

JV V15.0

Jobvariablen

Stand der Beschreibung

BS2000/OSD-BC V8.0

HIPLEX MSCF V6.0

SDF V4.7

SECOS V5.3

Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Copyright © Fujitsu Technology Solutions GmbH 2010.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Inhalt

1	Einleitung	7
1.1	Zielsetzung und Zielgruppen des Handbuchs	8
1.2	Konzept des Handbuchs	9
1.3	Änderungen gegenüber dem Vorgänger-Handbuch	11
1.4	Darstellungsmittel	12
2	Verwaltung und Einsatz von Jobvariablen	13
2.1	Speichern von Jobvariablen	15
2.1.1	Permanente Benutzer-Jobvariablen	15
2.1.2	Temporäre Benutzer-Jobvariablen	15
2.1.3	Sonder-Jobvariablen	15
2.2	Benennen von Benutzer-Jobvariablen	16
2.2.1	Permanente Benutzer-Jobvariablen	16
2.2.2	Wildcard-Syntax in Pfadnamen	18
2.2.3	Temporäre Benutzer-Jobvariablen	19
2.2.4	Kettungsnamen für Benutzer-Jobvariablen	20
2.3	Zugriffsverwaltung für Jobvariablen	22
2.3.1	Eigentümerschaft von Jobvariablen	23
2.3.2	Schutzmechanismen zur Zugriffssteuerung	24
2.3.3	Default-Protection (Benutzerdefinierte Standardwerte)	36
2.3.4	Schutz bei der Auftragsüberwachung und -steuerung	40
2.4	Zugriffsverlust auf eine Jobvariable	41
2.5	Die Werte der Benutzer-Jobvariablen	42
2.6	Sicherung von Benutzer-Jobvariablen	42
2.7	Sonder-Jobvariablen	44
	Exit 033 für Sonder-Jobvariablen	51

2.8	Auftrags-/Programmüberwachung mit Jobvariablen	53
2.8.1	Werte für überwachende Jobvariablen	54
2.8.1.1	Auftragsüberwachung	55
2.8.1.2	Überwachung von IMPORT-/EXPORT-PUBSET	58
2.8.1.3	Programmüberwachung	59
2.8.2	Identifizierung von Aufträgen mit überwachenden Jobvariablen	61
2.8.3	Schutz von überwachenden Jobvariablen	62
2.8.4	Kettungsnamen überwachender Jobvariablen	64
2.9	Bedingungsabhängige Auftragssteuerung	65
2.9.1	Bedingungen und Ereignisse	65
2.9.2	Synchronisierung von Ereignissen	66
2.9.3	Kommandos/Makros der Bedingungsabhängigen Auftragssteuerung	66
2.10	Eingabe aus Jobvariablen	70
3	Kommandos	75
<hr/>		
3.1	Kommandos zur Verwaltung von Jobvariablen	76
	Kommandoübersicht	76
3.2	Kommandos zur Auftragsüberwachung	77
3.2.1	Kommandos zur Vereinbarung einer Auftragsüberwachung	77
3.2.2	Kommandos der Auftragsüberwachung	80
3.3	Kommandos zur Programmüberwachung	82
	Kommandoübersicht	82
3.4	Kommandos zur bedingungsabhängigen Auftragssteuerung	83
3.4.1	Kommandoübersicht	83
3.4.2	Bedingte Ausdrücke	83
4	Makroaufrufe	87
<hr/>		
4.1	Makroübersicht	87
4.2	Allgemeine Makro-Operanden	88
4.2.1	Der Operand MF	88
4.2.2	Der Operand PREFIX	89
4.2.3	Der Operand PARAM	89
4.2.4	Der Operand MACID	89
4.3	Erläuterung der verwendeten Metazeichen	90

4.4	Beschreibung der Makros	92
	CATJV	
	Jobvariable katalogisieren	92
	COPJV	
	Jobvariable kopieren	104
	CSWJV	
	Jobvariable prüfen und setzen	109
	DCLJV	
	Jobvariablen-Kettungsnamen definieren	114
	DONEVT	
	Bedingung eines Jobvariablen-Ereignisses löschen	118
	ERAJV	
	Jobvariable löschen	123
	GETJV	
	Wert ausgeben	131
	JVSEL	
	JV-Auswahl des STAJV-Makros auf bestimmte Merkmale beschränken	136
	LNKJV	
	Jobvariablen JV-LINK-Einträge ausgeben	151
	ONEVT	
	Bedingung für ein Jobvariablen-Ereignis setzen	156
	RELJV	
	JV-LINK-Eintrag löschen	163
	SETJV	
	Jobvariable setzen	166
	STAJV	
	Merkmale von Jobvariablen ausgeben	172
	TERM	
	Programm und Prozedurabschnitt beenden	182
	TIMJV	
	Auftragsüberwachende JV modifizieren	186
5	Beispiele	191
6	Meldungen	227
6.1	Kommandoebene	227
6.2	Makroebene	229

7	Der privilegierte Benutzer	235
7.1	Installation	235
7.2	Privilegierte Zugriffsrechte	237
7.3	Systemdatei SYSCAT.JV	237
7.4	Überwachung bei der Pubset-Verwaltung	238
7.5	Jobvariablen als Objekt der Systemüberwachung	241
	Abkürzungen	243
	Literatur	245
	Stichwörter	247

1 Einleitung

Jobvariablen (JV's) sind Speicherbereiche zum Austausch von Informationen zwischen Aufträgen untereinander sowie zwischen Betriebssystem und Aufträgen. Sie haben einen Namen und einen Inhalt (Wert). Dieser Inhalt kann zur Steuerung von Aufträgen und Programmen genutzt werden. So kann z.B. veranlasst werden, dass ein Auftrag B erst dann gestartet wird, wenn Auftrag A die vorher vereinbarte Jobvariable V1 auf den Wert „B STARTEN“ gesetzt hat. Die zur Behandlung von Jobvariablen notwendigen Funktionen werden auf Kommando- oder Makroebene bzw. als COBOL-Anweisungen zur Verfügung gestellt.

Der Benutzer kann Jobvariablen

- erzeugen
- verändern
- abfragen
- löschen

oder das Betriebssystem anweisen, bei Änderungen des Zustands eines Auftrags oder Programms eine überwachende Jobvariable entsprechend zu setzen. Der Benutzer kann, statt durch Angabe der TSN, mit einer Jobvariablen den überwachten Auftrag in Kommandos ansprechen. Jobvariablen geben dem Benutzer somit ein Werkzeug zur flexiblen Auftragssteuerung in die Hand. Er kann mit einfachen Mitteln Abhängigkeiten zwischen Aufträgen definieren. Die bedingungsabhängige Auftragssteuerung ermöglicht ihm, die Durchführung bestimmter Aktionen (z.B. Starten von Aufträgen) von der Wertänderung gewisser, vorher bestimmter Jobvariablen abhängig zu machen. Dies geschieht durch Kommandos an das Betriebssystem, die nur - oder erst - dann ausgeführt werden, wenn das vom Benutzer definierte Ereignis eintritt (Jobvariable nimmt bestimmten Wert an). Wie Benutzer- oder Auftragschalter können diese Ereignisse für bedingte Sprünge innerhalb von Kommandofolgen und für synchrones Warten verwendet werden. Jobvariablen besitzen im Gegensatz zu Schaltern (Zustand ON oder OFF) eine Vielzahl von Wertausprägungen, deren Abfrage außerdem in bedingten Ausdrücken kombiniert werden kann.

Zusätzlich ist ein asynchrones Warten möglich (eine Kommandofolge wird erst bei Eintritt des Ereignisses ausgeführt und unterbricht dann erst die Verarbeitung).

Die im Folgenden beschriebenen Funktionen stehen sowohl in einem Einzelsystem als auch in einem BS2000-Rechnerverbund zur Verfügung.

1.1 Zielsetzung und Zielgruppen des Handbuchs

Das Handbuch beschreibt die Anwendungsmöglichkeiten des Produkts JV und enthält für den privilegierten und nicht-privilegierten Benutzer Hinweise zu Administration und Anwendung von Jobvariablen.

Die beschriebenen Jobvariablen-Funktionen sind im Grundausbau des BS2000 nicht enthalten. Für diese Funktionen wird ein Überblick über die Kommando-Schnittstelle gegeben. Die Assembler-Schnittstelle (Makroaufrufe) wird für diese Funktionen vollständig beschrieben. Über BS2000-Kommandos, die eine JV-Schnittstelle besitzen (MONJV-Funktion) wird ein tabellarischer Überblick gegeben.

Es werden Kenntnisse des BS2000, insbesondere der Kommandos, vorausgesetzt. Diese werden in den Handbüchern „Kommandos“ [1] und „Dialogschnittstelle SDF“ [2] vermittelt. Für den privilegierten Benutzer sind Kenntnisse zur Systembetreuung unerlässlich (Handbuch „Einführung in die Systembetreuung“ [3]).

Für den programmierenden Benutzer werden Kenntnisse in Assembler und der BS2000-Makroaufrufe vorausgesetzt. Soweit andere Software-Produkte Jobvariablen-Funktionen nutzen, ist die Beschreibung in den entsprechenden Produkt-Handbüchern zu finden.

1.2 Konzept des Handbuchs

Das Handbuch gliedert sich in sieben Kapitel und einen Anhang mit folgendem Inhalt:

Kapitel 1 „Einleitung“

enthält eine Kurzbeschreibung des Produkts JV, Zielsetzung, Zielgruppen und Aufbau des Handbuchs, Hinweise zur Readme-Datei und die Änderungen gegenüber dem Vorgängerhandbuch.

Kapitel 2 „Verwaltung und Einsatz von Jobvariablen“

informiert über die verschiedenen Arten von Jobvariablen und ihre Benennung, über den Zugriffsschutz, die Werte von Benutzer- und von Sonder-Jobvariablen, die Verwendung von Jobvariablen bei Auftrags-/Programmüberwachung und bei der bedingungsabhängigen Auftragssteuerung, sowie über die Eingabe aus Jobvariablen.

