des FHS-Dilaogmanagers in TIAM- und SDF-P-Anwendungen.

**[8] Netzzugang für Datenstationen (TRANSDATA)**

Benutzerhandbuch

*Zielgruppe*

- Jeder Anwender, der über eine Datenstation Leistungen eines Rechners in Anspruch nehmen will
- Netzadministrator

*Inhalt*

- Generieren des Netzzugangs
- Ablauf des Netzzugangs für alle zulässigen Varianten von Netzanschlüssen
- Netzkommandos und -meldungen

**[9] Datensichtstationen 9749, 9750, 9752 (TRANSDATA)****Schnittstelle für Programmierer**

Benutzerhandbuch

*Zielgruppe*

Programmierer (Anwendungsprogrammierer), der Ausgaben auf Datensichtstationen programmieren will und Eingaben von Datensichtstationen interpretieren und auswerten möchte

*Inhalt*

- Hinweise zur Konfiguration des TRANSDATA-Netzes, Beschreibung der
- Beschreibung der Funktionen der Datensichtstationen
- Hinweise zur physischen und logischen Programmierung dieser Funktionen
- Beschreibung des Nachrichtenaufbaus
- Datenaustausch mit Druckern.
- Beschreibung des Software-Produkts PLUS.

**[10] JV (BS2000/OSD)**

Jobvariablen

Benutzerhandbuch

*Zielgruppe*

Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.

*Inhalt*

Es beschreibt die Verwaltung und die verschiedenen Einsatzmöglichkeiten von Jobvariablen. Die Kommandobeschreibungen sind getrennt nach den Funktionsbereichen der JVs zu erhalten. Die Makroaufrufe sind in einem eigenen Kapitel beschrieben.

- [11] **BS2000/OSD-BC**  
Dienstprogramme  
Benutzerhandbuch
- Zielgruppe*  
Das Handbuch wendet sich sowohl an den nichtprivilegierten Anwender als auch an die Systembetreuung.
- Inhalt*  
Es beschreibt die Dienstprogramme CALENDAR V11.2A, CONDMPPD V11.2A, DPAGE V11.2A, INIT V11.2A, JMU V11.2A, LMSCONV V1.0B, MSGMAKER V1.1B PAMCONV V11.0A, PDPOOLS V11.2A, PVSREN V1.1A, RFUDP V11.0A, SODA V11.2A, SPCCNTRL V11.2A, TPCOMP2 V11.2A, VOLIN V11.2A.
- [12] **Datensichtstationen (TRANSDATA)**  
Funktionelle Beschreibung
- Zielgruppe*  
BS2000-Programmierer
- Inhalt*  
Funktionelle Eigenschaften und Anschlußmöglichkeiten von Datensichtstationen sowie den Betrieb mit angeschlossenen Druckern.
- [13] **Datensichtstationen (TRANSDATA)**  
Code-Tabellen
- Zielgruppe*  
BS2000-Programmierer
- Inhalt*  
Beschreibung aller Codes, die zur Programmierung der Datensichtstationen und zum Datenaustausch mit Druckern erforderlich sind.
- [14] **XHCS (BS2000/OSD)**  
8-bit-Code-Verarbeitung im BS2000/OSD  
Benutzerhandbuch
- Zielgruppe*  
Anwender der Zugriffsmethoden DCAM, TIAM und UTM sowie Systemverwalter, Anwender, die von EHCS auf XHCS umstellen.
- Inhalt*  
XHCS (Extended Host Code Support) ist ein Softwareprodukt des BS2000/OSD. Es ermöglicht Ihnen erweiterte Zeichensätze bei 8-bit-Datenstationen zu nutzen. XHCS ist die zentrale Informationsquelle über die codierten Zeichensätze im BS2000/SD. XHCS löst EHCS ab.