Kapitel 3 „Kommandos“

enthält Kommando-Übersichten zur Verwaltung von Jobvariablen, Auftragsüberwachung, Programmüberwachung und zur bedingungsabhängigen Auftragssteuerung.

Kapitel 4 „Makroaufrufe“

enthält eine Übersicht der Jobvariablen-Makroaufrufe, einen allgemeinen Abschnitt zu Makro-Operanden und Metasyntax, sowie alphabetisch sortiert die Beschreibungen aller Jobvariablen-Makroaufrufe.

Kapitel 5 „Beispiele“

enthält Beispiele mit Kommandos und Makros zur Verwaltung von Jobvariablen, Auftrags-/Programmüberwachung und zur bedingungsabhängigen Auftragssteuerung.

Kapitel 6 „Meldungen“

enthält Hinweise zu Systemmeldungen sowie eine Erläuterung aller Returncodes auf Programmebene, für die keine entsprechende Systemmeldung existiert.

Kapitel 7 „Der privilegierte Benutzer“

informiert den privilegierten Benutzer über die Installation des Produkts JV, über die Zugriffsrechte auf Jobvariablen des Systems, über die Möglichkeiten der Auftragsüberwachung bei der Pubset-Verwaltung und über Jobvariablen als Objekt der Systemüberwachung mit der Funktionseinheit SAT beim Einsatz von SECOS.

Am Ende des Handbuchs befinden sich Abkürzungs- und Tabellenverzeichnis, sowie Literatur- und Stichwortverzeichnis.

Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift ist im Literaturverzeichnis aufgeführt.

Readme-Datei

Funktionelle Änderungen der aktuellen Produktversion und Nachträge zu diesem Handbuch entnehmen Sie bitte ggf. der produktspezifischen Readme-Datei.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <http://manuals.ts.fujitsu.com> zur Verfügung. Alternativ finden Sie Readme-Dateien auch auf der Softbook-DVD.

Informationen unter BS2000/OSD

Wenn für eine Produktversion eine Readme-Datei existiert, finden Sie im BS2000-System die folgende Datei:

```
SYSRME.<product>.<version>.<lang>
```

Diese Datei enthält eine kurze Information zur Readme-Datei in deutscher oder englischer Sprache (<lang>=D/E). Die Information können Sie am Bildschirm mit dem Kommando `/SHOW-FILE` oder mit einem Editor ansehen.

Das Kommando `/SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product>` zeigt, unter welcher Benutzerkennung die Dateien des Produkts abgelegt sind.

Ergänzende Produkt-Informationen

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter <http://manuals.ts.fujitsu.com>.

1.3 Änderungen gegenüber dem Vorgänger-Handbuch

Das Handbuch JV V15.0 enthält gegenüber dem Vorgängerhandbuch JV V14.0A folgende Änderungen:

Änderung der Handbuchstruktur

Die Beschreibung der Jobvariablen-Kommandos sowie die allgemeinen Abschnitte „SDF-Metasyntax“ und „Kommando-Returncodes“ sind nicht mehr in diesem Handbuch enthalten. Eine vollständige Beschreibung der Kommandos sowie der allgemeinen Abschnitte „SDF-Metasyntax“ und „Kommando-Returncodes“ finden Sie im Handbuch „Kommandos“ [1].

Inhalt einer Jobvariablen kopieren

Das Kommando COPY-JV bzw. der Makro COPJV kopiert den Inhalt einer Jobvariablen (Sende-JV) in eine andere Jobvariable (Empfangs-JV). Wahlweise können neben dem Inhalt auch die Schutzmerkmale der Send-JV mit Ausnahme eines bestehenden MONJV- bzw. CJC-Schutzes kopiert werden. Empfangs- und Send-JV können sowohl permanente als auch temporäre Jobvariablen sein.

Fehlerbehandlung beim Einrichten einer Jobvariablen

Wenn beim Einrichten einer Jobvariablen mit dem Kommando CREATE-JV bzw. mit dem Makro CATJV und STATE=NEW die Jobvariable bereits existierte, wurde der Aufruf mit Fehler abgewiesen. Der neue Operand SUPPRESS-ERRORS bzw. NOSTEP steuert die Fehlerbehandlung. Optional kann der Fehler auch unterdrückt werden, d.h. der Aufruf wird nicht ausgeführt und es wird kein Fehler zurückgemeldet.

Kataloginformationen

Das Kommando SHOW-JV-ATTRIBUTES bzw. der Makro STAJV gibt angeforderten Kataloginformationen in alphabetischer Reihenfolge aus. Der neue Operanden OUTPUT-OPTIONS(SORT-LIST=) bzw. SORT steuert die Sortierreihenfolge. Optional können die Einträge jetzt auch in der Reihenfolge, in der sie im Katalog stehen, ausgegeben werden.

Datensicherung

Bei der Pubset-Sicherung auf Snapsets werden Dateien und Jobvariablen logisch gesichert. Der Benutzer kann sich mit Kommandos bzw. Makroaufrufen über Jobvariablen, die auf einem Snapset gesichert sind, informieren und diese bei Bedarf aus der Sicherung rekonstruieren. Siehe [Abschnitt „Sicherung von Benutzer-Jobvariablen“ auf Seite 42](#).

Neue Sonderjobvariablen

Die Sonderjobvariable `$$SYSJV.TASK-CPU-USED-LONG` gibt die verbrauchte CPU-Zeit für sogenannte „Langläufer“ aus.

Die Sonderjobvariable `$$SYSJV.REMAINING-BS2000-RUNTIME` zeigt auf einem SQ-Server die Restlaufzeit, die dem BS2000-System zum Herunterfahren zur Verfügung steht, bevor die reale bzw. virtuelle Maschine angehalten wird.

1.4 Darstellungsmittel

In diesem Handbuch werden folgende Darstellungsmittel verwendet:



Dieses Zeichen kennzeichnet Hinweise auf wichtige Informationen



Dieses Zeichen kennzeichnet einen Warnhinweis, der auf die Möglichkeit des Datenverlustes oder anderer ernsthafter Schäden an Daten hinweist.



Literaturhinweise werden im Text in Kurztiteln angegeben. Der vollständige Titel jeder Druckschrift, auf die durch eine Nummer verwiesen wird, ist im Literaturverzeichnis hinter der entsprechenden Nummer aufgeführt.

Eingabe

In Anwendungsbeispielen sind Eingaben an das System und Ausgaben des Systems in *Schreibmaschinenschrift* dargestellt.

2 Verwaltung und Einsatz von Jobvariablen

Dieses Kapitel beschreibt:

- welche Arten von Jobvariablen existieren
- wie Jobvariablen benannt, gespeichert und geschützt werden
- wie sie zur Auftrags-/Programmüberwachung eingesetzt werden können
- wie sie zur Bedingungsabhängigen Auftragssteuerung eingesetzt werden können
- wie sie bei der Eingabe von Kommandos und Anweisungen als Eingabequelle verwendet werden können.

Die Syntax und die Operanden der verwendeten Kommandos sind im Handbuch „Kommandos“ [1] beschrieben.

Es gibt zwei Typen von Jobvariablen:

- Benutzer-Jobvariablen und
- Sonder-Jobvariablen

Benutzer-Jobvariablen können als permanente oder, falls vom System erlaubt, als temporäre Jobvariablen angelegt werden.

Sie werden vom Benutzer alleinverantwortlich erstellt und mit Werten versorgt. Eine Ausnahme bilden Jobvariablen, die zur Überwachung von Aufträgen oder Programmen eingesetzt werden. Hier werden Informationen vom System eingetragen.

Sonder-Jobvariablen (siehe [Seite 44](#)) enthalten bestimmte Informationen des Betriebssystems, die vom Benutzer nur abgefragt bzw. in eine Benutzer-Jobvariable übertragen werden können.

Kommando	Makro	Funktion
CREATE-JV	CATJV	Jobvariable neu erzeugen
COPY-JV	COPJV	Jobvariable kopieren
DELETE-JV	ERAJV	Jobvariable löschen
MODIFY-JV	SETJV	Jobvariableninhalt ändern
MODIFY-JV-ATTRIBUTES	CATJV	Katalogeintrag für Jobvariable ändern
MODIFY-JV-CONDITIONALLY	CSWJV	Jobvariableninhalt bedingt ändern
MODIFY-MONJV	TIMJV	Elemente im Systemteil einer auftragsüberwachenden JV setzen
REMOVE-JV-LINK	RELJV	JV-LINK-Eintrag löschen
SET-JV-LINK	DCLJV	Jobvariablen-Kettungsnamen definieren
SHOW-JV	GETJV	Jobvariableninhalt ausgeben
SHOW-JV-ATTRIBUTES	STAJV	Jobvariablen-Attribute ausgeben
SHOW-JV-LINK	LNKJV	JV-LINK-Eintrag ausgeben

Tabelle 1: Kommandos bzw. Makroaufrufe zur Verwaltung von Jobvariablen

Die Kommandos `MODIFY-JV` und `SHOW-JV` stehen auch dem Operator zur Verfügung (siehe [Seite 237](#)).

2.1 Speichern von Jobvariablen

Eine Benutzer-Jobvariable wird mit dem Kommando CREATE-JV erzeugt. Für die Jobvariable wird dabei im Dateikatalog (TSOSCAT) ein Jobvariablen-Eintrag erzeugt. Der Eintrag enthält einen Verwaltungsteil (z.B. Schutzmerkmale und Name der Jobvariablen). Im Anschluss an den Verwaltungsteil sind 256 Bytes zur Speicherung des Jobvariablen-Wertes reserviert.

Jeder Jobvariablen-Eintrag in der Datei TSOSCAT enthält:

- Bytes für Kontrollinformationen
- eine feste Anzahl von Bytes für den Namen der Jobvariablen
- eine variable Anzahl von Bytes für den Wert (0 bis 256 Bytes)

2.1.1 Permanente Benutzer-Jobvariablen

Eine im Dateikatalog (TSOSCAT) eingetragene permanente Jobvariable (JV) existiert so lange, bis sie mit DELETE-JV gelöscht wird, unabhängig von der Lebensdauer des erzeugenden Auftrags. Das bedeutet, eine Jobvariable kann von einem Auftrag erzeugt, von einem anderen verwendet und von einem Dritten gelöscht werden, ohne dass die Aufträge jemals gleichzeitig existierten.

2.1.2 Temporäre Benutzer-Jobvariablen

Mit dem Klasse-2-Systemparameter TEMPFILE (siehe Handbuch „Einführung in die Systembetreuung“ [3]) kann die Systembetreuung die Verwendung temporärer Dateien und Jobvariablen zulassen. Wird z.B. TEMPFILE='#' gesetzt, werden alle Jobvariablen und Dateien, deren Name mit dem Zeichen # beginnt, als „temporär“ gewertet, d.h. sie werden bei Auftragsende für den erzeugenden Auftrag automatisch gelöscht. Standardeinstellung im System ist TEMPFILE='NO', d.h. es sind keine temporären Dateien und Jobvariablen zugelassen. Zulässige Werte für TEMPFILE sind # und @. In den Beispielen wird die Einstellung TEMPFILE='#' angenommen.

Nur der erzeugende Auftrag kann auf temporäre Jobvariablen zugreifen.

2.1.3 Sonder-Jobvariablen

Für Sonder-Jobvariablen existiert **kein** Eintrag im Dateikatalog.

Beim „Lesen“ einer Sonder-Jobvariablen werden die entsprechenden Daten aus Systemtabellen ausgegeben (z.B. Datum, Uhrzeit; mögliche Informationen siehe [Seite 44](#)).

2.2 Benennen von Benutzer-Jobvariablen

2.2.1 Permanente Benutzer-Jobvariablen

Jobvariablen werden ähnlich wie Dateien mithilfe einer Zeichenfolge identifiziert, die aus drei durch Begrenzer getrennten Teilen besteht:

1. Katalogkennung (catid) bezeichnet den Pubset (PVS), in dessen Dateikatalog (TSOSCAT) die Jobvariable gespeichert ist.
2. Benutzerkennung (userid) bezeichnet den Benutzereintrag im Benutzerkatalog (SYSSRPM) des Pubsets.
3. Jobvariablen-Name (jvname) identifiziert die Jobvariable innerhalb der Benutzerkennung eindeutig.

Diese dreiteilige Zeichenkette heißt „Pfadname“.
Sie hat folgendes Format:

```
: c a t i d : $ u s e r i d . j v n a m e
|<----- Pfadname ----->|
```

- Folgende Zeichen sind als Begrenzer zu verwenden:
Eine angegebene Katalogkennung „catid“ ist in Doppelpunkte einzuschließen.
Eine angegebene Benutzerkennung „userid“ ist zwischen ein Dollarzeichen und einen Punkt zu setzen.
- Ein Pfadname heißt unvollständig, wenn wenigstens einer der beiden Teile „:catid:“ oder „\$userid.“ nicht vorhanden ist; andernfalls heißt er vollständig.
- Wird ein Pfadname unvollständig angegeben, so vervollständigt ihn das DVS mit Standardwerten. Eine fehlende „catid“ wird durch die Standard-Katalogkennung der Benutzerkennung, eine fehlende „userid“ durch die Benutzerkennung des anfordernden Auftrags ergänzt.
- Die Standard-Katalogkennung ist im Benutzereintrag festgelegt. Sie kann mit dem Kommando SHOW-USER-ATTRIBUTES abgefragt werden.
- Maximale Längen einschließlich Begrenzer:

```
:catid:      6 Zeichen
$userid.    10 Zeichen
jvname      41 Zeichen
```


Die Gesamtlänge des Pfadnamens darf 54 Zeichen nicht überschreiten. Die maximale Länge von `jobname` verkürzt sich folglich, wenn `:catid:$userid` länger als 13 Zeichen ist. Bei vierstelliger Katalogkennung (6 Zeichen mit Begrenzer) und achtstelliger Benutzerkennung (10 Zeichen mit Begrenzer) kann `jobname` nur noch 38 Zeichen lang sein. Deshalb sollte `jobname` aus maximal 38 Zeichen bestehen, damit ein Wechsel der Katalog- oder Benutzerkennung immer möglich ist.

- Wird nur „\$.“ (nur Begrenzer ohne `userid`) angegeben, sucht das System die Jobvariable unter der Kennung, die mit dem Klasse-2-Systemparameter `DEFLUID` benannt wurde.

Standardeinstellung: `DEFLUID=TSOS`.

- Ein Pfadname heißt teilqualifiziert, wenn „`jobname`“ nicht angegeben ist oder mit einem Punkt endet; andernfalls heißt er vollqualifiziert.

- In `jobname` sind die gleichen Zeichen erlaubt wie für Dateinamen:

Alle Buchstaben

Ziffern 0 ... 9

Sonderzeichen - , @ , # , \$, .

Der Jobvariablenname muss mindestens einen Buchstaben enthalten.

Das erste Zeichen für eine permanente Jobvariable kann nicht das `TEMPFILE`-Zeichen sein. Die Zeichen „-“ und „\$“ sind als erstes Zeichen nicht möglich.

Beispiel

Dieses Beispiel soll anhand unterschiedlicher Eingaben von Pfadnamen die Begriffe teil-/vollqualifiziert und unvollständiger/vollständiger Pfadname erklären.

JV-Name	teilqualifiziert	vollqualifiziert	unvollständig	vollständig
<code>:V:\$PM211052.JOBVAR</code>		x		x
<code>\$PM211194.JV.JOB.STEUER</code>		x	x	
<code>:A:HILFSVARIABLE</code>		x	x	
<code>JOBV.VARIABLE.A</code>		x	x	
<code>:V:</code>	x		x	
<code>\$PM211052.</code>	x		x	
<code>JOBV.A.</code>	x		x	
<code>JOB.MON.</code>	x		x	

Tabelle 2: Beispiele für Pfadnamen-Angaben

2.2.2 Wildcard-Syntax in Pfadnamen

In folgenden Kommandos bzw. Makroaufrufen können innerhalb eines Pfadnamens Musterzeichen (Wildcards) stellvertretend für einzelne oder mehrere Zeichen verwendet werden:

- DELETE-JV bzw. ERAJV
- SHOW-JV-ATTRIBUTES bzw. STAJV
- SHOW-JV-LINK bzw. LNKJV

Musterzeichen können im Namen von temporären und permanenten Benutzer-Jobvariablen sowie im Namen von Sonderjobvariablen angegeben werden. Für den nicht privilegierten Benutzer ist die Verwendung innerhalb der Katalogkennung und des Jobvariablen-Namens zulässig. Bei Sonderjobvariablen dürfen Musterzeichen nicht im Namensteil \$SYSJV verwendet werden.

Musterzeichen können keine Begrenzer im Pfadnamen ersetzen. In Kommandos wird die zulässige Verwendung von Musterzeichen durch den Zusatz with-wild (n) (with-wildcards) in den Datentypen <filename> und <partial-filename> angezeigt. Die angegebene Länge n=80 bzw. n=79 bezieht sich dabei nur auf die Länge der Eingabe. Durch Verwendung von Musterzeichen ist es möglich, dass der eingegebene Pfadname einschließlich verwendeter Musterzeichen länger als 54 Zeichen ist. Die durch diese Zeichenfolge abgebildeten Pfadnamen können trotzdem nur maximal 54 Zeichen enthalten. Die Angabe von Musterzeichen in Benutzerkennungen ist dem privilegierten Benutzer vorbehalten. Zur Angabe von Musterzeichen siehe Handbuch „Kommandos“ [1].

Beispiele

SHOW-JV-ATTR JV=*ALL

zeigt alle permanenten Jobvariablen der Benutzerkennung, sowie in einer Summenzeile die Anzahl aller permanenten Jobvariablen und die Gesamtlänge ihrer Werte.

SHOW-JV-ATTR JV=**ALL

zeigt alle JV, die mit der Zeichenfolge „ALL“ enden. Das Musterzeichen „*“ zu Beginn der Zeichenfolge muss doppelt angegeben werden!

SHOW-JV-ATTR JV=///

zeigt alle Jobvariablen, deren Name genau drei Zeichen lang ist, sowie in einer Summenzeile die Anzahl aller drei Zeichen langen Jobvariablen und die Gesamtlänge ihrer Werte.

(z.B. :catid:\$userid.ABC, :catid:\$userid.DEF, usw.).

SHOW-JV-ATTR JV=<D,M:O>*

zeigt alle Jobvariablen, deren Name entweder mit D oder mit einem Zeichen aus dem Bereich M bis O beginnt, sowie in einer Summenzeile die Anzahl dieser Jobvariablen und die Gesamtlänge ihrer Werte.

(z.B. :catid:\$userid.DORA,:catid:\$userid.DIETER, :catid:\$userid.MARTHA, :catid:\$userid.NORDPOL, :catid:\$userid.OTTO).

SHOW-JV-ATTR JV=-*HA

zeigt alle Jobvariablen, die *nicht* auf die Zeichenfolge „HA“ enden, sowie in einer Summenzeile die Anzahl dieser Jobvariablen und die Gesamtlänge ihrer Werte.

(z.B. :catid:\$userid.ANTON, :catid:\$userid.HANS, usw.).

SHOW-JV-ATTR \$SYSJV.M*

zeigt alle Sonderjobvariablen, die mit M beginnen.