- [15] **POSIX V1.0A (BS2000/OSD)**  
Kommandos  
Benutzerhandbuch  
*Zielgruppe*  
Das Handbuch wendet sich an alle Benutzer der POSIX-Shell.  
*Inhalt*  
Dieses Handbuch ist ein Nachschlagewerk. Es beschreibt das Arbeiten mit der POSIX-Shell sowie die Kommandos der POSIX-Shell in alphabetischer Reihenfolge.
- [16] **POSIX (BS2000/OSD)**  
Grundlagen für Anwender und Systemverwalter  
Benutzerhandbuch  
*Zielgruppe*  
BS2000-Systemverwalter, POSIX-Verwalter, BS2000-Benutzer, Benutzer von UNIX-/SINIX-Workstations.  
*Inhalt*  
Einführung und Arbeiten mit POSIX; BS2000-Softwareprodukte im Umfeld von POSIX; POSIX-Subsystem installieren, steuern und beenden; POSIX-Benutzer durch das BS2000 verwalten.
- [17] **C-Bibliotheksfunktionen für POSIX-Anwendungen (BS2000/OSD)**  
Referenzhandbuch

## Bestellen von Handbüchern

Die aufgeführten Handbücher finden Sie mit ihren Bestellnummern im *Druckschriftenverzeichnis* der Siemens Nixdorf Informationssysteme AG. Neu erschienene Titel finden Sie in den *Druckschriften-Neuerscheinungen*.

Beide Veröffentlichungen erhalten Sie regelmäßig, wenn Sie in den entsprechenden Verteiler aufgenommen sind. Wenden Sie sich bitte hierfür an Ihre zuständige Geschäftsstelle. Dort können Sie auch die Handbücher bestellen.



# Stichwörter

## \$DIALOG

logische Verbindung 1

3270-Datenstation 203

## A

Aliasname 9

Anweisung

Syntaxdarstellung 7

Aufbereitung 111

Aufbereitungsfunktion 58, 91, 139, 142

Auftrag

Ablaufprotokoll 40

abnormal beenden siehe EXIT-JOB 11

beenden siehe EXIT-JOB 11

beenden siehe LOGOFF 14

starten 41

überwachen (Jobvariable) 38

## B

Bildschirmüberlauf steuern 28

## C

cctrc.h 188

cinfo.h 189

Coded-Character-Set 31

C-Sprache 187

CUPAB 50

### D

- Daten
  - ausgeben nach SYSOUT 88, 154, 177, 184, 193
  - lesen von SYSDTA 56, 147, 176, 183, 192
- Datenstationseigenschaften
  - abfragen 78
  - ändern 74
- Datenstationseigenschaften ändern siehe MODIFY-TERMINAL-OPTIONS 26
- Datenstationseigenschaften anzeigen siehe SHOW-TERMINAL-OPTIONS 44
- Datenstrukturen
  - FORTRAN 172
- Dialogauftrag einleiten siehe SET-LOGON-PARAMETERS 35

### E

- Ein-/Ausgabe-Synchronisation 197
- Empfang von Nachrichten steuern siehe MODIFY-MSG-OPTIONS 24
- EXIT-JOB
  - (Kommando) 11
- EXTEND 96, 116, 155, 162

### F

- FHS-MAIN-PAR 155, 162
- FHS-MODE 139
- FORCTRC 172
- FORINFO 173
- Format-Modus 92, 113
- Füllzeichen 154, 160
- Funktionstastencode 62, 119, 142, 202

### I

- Informationszeile 95, 140
- ISAM-Schlüssel 60, 144, 145

### J

- Jobvariable
  - Auftrag überwachen 38

**K**

Kennwort angeben

Benutzerkennung 37

Jobvariable 38

Kombinierte Aus- und Eingabe 108, 160, 178, 185, 194

Kommando

EXIT-JOB 11

LOGOFF 14

MODIFY-JOB-OPTIONS 17

MODIFY-MSG-OPTIONS 24

MODIFY-TERMINAL-OPTIONS 26

SET-LOGON-PARAMETERS 35

SHOW-TERMINAL-OPTIONS 44

Syntaxdarstellung 7

Kurzname 9

**L**

LINE-MODE-CONTROL-CHARACTER 180, 188

logische Verbindung

\$DIALOG 1

LOGOFF

(Kommando) 14

**M**

Metasyntax SDF 8

Metazeichen 47

MODIFY-JOB-OPTIONS

(Kommando) 17

MODIFY-MSG-OPTIONS

(Kommando) 24

MODIFY-TERMINAL-OPTIONS

(Kommando) 26

**N**

Nachricht 88, 108, 154, 160, 177, 178, 184, 185, 193, 194

**O**

Operandentabelle

für Dialogkommunikation definieren 50

### P

- PL1CTRC 180
- PL1INFO 181
- POSIX 200
  - Ein-/Ausgabeanforderungen 197
  - Gültigkeit von Datenstationseigenschaften 199
- POSIX Ein-/Ausgabeschnittstellen 197
- POSIX-Steuerzeichen 198
- Priorität definieren
  - Stapelauftrag 40
- Programmmodus 1
- Protokoll
  - SYSLST 40
- Protokollierung und Meldungsausgabe steuern siehe MODIFY-JOB-OPTIONS 17