(z.B. \$SYSJV.MONAT und \$SYSJV.MONTH)

2.2.3 Temporäre Benutzer-Jobvariablen

Der Name einer temporären Jobvariablen beginnt mit dem im Systemparameter TEMPFILE festgelegten Zeichen (# oder @), dem ein bis zu 27 Zeichen langer jvname folgt, der auch teilqualifiziert sein darf: (nachfolgend wird # als vereinbartes TEMPFILE-Zeichen angenommen)

#jvname z.B. #JOBVARIABLE17
 #AUFTR.STEUER.JV2

Mit dem Sonderzeichen allein werden alle temporären Jobvariablen eines Auftrags angesprochen, z.B. SHOW-JV-ATTR JV=#.

Temporäre Jobvariablen werden unter der Standard-Katalogkennung und der Benutzerkennung des erzeugenden Auftrags unter einem internen Namen gespeichert. Nicht erlaubt ist die Angabe einer fremden Benutzerkennung oder Katalogkennung.

Der interne Name einer temporären Jobvariablen lautet:

S.sys.nnnn.jvname	nnnn	=	Task Sequence Number des erzeugenden Auftrags
	sys	=	interne Nummer des Rechner, da die TSN im Mehrrechnerverbund mit mehrbenutzbaren Pubsets (Shared-Pubset) nicht eindeutig ist

Der interne Namensteil S.sys.nnnn. wird von dem Benutzer durch das TEMPFILE-Zeichen spezifiziert. Vom System wird immer der **interne** Name ausgegeben.

Die Länge von „jvname“ kann maximal 27 Zeichen betragen.

Pfadname:	maximal 54 Zeichen
:catid:\$userid.S.sys.nnnn.jvname	(6+10+38=54)
S.sys.nnnn.jvname	(11+27=38) (interner Namensteil 11 Zeichen)
#jvname	('#+27)

Bei Statusabfragen SHOW-JV-ATTRIBUTES wird immer der **interne** JV-Name ausgegeben. Das Gleiche gilt für Systemmeldungen, die den JV-Namen enthalten.

Beispiel

```
/create-jv jv=#work
/show-jv-attr jv=#
%0000000 :10SN:$USER1.S.100.0MH4.WORK
%SUM      00001 JV'S; JV-VALUE = 00000000 BYTES
/del-jv jv=#,dialog-contr=*jv-change
% JVS0469 DELETE JOB VARIABLE ':10SN:$USER1.S.100.0MH4.WORK'? REPLY
(Y=YES; N=NO; T=TERMINATE; ,CHECK=NEW MODE)?y
/
```

Hinweis

Die Gestaltung des internen Namens von Jobvariablen und Dateien ist nicht Bestandteil der Benutzerschnittstelle und kann von Version zu Version ohne Vorankündigung geändert werden.

2.2.4 Kettungsnamen für Benutzer-Jobvariablen

Für Benutzer-Jobvariablen kann wie für Dateien ein Kettungsname vereinbart werden. Die Zuordnung einer Jobvariablen zu einem Kettungsnamen wird mit dem Kommando SET-JV-LINK vorgenommen. Dabei wird unter dem angegebenen Kettungsnamen in der JV-LINK-Tabelle des Auftrags der Pfadname der zugeordneten Jobvariablen gespeichert.

Die JV-LINK-Tabelle bleibt bis zum Ende des erzeugenden Auftrags bestehen.

Ein Kettungsname ist immer eindeutig einer Jobvariablen zugeordnet. Für eine Jobvariable können mehrere Einträge unter verschiedenen Kettungsnamen bestehen. Wird ein Kettungsname nochmals vergeben, so wird der bereits bestehende Eintrag überschrieben (implizites Löschen).

Die Kettungsnamen **SMONJVJ** und **SMONJVP** sollten nicht verwendet werden. Unter diesen Standard-Kettungsnamen erstellt das System Einträge in der JV-LINK-Tabelle des Auftrags bei Auftrags- bzw. Programmüberwachung (siehe [Seite 64](#)).

Ein gültiger Kettungsname besteht aus maximal sieben Zeichen (0..9,A..Z,#,@,\$), denen in allen Makros, bei Jobvariablen-Ersetzung und in den Kommandos der bedingungsabhängigen Auftragssteuerung ein Stern (*) vorangestellt werden muss. In den übrigen Kommandos werden die Kettungsnamen ohne Stern angegeben (aus Kompatibilitätsgründen ist allerdings die wahlweise Voranstellung des Sterns möglich).

Über den Kettungsnamen kann der erzeugende Auftrag in folgenden Kommandos bzw. Makros (siehe Tabelle 5) die jeweils zugeordnete Jobvariable ansprechen (über die Operanden `JV=*LINK` oder `LINK-NAME`).

Andere Aufträge haben keinen Zugriff über den vereinbarten Kettungsnamen. Mit Verwendung von Kettungsnamen können die Namen von verwendeten Jobvariablen in Programmen und Prozeduren variabel gehalten werden.

Mit dem Kommando `SHOW-JV-LINK` können die Einträge der `JV-LINK`-Tabelle ausgegeben werden. Dabei ist zu beachten, dass eingetragene Jobvariablen *nur* bei Erstellung des `JV-LINK`-Eintrages zugreifbar sein müssen, dass aber eine danach angezeigte Jobvariable durchaus nicht mehr zugreifbar sein kann (z.B. inzwischen gelöscht).

Mit dem Kommando `REMOVE-JV-LINK` können ein oder mehrere Einträge in der `JV-LINK`-Tabelle gelöscht werden.

Kommando	Makro	Funktion
<code>COPY-JV</code>	<code>COPJV</code>	Jobvariable kopieren
<code>DELETE-JV</code>	<code>ERAJV</code>	Jobvariable löschen
<code>MODIFY-JV</code>	<code>SETJV</code>	Jobvariableninhalt ändern
<code>MODIFY-JV-CONDITIONALLY</code>	<code>CSWJV</code>	Jobvariable prüfen und setzen
<code>REMOVE-JV-LINK</code>	<code>RELJV</code>	<code>JV-LINK</code> -Eintrag löschen
<code>SET-JV-LINK</code>	<code>DCLJV</code>	Jobvariablen-Kettungsnamen definieren
<code>SHOW-JV</code>	<code>GETJV</code>	Jobvariableninhalt ausgeben
<code>SHOW-JV-ATTRIBUTES</code>	<code>STAJV</code>	Jobvariablen-Attribute ausgeben
<code>SHOW-JV-LINK</code>	<code>LNKJV</code>	<code>JV-LINK</code> -Eintrag ausgeben

Tabelle 3: Kommandos/Makros, in denen ein Kettungsname verwendet werden kann

Kettungsnamen können außerdem in bedingten Ausdrücken (siehe [Seite 83](#)) und bei Jobvariablen-Ersetzung (siehe [Seite 70](#)) verwendet werden.

2.3 Zugriffsverwaltung für Jobvariablen

Der Eigentümer und eventuelle Mit-Eigentümer (die Begriffe werden im folgenden Abschnitt erläutert) können für jede Jobvariable festlegen, wer unter welchen Bedingungen auf die Jobvariable zugreifen darf. Damit können sie sowohl unberechtigtes Lesen als auch unberechtigtes Verändern einer Jobvariable verhindern.

Mit dem Produkt JV allein lassen sich die folgenden Schutzattribute der Standard-Zugriffskontrolle definieren:

- Zugriff nur für den Eigentümer oder für alle Benutzer
- Genereller Lese- oder Schreibzugriff
- Allgemeine Schutzfrist
- Lese- und Schreib-Kennwörter

Erheblich feinere Einstellungen der Zugriffsrechte sind mit dem Produkt SECOS bzw. mit dessen Bestandteilen GUARDS, GUARDDEF (Default-Protection) und GUARDCOO (Co-Owner Protection) möglich:

- Rechte zum Lesen und Schreiben von Jobvariablen festlegen mithilfe eines individuellen Zugriffsprofils (Guard) für jede Kennung
- Rechte zur Verwaltung von Jobvariablen festlegen: Mit GUARDCOO können Mit-Eigentümer von Jobvariablen festgelegt werden
- Festlegung von Default-Werten für Schutzmerkmale von Jobvariablen mit der Funktion „Default-Protection“

Diese Rechte können unabhängig voneinander an beliebig viele Benutzerkennungen oder Benutzergruppen vergeben werden.

Nur der erzeugende Auftrag oder die Systembetreuung können auf eine temporäre Jobvariable zugreifen. Für temporäre Jobvariablen können keine Schutzmerkmale abweichend von der Standardeinstellung bei CREATE-JV vereinbart werden.

2.3.1 Eigentümerschaft von Jobvariablen

Das Recht zur Verwaltung von Jobvariablen, also zum Anlegen, Ändern, Löschen und zur Festlegung der Attribute von Jobvariablen, haben der Eigentümer und gegebenenfalls weitere Mit-Eigentümer.

Der Eigentümer ist die Benutzerkennung, in deren Katalog eine Jobvariable angelegt wurde. Diese Eigentümerschaft kann nachträglich nicht verändert werden.

Der Mit-Eigentümer ist standardmäßig die Benutzerkennung TSOS. Diese Mit-Eigentümerschaft kann mit Hilfe des Produkts SECOS verändert werden: Es können weitere Benutzerkennungen als Mit-Eigentümer definiert werden. Außerdem kann auch die Mit-Eigentümerschaft von TSOS eingeschränkt werden. Eine Ausnahme bilden temporäre Jobvariablen; für diese ist keine Steuerung der Mit-Eigentümerschaft möglich.

Einige Schutzattribute werden durch die Systemadministration (unabhängig von der Eigentümerschaft) festgelegt:

- Der Systemparameter FSHARING legt fest, ob Jobvariablen auch für Benutzerkennungen zugreifbar sind, die keinen Benutzereintrag auf dem Pubset der Jobvariable haben.
- Bei Einsatz von SECOS kann die Systembetreuung pubset-globale Voreinstellungen für Schutzattribute von Jobvariablen festlegen. Diese werden wirksam, wenn weder beim Einrichten einer Jobvariable noch über deren benutzerspezifische Voreinstellungen Schutzattribute vergeben wurden.

Steuerung der Mit-Eigentümerschaft

Die Mit-Eigentümerschaft ermöglicht das Anlegen, Modifizieren und Löschen von Jobvariablen unter einer fremden Benutzerkennung mit denselben Rechten, die auch der Eigentümer der Jobvariablen hat.

Mit dem Subsystem GUARDCOO ist es möglich, Mit-Eigentümer (Co-Owner) von Jobvariablen zu definieren. Die Mit-Eigentümerschaft ermöglicht das Anlegen, Modifizieren und Löschen von Jobvariablen unter fremder Benutzerkennung mit denselben Rechten wie der Eigentümer. Ist das Subsystem GUARDCOO nicht aktiviert oder tritt bei der Überprüfung der Zugriffsrechte ein Fehler auf, wird eine Mit-Eigentümerschaft ignoriert, d.h. es wirken nur die herkömmlichen Zugriffsrechte.

GUARDCOO ist Teil des Software-Produkts SECOS.

Die Definition der Mit-Eigentümer erfolgt mit GUARDS in einem zweistufigen Verfahren:

- Über Namensmuster (Regeln) in sogenannten Regelbehältern (Guards) werden Jobvariablen ausgewählt, für die Mit-Eigentümer festgelegt werden sollen. Die Namensmuster werden sowohl für bereits bestehende als auch für neu anzulegende, nicht aber für temporäre Jobvariablen ausgewertet.

- Für jeden Regelbehälter wird dann festgelegt, wer unter welchen Bedingungen Verwaltungsfunktionen ausführen kann. Die Miteigentümerschaft kann für bestimmte Benutzerkennungen, Mitglieder bestimmter Benutzergruppen oder Inhaber bestimmter systemglobaler Privilegien vergeben werden.

Näheres über die Steuerung der Mit-Eigentümerschaft mit SECOS finden Sie in den SECOS-Handbüchern (siehe „SECOS“ [10]) unter dem Stichwort „Miteigentümerschutz (Coowner-Protection)“).

Eingeschränkte TSOS-Miteigentümerschaft

Der Eigentümer kann die Zugriffs- und Verwaltungsrechte der Benutzerkennung TSOS einschränken. Unter der Benutzerkennung TSOS können dann die Schutzmerkmale einer fremden Jobvariablen (d.h. TSOS ist nicht der Eigentümer) nicht mehr verändert werden, auch das Löschen einer Jobvariablen unter Umgehung der Schutzattribute (IGNORE=ACCESS) ist nicht mehr möglich. Wenn für ein Namensmuster die TSOS-Mit-Eigentümerschaft auf *RESTRICTED gesetzt ist, dann wird die Angabe IGNORE=ACCESS ignoriert und statt dessen die Berechtigungsprüfung anhand der Schutzmerkmale für ACCESS, BASIC-ACL und GUARDS verwendet.

Siehe dazu auch das GUARDS-Kommando /ADD-COOWNER-PROTECTION-RULE im Handbuch „SECOS“ [9]. Betroffen durch die Einschränkung sind primär die Jobvariablen-Kommandos /DELETE-JV und /MODIFY-JV-ATTRIBUTES (siehe Handbuch „Kommandos“ [1]).

Unter der Benutzerkennung TSOS ist die Einschränkung der Mit-Eigentümerschaft bei Jobvariablen wirkungslos, weil TSOS der (unveränderliche) Eigentümer der Jobvariablen ist.

Näheres über die Einschränkung der Mit-Eigentümerschaft mit SECOS finden Sie in den SECOS-Handbüchern (siehe „SECOS“ [9]) unter dem Stichwort „Miteigentümerschutz (Coowner-Protection)“).

2.3.2 Schutzmechanismen zur Zugriffssteuerung

Bei einer Jobvariable können folgende Zugriffsmerkmale gesteuert werden (in Klammern die Bezeichnung der Schutzmöglichkeiten):

- Zugriffsberechtigte Benutzer bzw. Benutzergruppen (Mehrbenutzbarkeit, Mehrbenutzerzugriff)
- Lese- und/oder Schreibberechtigung ggf. nach Zugriffsberechtigten differenziert (Zugriffsart, Mehrbenutzerzugriff)
- Kennwörter für den Zugriff (Kennwörter)
- Zeitliche Befristung des Zugriffs (Schutzfrist)

Schutzmechanismus	Schutzmöglichkeiten	Granularität der Zugriffsberechtigten
GUARDS ↑	– Weitere Verfeinerung von Mehrbenutzerzugriff und Zugriffsart	individuelle Zugriffsprofile ↑
BASIC-ACL ↑	– Verfeinerung von Mehrbenutzerzugriff und Zugriffsart	Zugriffsprofile für Gruppen ↑
Standardschutz	Standardzugriffskontrolle – Zugriffsart (Lese-/Schreib-Zugriff) – Mehrbenutzbarkeit Weitere Schutzmöglichkeiten – Kennwörter – Schutzfrist	Eigentümer oder alle

Tabelle 4: Übersicht der Schutzmöglichkeiten für eine Jobvariable

Die Schutzmöglichkeiten für eine Jobvariable sind auf den drei Ebenen Standardschutz (unterste Ebene), BASIC-ACL und GUARDS (höchste Ebene) angeordnet (siehe [Tabelle 4](#)). Für den Schutz einer Jobvariable gilt immer der stärkste aktivierte Schutzmechanismus.

Der Einsatz von Kennwörtern und der Schutzfrist (Freigabedatum) ist unabhängig vom aktiven Schutzmechanismus, d.h. er ist auch bei BASIC-ACL und GUARDS möglich. Die Zugriffskontrolle (Mehrbenutzbarkeit und Zugriffsart) wird bei diesen beiden Schutzmechanismen verfeinert, indem bei BASIC-ACL drei Gruppen (OWNER, GROUP und OTHERS) sowie bei GUARDS Zugriffsprofile für einzelne Benutzer gebildet werden.

Koexistenz der Zugriffsmechanismen



Bild 1: Hierarchie der Schutzmechanismen für Jobvariablen

Bild 1 verdeutlicht, dass die jeweils höhere Schutzebene die Funktionalität der darunter liegenden Schicht voll abdeckt.

Die Koexistenz zwischen Standardzugriffskontrolle (USER-ACCESS, ACCESS), BASIC-ACL und Guards wird folgendermaßen geregelt:

1. Die Werte der USER-ACCESS- bzw. ACCESS-Operanden werden in den Katalog eingetragen, unabhängig davon, ob sie im aktuellen Schutzzustand durch höherwertige Schutzmechanismen (BASIC-ACL oder Guards) auch aktuell wirksam werden. Besteht ein höher priorisierter Schutz, werden die Einträge erst nach dessen Deaktivierung berücksichtigt.
2. Der BASIC-ACL-Operand setzt die BASIC-ACL-Indikatoren im Katalogeintrag der Jobvariable unabhängig davon, ob der Schutz der Jobvariable durch BASIC-ACL oder einen höher priorisierten Schutz (Guards) bestimmt wird. Besteht ein höher priorisierter Schutz, werden die Einträge erst nach dessen Deaktivierung berücksichtigt.

Standard-Zugriffsschutz

Die Standard-Zugriffskontrolle bietet die Schutzmerkmale Zugriffsart und Mehrbenutzbarkeit. Darüber hinaus können im Standard-Zugriffsschutz (aber auch bei BASIC-ACL und GUARDS) eine Schutzfrist und Kennwortschutz für eine Jobvariable festgelegt werden.

Zugriffsart

Mit dem Operanden der Kommandos CREATE-JV und MODIFY-JV-ATTRIBUTES kann ein Benutzer festlegen, welche Zugriffsart er auf eine Jobvariable erlaubt.

ACCESS=*BY-PROTECTION-ATTR (Standardwert) Die Zugriffsart erfolgt abhängig vom Wert des Operanden PROTECTION-ATTR (siehe [Abschnitt „Default-Protection \(Benutzerdefinierte Standardwerte\)“](#) auf Seite 36)

ACCESS=WRITE Lese- und Schreibzugriffe erlaubt

ACCESS=READ Lesezugriff erlaubt

Auf Makro-Ebene (Makro CATJV) entspricht der Operandenwert PROTECT=DEFAULT dem Operandenwert ACCESS=*BY-PROTECTION-ATTR bei den Kommandos. Der PROTECT-Operand bezieht sich jedoch nicht nur auf die Zugriffsart, sondern auf alle Schutzmerkmale. Die Operandenwerte für Lese- und Schreibzugriff sind identisch mit denen der Kommando-Operanden.

Mehrbenutzbarkeit

In den Kommandos CREATE-JV und MODIFY-JV-ATTRIBUTES wird mit dem Operanden USER-ACCESS festgelegt, welche Benutzerkennungen auf die Jobvariable mit den zuvor vergebenen Zugriffstypen zugreifen können (Mehrbenutzbarkeit):

USER-ACCESS=*BY-PROTECTION-ATTR (Standardwert) Der Zugriff erfolgt abhängig vom Wert des Operanden PROTECTION-ATTR (siehe [Abschnitt „Default-Protection \(Benutzerdefinierte Standardwerte\)“](#) auf Seite 36)

USER-ACCESS=*OWNER-ONLY Nur der JV-Eigentümer kann zugreifen

USER-ACCESS=*ALL-USERS Alle Benutzerkennungen dürfen zugreifen

Auf Makro-Ebene (Makro CATJV) entsprechen die Operandenwerte SHARE=NO / YES den Werten USER-ACCESS=*OWNER-ONLY/ *ALL-USERS.

Schutzfrist

Mit Vereinbarung einer Schutzfrist kann eine Jobvariable für einen festgesetzten Zeitraum gegen Veränderung und Löschen geschützt werden.

Beim Erzeugen der Jobvariablen wird im Ausgabefeld EXPIR-DATE des Kommandos SHOW-JV-ATTRIBUTES standardmäßig das aktuelle Tagesdatum eingetragen. Dieses Datum wird automatisch aktualisiert, wenn der Inhalt der Jobvariablen verändert wird.