### R

- RDATA 56, 146
- RS
  - Rücksprungschalter 77
- Rücksprungschalter
  - RS 77

### S

- Satzlängenfeld 58, 90, 112
- Schutz
  - vertraulicher Daten 63
- SDF
  - Syntaxdarstellung 7
- SDF-Kommandos 10
- SET-LOGON-PARAMETERS
  - (Kommando) 35
- SHOW-TERMINAL-OPTIONS
  - (Kommando) 44
- Stapelauftrag
  - Priorität definieren 40
  - starten 41
  - wiederholen 42
- Stapelauftrag einleiten siehe SET-LOGON-PARAMETERS 35
- Syntaxdarstellung SDF 7
- SYSFILE-Umgebung 200
  - POSIX 200
- SYSLST (Systemdatei)
  - ausgeben (Band) 15
  - ausgeben (Drucker) 15

Protokoll 40  
SYSOPT (Systemdatei)  
    ausgeben (Band) 15  
SYSOUT (Systemdatei)  
    ausgeben (Band) 15  
    ausgeben (Drucker) 15  
Systemdatei  
    ausgeben (Band) 15  
    löschen 15  
Systemmodus 1

**T**  
TCHNG 74  
TIAM-C-Aufrufe 188, 192  
TIAM-COBOL-Aufrufe 146  
TIAM-COBOL-Schnittstelle 135  
TIAMCTRC  
    COBOL 136  
TIAM-FORTRAN-Aufrufe 176  
TIAMINFO  
    COBOL 137  
TIAM-Kommando 7  
TIAM-Makroaufrufe 47  
TIAM-PL/I-Aufrufe 180, 183  
TIMER 63, 120  
TSTAT 78

**U**  
Überlaufkontrolle 75

**V**  
VTSUCB  
    FORTRAN 172  
VTSUCBA 63, 98, 120  
VTSUCBC  
    COBOL 136

**W**  
WROUT 88, 146, 154, 177, 184, 193  
WRTRD 108, 146, 160, 178, 185, 194

**Z**  
Zugriffsmethode 1





# Inhalt

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
1.1	Zielgruppen des TIAM-Handbuchs .....	3
1.2	Konzept des TIAM-Handbuchs .....	3
1.3	Änderungen gegenüber Vorgängerausgabe .....	4
1.4	Readme-Datei .....	5
<b>2</b>	<b>TIAM-Kommandos</b> .....	<b>7</b>
2.1	SDF-Syntaxdarstellung .....	7
2.2	Kurzbeschreibung der Kommandos .....	10
2.3	Auftrag beenden: EXIT-JOB .....	11
2.4	Auftrag beenden: LOGOFF .....	14
2.5	Protokollierung des Auftrags festlegen: MODIFY-JOB-OPTIONS .....	17
2.6	Empfang von Operatormeldungen steuern: MODIFY-MSG-OPTIONS .....	24
2.7	Datenstationseigenschaften ändern: MODIFY-TERMINAL-OPTIONS .....	26
2.8	Dialogauftrag starten: SET-LOGON-PARAMETERS .....	35
2.9	Datenstationseigenschaften anzeigen: SHOW-TERMINAL-OPTIONS .....	44
<b>3</b>	<b>TIAM-Makroaufrufe</b> .....	<b>47</b>
3.1	Verwendete Metasprache .....	47
3.2	Operandentabelle für Dialogkommunikation definieren: CUPAB .....	50
3.3	Daten lesen von SYSDTA: RDATA .....	56
3.4	Datenstationseigenschaften ändern: TCHNG .....	74
3.5	Datenstationseigenschaften abfragen: TSTAT .....	78
3.6	Daten ausgeben nach SYSOUT: WROUT .....	88
3.7	Kombinierte Aus- und Eingabe: WRTRD .....	108
<b>4</b>	<b>TIAM-COBOL-Schnittstelle</b> .....	<b>135</b>
4.1	Datenstrukturen für die TIAM-COBOL-Aufrufe .....	136
	Datenstruktur TIAM-CONTROL-INFO .....	137
4.2	TIAM-COBOL-Aufrufe .....	146
4.2.1	Daten lesen von SYSDTA: CALL "RDATA" .....	147