Mit dem Kommando MODIFY-JV-ATTRIBUTES kann im Operanden RETENTION-PERIOD angegeben werden, wie viele Tage ab dem aktuellen Tagesdatum die Jobvariable gegen Wertänderungen und gegen Löschen geschützt sein soll. Bei Vereinbarung einer Schutzfrist wird das Tagesdatum um die angegebene Anzahl Tage hochgesetzt. Das Datum (in EXPIR-DATE) zeigt jetzt an, wann die Jobvariable erstmals wieder verändert werden darf. Der Eintrag im Ausgabefeld EXPIR-DATE bleibt bis Ablauf der Schutzfrist unverändert. Danach wird er wieder automatisch dem aktuellen Tagesdatum angepasst, wenn der Inhalt der Jobvariablen verändert wird.

Auf Makro-Ebene (Makro CATJV) entspricht der Operand RETPD dem Operanden RETENTION-PERIOD.

Kennwortschutz

Zusätzlich kann der Zugriff auf eine Jobvariable von der Kenntnis eines Kennwortes abhängig gemacht werden.

Im Kommando CREATE-JV bzw. MODIFY-JV-ATTRIBUTES kann im Operanden READ-PASSWORD ein Lese-Kennwort und im Operanden WRITE-PASSWORD ein Schreib-Kennwort vereinbart werden.

Im Jobvariablen-Eintrag wird in den Feldern READ-PASS und WRITE-PASS mit dem Wert YES angezeigt, dass ein Lese- bzw. Schreib-Kennwort vereinbart wurde. Ist kein Kennwort vereinbart, so ist der Wert NONE.

Standardmäßig ist eine neu erzeugte Jobvariable nicht mit Kennwörtern geschützt.

Vor dem Zugriff ist das erforderliche Kennwort mit dem Kommando ADD-PASSWORD in die Kennwort-Tabelle des Auftrags einzutragen. Dieser Eintrag hat Gültigkeit bis Auftragsende. In einigen Kommandos kann das Kennwort im Operanden PASSWORD bzw. JV-PASSWORD angegeben werden. Die Zugriffsberechtigung besteht dann nur für Zugriffe bei Ausführung des Kommandos.

Ist ein Kennwort vereinbart, muss dieses von allen Benutzern ohne Systemverwaltungsrechte angegeben werden, auch wenn mit dem Kommando MODIFY-JV-ATTRIBUTES Schutzmerkmale geändert werden sollen.

Kennwortschutz		zusätzlicher Schutz durch Kennwort	erforderliche Angaben für	
READ-PASS	WRITE-PASS		Lesezugriff	Schreibzugriff
NONE	NONE	kein Schutz	keine	keine
YES	NONE	Lesezugriff Schreibzugriff	Lese-Kennwort	Lese-Kennwort
NONE	YES	Schreibzugriff	keine	Schreib-Kennwort
YES	YES	Lesezugriff Schreibzugriff	Lese- oder Schreib-Kennwort	Schreib-Kennwort

Tabelle 5: Kennwortschutz-Kombinationen und Angaben für Lese- oder Schreibzugriff

Bei dem Versuch, auf eine Jobvariable ohne das erforderliche Kennwort zuzugreifen, wird der nicht privilegierte Benutzer mit einer Zeitstrafe belegt. Die Dauer der Zeitstrafe ist abhängig von dem eingestellten Systemparameter PWPENTI und kann einen Wert von null bis 60 Sekunden annehmen. Während dieser Zeit ist keine weitere Eingabe möglich.

In den Kommandos SHOW-JV, MODIFY-JV, MODIFY-JV-CONDITIONALLY, MODIFY-MONJV, DELETE-JV und den entsprechenden Makros kann das Kennwort noch im Kommando angegeben werden (Operand PASSWORD). Bei der Angabe eines falschen Kennwortes wird geprüft, ob das Kennwort bereits in der Kennwort-Tabelle enthalten ist. Ist das Kennwort dort nicht enthalten ist, handelt es sich um einen unberechtigten Zugriffsversuch. Im anderen Fall ist der Zugriff berechtigt und das Kommando wird ausgeführt.

Auf Makro-Ebene (Makro CATJV) entsprechen die Operanden RDPASS und WRPASS den Operanden READ-PASSWORD und WRITE-PASSWORD.

Schnittstellenübersicht

Kommando/Makro	Funktion
CREATE-JV/CATJV	Jobvariable anlegen und Festlegen von Schutzmerkmalen mit den Kommando-Operanden ACCESS,USERACCESS, READ-PASSWORD und WRITE-PASSWORD bzw. mit den Makro-Operanden ACCESS, SHARE, RDPASS und WRPASS.
MODIFY-JV-ATTRIBUTES / CATJV	Ändern der Schutzmerkmale mit den Kommando-Operanden ACCESS,USER-ACCESS, READ-PASSWORD, WRITE-PASSWORD und RETENTION-PERIOD bzw. mit den Makro-Operanden ACCESS, SHARE, RDPASS, WRPASS und RETPD.
SHOW-JV-ATTRIBUTES / STAJV	Ausgeben der Schutzmerkmale Die Jobvariablen können auch nach bestimmten Schutzmerkmalen selektiert werden, und zwar mit den Kommando-Operanden ACCESS, USER-ACCESS, PASSWORD und EXPIRATION-DATE innerhalb von SELECT=*BY-ATTRIBUTES(...). Auf Makroebene ist die Selektion im STAJV-Makro über den Makro JVSEL mit den Operanden ACCESS, SHARE, PASS und EXDATE möglich.

Tabelle 6: Kommandos und Makros zur Festlegung von Standardschutz-Merkmalen

Die Tabelle gibt einen zusammenfassenden Überblick über die Kommandos und Makros zur Festlegung der Standardschutzmerkmale Zugriffsart (ACCESS), Mehrbenutzbarkeit (USER-ACCESS / SHARE), Schutzfrist (RETENTION-PERIOD / RETPD) und Kennwortschutz (READ-PASSWORD und WRITE-PASSWORD / RDPASS und WRPASS).

Beispiel

```
/create-jv jv=status,prot=*par(access=*write,user-access=*all-users,
read-pass=c'rdpw',write-pass=c'wrpw')
```

Die Jobvariable status wird angelegt mit folgenden Eigenschaften:

- Lese- und Schreibzugriff sind erlaubt
- Alle Benutzerkennungen dürfen zugreifen
- Ein Leseschutz-Kennwort ('rdpw') wird gesetzt
- Ein Schreibschutz-Kennwort ('wrpw') wird gesetzt

BASIC-ACL-Schutz

In einer BASIC-ACL (einfache Zugriffskontroll-Liste) können die Zugriffsrechte Lesen und Schreiben für die Benutzerklassen OWNER, GROUP und OTHERS explizit vergeben werden.

Diese Schutz-Möglichkeit besteht nicht für temporäre Jobvariablen.

Bei aktiviertem BASIC-ACL-Schutz muss für einen Zugriff jeweils das erforderliche Zugriffsrecht explizit gesetzt sein. Im Gegensatz zur Standard-Zugriffskontrolle (mit ACCESS und USER-ACCESS) folgt in einer BASIC-ACL aus dem Schreibrecht *nicht* implizit das Lese-recht. Für den Lesezugriff muss das Leserecht explizit gesetzt sein.

Die Benutzer werden in folgende Benutzerklassen eingeteilt:

OWNER ist die Benutzerkennung des Eigentümers bzw. der Systembetreuung.

GROUP sind alle Benutzerkennungen, die zur Benutzergruppe des Eigentümers gehören. In der Grundfunktion (ohne SECOS) sind dies alle anderen Benutzerkennungen, da nur eine Benutzergruppe existiert. Bei Einsatz des Software-Produktes SECOS können mit der Funktionseinheit SRPM mehrere Benutzergruppen definiert werden. Zugriffe von Benutzern, die nicht explizit zu einer anderen Gruppe als der des Eigentümers gehören, werden weiterhin gemäß der Zugriffsrechte von GROUP ausgewertet (Der Eigentümer einer Jobvariable kann nur maximal einer Gruppe angehören). Zugriffe der Benutzer, die explizit einer anderen Benutzergruppe angehören, werden dann gemäß der Zugriffsrechte von OTHERS ausgewertet.

OTHERS sind alle Benutzerkennungen, die explizit einer anderen Benutzergruppe als der des Eigentümers oder keiner Benutzergruppe angehören. Falls SECOS nicht eingesetzt wird, sollten die Zugriffsrechte wie für GROUP vergeben werden, um Änderungen bei zukünftigem Einsatz von SECOS zu vermeiden.

Die BASIC-ACL-Schutzmerkmale werden nur gesetzt, wenn mindestens eine Zugriffsberechtigung explizit vergeben wurde (BASIC-ACL wird aktiviert). Mit dem Kommando SHOW-JV-ATTRIBUTES werden die BASIC-ACL-Schutzmerkmale in den Ausgabefeldern OWNER, GROUP und OTHERS angezeigt. Je Benutzerklasse können die Werte „R W“ (lesen und schreiben), „R -“ (nur lesen), „- W“ (nur schreiben) oder „- -“ (kein Zugriff) angezeigt werden. Die Felder werden nur bei aktivierter BASIC-ACL angezeigt.

Mit dem Kommando CREATE-JV bzw. MODIFY-JV-ATTRIBUTES können im Operanden BASIC-ACL mit explizitem Setzen von Zugriffsrechten der BASIC-ACL-Schutz aktiviert, Zugriffsrechte verändert, oder der BASIC-ACL-Schutz wieder ausgeschaltet werden. Standardmäßig wird eine Jobvariable ohne BASIC-ACL erzeugt.

Bei aktivierter BASIC-ACL erfolgt die Zugriffskontrolle entsprechend den gesetzten Zugriffsrechten. Die Standard-Schutzmerkmale Mehrbenutzbarkeit und Zugriffsart werden dann nicht ausgewertet.

Bei Einsatz von SECOS können mehrere Benutzergruppen existieren. Damit besteht auch die Möglichkeit, die Zugriffsrechte der Gruppe des Eigentümers und anderer Benutzergruppen unterschiedlich zu vergeben.

Bei Aktivierung des BASIC-ACL-Schutzes für eine bereits bestehende Jobvariable kann der Benutzer mit der Angabe BASIC-ACL=*PREVIOUS eine BASIC-ACL erstellen, die den Werten der Standard-Zugriffskontrolle entspricht (siehe Kommando MODIFY-JV-ATTRIBUTES im Handbuch „Kommandos“ [1]). Beim Erzeugen einer Jobvariablen kann mit der Angabe BASIC-ACL=*STD eine BASIC-ACL erstellt werden, in der nur der Eigentümer alle Zugriffsrechte besitzt.

Die Makros und ihre Operanden zur Festlegung des BASIC-ACL-Schutzes sind (zusammen mit den Kommandos) in der folgenden Schnittstellenübersicht aufgeführt.

Schnittstellenübersicht

Kommando/Makro	Funktion
CREATE-JV/CATJV	Jobvariable anlegen und festlegen von Schutzmerkmalen mit den Kommando-Operanden BASIC-ACL (Unteroperanden OWNER, GROUP und OTHERS), READ-PASSWORD und WRITE-PASSWORD bzw. mit den Makro-Operanden OWNERAR, GROUPAR, OTHERAR, RDPASS und WRPASS.
MODIFY-JV-ATTRIBUTES / CATJV	Ändern der Schutzmerkmale mit den Kommando-Operanden BASIC-ACL (Unteroperanden OWNER, GROUP und OTHERS), READ-PASSWORD, WRITE-PASSWORD und RETENTION-PERIOD bzw. mit den Makro-Operanden OWNERAR, GROUPAR, OTHERAR, RDPASS, WRPASS und RETPD.
SHOW-JV-ATTRIBUTES / STAJV	Ausgeben der Schutzmerkmale Die Jobvariablen können auch nach bestimmten Schutzmerkmalen selektiert werden, und zwar mit den Kommando-Operanden BASIC-ACL (Unteroperanden OWNER, GROUP und OTHERS), PASSWORD und EXPIRATION-DATE innerhalb von SELECT=*BY-ATTRIBUTES(...). Auf Makroebene ist die Selektion im STAJV-Makro über den Makro JVSEL mit den Operanden BASACL, OWNERAR, GROUPAR, OTHERAR, PASS und EXDATE möglich.
ADD-USER-GROUP (SECOS-Kommando)	Benutzergruppe in den Benutzerkatalog eines Pubsets eintragen und Zuordnung von Benutzerkennungen zu einer Benutzergruppe
REMOVE-USER-GROUP (SECOS-Kommando)	Benutzergruppe löschen
SHOW-USER-GROUP (SECOS-Kommando)	Benutzergruppeneintrag anzeigen

Tabelle 7: Kommandos und Makros zur Festlegung des Zugriffsschutzes mit BASIC-ACL-Schutz

Beispiel

```

/create-jv jv=jv.develop _____ (1)
/add-user-group group-id=develop1,add-group-member=user1 _____ (2)
/mod-jv-attr jv=jv.develop,protection=(basic-acl=(
    owner=( read=*yes,write=*yes),
    group=( read=*yes)
    others=( read=*no) _____ (3)

```

- (1) Eine Jobvariable „jv.develop“ wird erzeugt
- (2) Eine Gruppe mit dem Namen „develop1“ wird vom Eigentümer der Jobvariable „jv.develop“ angelegt und die eigene Benutzerkennung (hier USER1) als Gruppenmitglied eingetragen.
- (3) Für die Jobvariable „jv.develop“ wird der Zugriffsschutz über eine BASIC-ACL vereinbart. Der Eigentümer der Jobvariable darf lesend und schreibend auf sie zugreifen. Die Mitglieder der Gruppe des Jobvariablen-Eigentümers („develop1“) dürfen die Jobvariable nur lesen. Alle anderen Benutzerkennungen („others“) dürfen nicht auf die Jobvariable zugreifen.

GUARDS-Schutz

Die Zugriffskontrolle für Jobvariablen kann über Guards (**Generally Usable Access ContRol ADministration System**) erfolgen. Dieser Zugriffsschutz wird nur wirksam, falls das Subsystem GUARDS geladen ist (Teil des Softwareproduktes SECOS, siehe [9]). Die Schutzmöglichkeit besteht nicht für temporäre Jobvariablen.

Der Zugriff auf eine Jobvariable wird über ein spezielles Zugriffsprofil (Guard) geregelt, das alle Bedingungen enthält, unter denen ein Zugriff erlaubt oder verwehrt wird (Datum, Uhrzeit, Zeitraum, Benutzerkennung). Jedes Zugriffsprofil wird mit entsprechenden GUARDS-Kommandos erstellt und unter einem vom Anwender vergebenen Namen im Guardskatalog als Eintrag abgespeichert. Jeder Pubset verfügt über einen Guardskatalog, der unabhängig von den Benutzerdateien verwaltet wird.

Der Zugriff auf eine mit einem Guard geschützte Jobvariable ist nur möglich, falls die im Guard-Eintrag festgelegten Bedingungen dies zulassen.

Aktivierung des Guards-Schutzes

Der Guard-Schutz wird aktiviert, wenn bei einem CREATE-JV-Aufruf der Operand GUARDS mit einem Wert ungleich *NONE angegeben wird.

Werden dabei nicht für alle Zugriffsrechte Guards zugewiesen, so werden die nicht angegebenen Zugriffsrechte auch als „nicht angegeben“ (*NONE) in den Dateikatalog eingetragen. Bei der Zugriffskontrolle wird ein entsprechender Zugriff abgewiesen, weil keine Rechte festgelegt wurden. Das Schreibrecht impliziert bei GUARDS nicht das Leserecht.

Die Makros und Operanden zur Festlegung des GUARDS-Schutzes sind (zusammen mit den Kommandos) in der folgenden Schnittstellenübersicht aufgeführt.

Schnittstellenübersicht

Kommando/Makro	Funktion
CREATE-JV/CATJV	<ul style="list-style-type: none"> – Jobvariable anlegen und Aktivierung des GUARDS-Schutzes mit den Unteroperanden READ und WRITE des Operanden GUARDS (Gleiche Operandenbezeichnungen bei Makro CATJV) – Festlegung von weiteren Schutzmerkmalen mit den Kommando-Operanden READ-PASSWORD und WRITE-PASSWORD bzw. mit den Makro-Operanden RDPASS und WRPASS
MODIFY-JV-ATTRIBUTES / CATJV	<ul style="list-style-type: none"> – Aktivierung/Deaktivierung des GUARDS-Schutzes mit den Unteroperanden READ und WRITE des Operanden GUARDS (Gleiche Operandenbezeichnungen bei Makro CATJV) – Festlegung von weiteren Schutzmerkmalen mit den Kommando-Operanden READ-PASSWORD, WRITE-PASSWORD und RETENTION-PERIOD bzw. mit den Makro-Operanden RDPASS, WRPASS und RETPD
SHOW-JV-ATTRIBUTES / STAJV	<p>Ausgeben der Schutzmerkmale</p> <p>Die Jobvariablen können auch nach bestimmten Schutzmerkmalen selektiert werden, und zwar mit den Kommando-Operanden GUARDS (Unteroperanden READ und WRITE), PASSWORD und EXPIRATION-DATE innerhalb von SELECT=*BY-ATTRIBUTES(...). Auf Makroebene ist die Selektion im STAJV-Makro über den Makro JVSEL mit den Operanden GUARDS (Unteroperanden READ und WRITE), PASS und EXDATE möglich.</p>
CREATE-GUARD (SECOS-Kommando)	Guard anlegen (enthält noch keinen Schutzmechanismus)
DELETE-GUARD (SECOS-Kommando)	Guard löschen

Tabelle 8: Kommandos und Makros zur Festlegung des Zugriffsschutzes mit GUARDS (Teil 1 von 2)

Kommando/Makro	Funktion
ADD-ACCESS-CONDITIONS (SECOS-Kommando)	Zugriffsbedingungen eines Guards festlegen
MODIFY-ACCESS-CONDITIONS (SECOS-Kommando)	Zugriffsbedingungen eines Guards ändern
SHOW-ACCESS-CONDITIONS (SECOS-Kommando)	Zugriffsbedingungen eines Guards ausgeben
SHOW-ACCESS-ADMISSION (SECOS-Kommando)	Zugriffsbedingungen eines Guards ausgeben, die für die aufrufende Benutzerkennung gültig sind

Tabelle 8: Kommandos und Makros zur Festlegung des Zugriffsschutzes mit GUARDS (Teil 2 von 2)

Verknüpft wird eine Jobvariable mit einem Guard-Eintrag, indem bei den entsprechenden Operanden des Makros/Kommandos (CATJV/CREATE-JV, MODIFY-JV-ATTRIBUTES) ein Guard-Name eingetragen wird. Die Verknüpfung wird im Katalogeintrag der Jobvariablen festgehalten.

Beispiel

```
/create-guard guard-name=protjv _____ (1)
/create-jv guardjv,protection=(guards=(read=protjv,write=*none)) _____ (2)
/add-access-conditions guard-name=protjv,
    subjects=*user(user-id=mueller),
    admission=*par(time=*interval(from=07:00,to=17:00)) (3)
```

- (1) Das Guard „protjv“ wird angelegt.
- (2) Die Jobvariable „guardjv“ wird angelegt und der Zugriffsschutz über GUARDS aktiviert durch die Verknüpfung mit dem Guard „protjv“. Der Leseschutz wird über diesen Guard geregelt, schreibende Zugriffe sind nicht erlaubt.
- (3) In das Guard „protjv“ wird eingetragen, dass der Benutzer „mueller“ (auf die Jobvariable „guardjv“) zugreifen darf und zwar zwischen 7 und 17 Uhr.
(Falls das angegebene Guard noch nicht existiert, wird es automatisch angelegt.)