4.2.2	Daten ausgeben nach SYSOUT: CALL "WROUT" .....	154
4.2.3	Kombinierte Aus- und Eingabe: CALL "WRTRD" .....	160
<b>5</b>	<b>TIAM-FORTRAN-Schnittstelle .....</b>	<b>171</b>
5.1	Datenstrukturen für die TIAM-FORTRAN-Aufrufe .....	172
	Datenstruktur TIAM-CONTROL-INFO .....	173
5.2	TIAM-FORTRAN-Aufrufe .....	176
5.2.1	Daten lesen von SYSDTA: CALL RDATA .....	176
5.2.2	Daten ausgeben nach SYSOUT: CALL WROUT .....	177
5.2.3	Kombinierte Aus- und Eingabe: CALL WRTRD .....	178
<b>6</b>	<b>TIAM-PL/I-Schnittstelle .....</b>	<b>179</b>
6.1	Datenstrukturen für die TIAM-PL/I-Aufrufe .....	180
	Datenstruktur TIAM-CONTROL-INFO .....	181
6.2	TIAM-PL/I-Aufrufe .....	183
6.2.1	Daten lesen von SYSDTA: CALL RDATA .....	183
6.2.2	Daten ausgeben nach SYSOUT: CALL WROUT .....	184
6.2.3	Kombinierte Aus- und Eingabe: CALL WRTRD .....	185
<b>7</b>	<b>TIAM-C-Schnittstelle .....</b>	<b>187</b>
7.1	Datenstrukturen für die TIAM-C-Aufrufe .....	188
	Datenstruktur TIAM-CONTROL-INFO .....	189
7.2	TIAM-C-Aufrufe .....	192
7.2.1	Daten lesen von SYSDTA .....	192
7.2.2	Daten ausgeben nach SYSOUT .....	193
7.2.3	Kombinierte Aus- und Eingabe .....	194
<b>8</b>	<b>POSIX-Schnittstelle .....</b>	<b>197</b>
8.1	Parallele Ein-/Ausgabeoperationen unter BS2000 und POSIX .....	197
8.1.1	Einschränkungen .....	198
8.1.2	Behandlung von Sonderzeichen .....	198
8.1.3	BS2000-Ein-/Ausgabe und POSIX-Ein-/Ausgabe im Vergleich .....	199
8.2	TCHNG und TSTAT in Verbindung mit POSIX .....	199
8.3	SYSFILE-Umgebung für mit fork() erzeugte Tasks .....	200
<b>9</b>	<b>Anhang .....</b>	<b>201</b>
9.1	Tabelle des normierten Funktionstastencodes .....	201
9.2	Besonderheiten von 3270-Datenstationen .....	203
	<b>Literatur .....</b>	<b>205</b>
	<b>Stichwörter .....</b>	<b>211</b>

---

# TIAM V11.2 (BS2000/OSD, TRANSDATA)

## Terminal Interactive Access Method

### *Zielgruppe*

Das Handbuch richtet sich an BS2000-Anwender (nicht privilegiert) und Programmierer, die Kommunikationsanwendungen mit Hilfe der TIAM-Schnittstellen programmieren.

### *Inhalt*

Das Handbuch beschreibt die Arbeit mit der Terminal Interactive Access Method (TIAM). Es sind die TIAM-Kommandos und -Makroaufrufe sowie die TIAM-COBOL-, FORTRAN-, PL/I- und C-Schnittstelle beschrieben. Außerdem ist die POSIX-Unterstützung durch TIAM beschrieben.

**Ausgabe: Juli 1995**

**Datei: TIAM.PDF**

BS2000 ist ein eingetragenes Warenzeichen der Siemens Nixdorf Informationssysteme AG  
Copyright © Siemens Nixdorf Informationssysteme AG, 1995.

Alle Rechte vorbehalten.  
Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.



## Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

The Internet pages of Fujitsu Technology Solutions are available at [http://ts.fujitsu.com/...](http://ts.fujitsu.com/)

and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

## Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter [http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009