Der Katalogeintrag für den Guard-Schutz kann auch ohne das Subsystem GUARDS erstellt werden. Es besteht dann jedoch kein Zugriffsrecht auf die Jobvariable.

Wird für eine Jobvariable ein Guard-Schutz aktiviert, werden alle nicht explizit gesetzten Zugriffsarten des GUARDS-Operanden (READ und/oder WRITE) mit *NONE belegt. Der Zugriff über diese Zugriffsarten ist dann **nicht** möglich. Beispielsweise wird bei Ausführung von `/modify-jv-attributes jvtest,protection=(guards=(read=protjv))` automatisch `protection=(guards=(read=protjv,write=*none))` gesetzt.

Erst zum Zeitpunkt des Zugriffs auf eine mit Guards geschützte Jobvariable wird geprüft, ob der angegebene Guard-Eintrag (Guard-Name) existiert, ob er verwendet werden darf und ob dem Benutzer auf Grund dieses Zugriffsprofils ein Zugriff in der entsprechenden Zugriffsart erlaubt ist.

Hinweis

Auf eine Jobvariable kann nicht zugegriffen werden, wenn in deren Katalogeintrag Schutz über Guards eingetragen ist, für den angegebenen Guard-Namen jedoch noch kein Zugriffsprofil im Guard-Katalog definiert ist. Gleiches gilt, wenn der Katalogeintrag für den Zugriffsschutz über Guards ohne das Softwareprodukt SECOS erstellt wurde.

Für Guards einer fremden Benutzerkennung muss die Zugriffsberechtigung bestehen. Die Zugriffsberechtigung für einen Guard wird erst geprüft, wenn dieser Guard zur Zugriffskontrolle benötigt wird.

Deaktivierung des Guards-Schutzes

Ein Guard-Eintrag wird nur durch die Angabe des Operanden GUARDS=*NONE (in den Kommandos CREATE-JV und MODIFY-JV-ATTRIBUTES) zurückgesetzt. Besitzt die Jobvariable anschließend keinen BASIC-ACL-Eintrag, wird der Zugriffsschutz der Jobvariable nur durch die Standardzugriffskontrolle, die Kennwörter und die Schutzfrist geregelt.

2.3.3 Default-Protection (Benutzerdefinierte Standardwerte)

Die Belegung der Schutzmerkmale von Jobvariablen soll oft vorgegebenen Mustern entsprechen. Mit der SECOS-Funktion „Default-Protection“ kann der Benutzer eigene Standardwerte definieren, die dann anstatt der Systemstandardwerte verwendet werden können. Diese Standardwerte können spezifisch für einen Benutzer oder pubsetglobal festgelegt werden. Sie werden in Attribut-Guards (Guards zur Festlegung von Standardwerten für Schutzmerkmale) hinterlegt.

Bei folgenden Schutzmerkmalen von Jobvariablen können über die Kommandos ADD-/MODIFY-DEFAULT-PROTECTION-ATTR Default-Protection-Werte gesetzt werden:

Schutzmerkmal	Bedeutung
ACCESS	Standard-Zugriffskontrolle (Zugriffsart)
USER-ACCESS	Standard-Zugriffskontrolle (Zugriff fremder Benutzer)
BASIC-ACL	Einfache Zugriffskontrollliste
GUARDS	Zugriffskontrolle über Guards
READ-PASS-WORD	Lesekennwort
WRITE-PASSWORD	Schreibkennwort

Tabelle 9: Schutzmerkmale von Jobvariablen

Jedes Attribut-Guard ist mit einem Regelbehälter verknüpft, der Regeln enthält, die festlegen, für welche Jobvariablen-Namen diese Vereinbarungen gültig sind. Siehe dazu auch die SECOS-Kommandos ADD-/MODIFY-DEFAULT- PROTECTION-RULE im Handbuch „SECOS“ [9]. Durch die Erstellung eines Regelbehälters wird die Default-Protection aktiviert.

Kommandos und Makros zur Anwendung von Default-Protection bei Jobvariablen

Default-Protection-Werte für Schutzmerkmale werden mit dem Kommando CREATE-JV PROTECTION=*STD (siehe Handbuch „Kommandos“ [1]) bzw. dem Makro CATJV ..., PROTECT=DEFAULT (siehe Seite 92) realisiert. Die Angaben *STD bzw. DEFAULT sind jeweils Voreinstellung.

Wird Default-Protection nicht aktiviert oder wird /CREATE-JV PROTECTION-ATTR=*STD bzw. CATJV ...,PROTECT=STD angegeben, gelten beim Erzeugen einer Jobvariablen (Ersteintrag von Schutzmerkmalen) System-Standardwerte. Eine Übernahme der Default-Protection-Werte für bereits existierende Jobvariablen ist nicht möglich. Die Schutzfrist (EXPIRATION-DATE) kann beim Ersteintrag mit Default-Protection nicht beeinflusst werden. Sie ist implizit mit *TODAY vorbelegt.

Die Default-Protection kann auf temporäre Jobvariablen nicht angewendet werden.

Auswirkungen der Operanden PROTECTION-ATTR (Kommando CREATE-JV) bzw. PROTECT (Makro CATJV) auf die Default-Werte von Schutzmerkmalen bei Jobvariablen:

- PROTECTION-ATTR=*STD
Für die System-Standardwerte der einzelnen Schutzmerkmale werden System-standardwerte (siehe [Tabelle 10](#)) verwendet, wenn nicht explizit anders vereinbart.
- PROTECTION-ATTR=*BY-DEF-PROT-OR-STD
 - Default-Protection ist aktiv: Verwendung der Default-Protection-Werte (aus dem Attribut-Guard)
 - Default-Protection ist nicht aktiv: Verwendung der System-Standardwerte (siehe [Tabelle 10](#))

Schutzmerkmal	System-Standardwerte
ACCESS	*WRITE
USER-ACCESS	*OWNER-ONLY
BASIC-ACL	*NONE
GUARDS	*NONE
READ-PASSWORD	*NONE
WRITE-PASSWORD	*NONE

Tabelle 10: System-Standardwerte für Jobvariablen-Schutzmerkmale

Beispiel

```

/add-default-protection-attr guard-name=defprot,user-access=*all-users (1)
/show-default-protection-attr d*_____ (2)
%-----
%GUARD :20SG:$USER1.DEFPROT                                DEFAULT PROTECTION
ATTRIBUTES
%-----
%                               % SCOPE: CREATE-OBJECT      % SCOPE: MODIFY-OBJECT-ATTR
%                               % -----                % -----
%ACCESS                          % *SYSTEM-STD              % *SYSTEM-STD
%USER-ACCESS                      % *ALL-USERS            % *ALL-USERS
%BASIC-ACL                        % *SYSTEM-STD          % *SYSTEM-STD
%GUARDS                          % *SYSTEM-STD          % *SYSTEM-STD
%READ-PASSWORD                   % *SYSTEM-STD          % *SYSTEM-STD
%WRITE-PASSWORD                  % *SYSTEM-STD          % *SYSTEM-STD
%EXEC-PASSWORD                   % *SYSTEM-STD          % *SYSTEM-STD
%DESTROY-BY-DELETE               % *SYSTEM-STD          % *SYSTEM-STD
%SPACE-RELEASE-LOCK              % *SYSTEM-STD          % *SYSTEM-STD
%EXPIRATION-DATE                 % *SYSTEM-STD          % *SYSTEM-STD
%FREE-FOR-DELETION               % *SYSTEM-STD          % *SYSTEM-STD
%-----
%GUARDS SELECTED: 1                                           END OF DISPLAY

```

```

/add-default-protection-rule rule-cont-guard=sys.udj,prot-rule=1stcharequj,
  protect-obj=*par(name=j*,attrib-guard=defprot) _____ (3)
/show-default-protection-rule _____ (4)
%-----
%RULE CONTAINER :20SG:$USER1.SYS.UDJ                USR ACTIVE  DEFAULT
PROTECTION
%-----
%1STCHAREQUJ   OBJECT      = J*
%              ATTRIBUTES  = $USER1.DEFPROT
%              USER-IDS    = *ANY-USER-ID
%-----
%RULE CONTAINER SELECTED: 1                                END OF DISPLAY

/create-jv jv.defprot _____ (5)
/show-jv-attributes jv.defprot,inf=*all _____ (6)
%0000000 :20SG:$USER1.JV.DEFPROT
% USER-ACC   = ALL-USERS   ACCESS    = WRITE
% CRE-DATE   = 2010-08-13  EXPIR-DATE = 2010-08-13
% CRE-TIME   = 14:38:42   EXPIR-TIME = 00:00:00
% READ-PASS  = NONE
% WRITE-PASS = NONE
%SUM        00001 JV'S; JV-VALUE = 00000000 BYTES

/create-jv no.defprot _____ (7)
/show-jv-attributes no.defprot,inf=*all _____ (8)
%0000000 :20SG:$USER1.NO.JVDEFPROT
% USER-ACC   = OWNER-ONLY  ACCESS    = WRITE
% CRE-DATE   = 2010-08-13  EXPIR-DATE = 2010-08-13
% CRE-TIME   = 14:41:00   EXPIR-TIME = 00:00:00
% READ-PASS  = NONE
% WRITE-PASS = NONE
%SUM        00001 JV'S; JV-VALUE = 00000000 BYTES

/create-jv jv.std-prot,protection-attr=*std _____ (9)
/show-jv-attributes jv.std-prot,inf=*all _____ (10)
%0000000 :20SG:$USER1.JV.STD-PROT
% USER-ACC   = OWNER-ONLY  ACCESS    = WRITE
% CRE-DATE   = 2010-08-13  EXPIR-DATE = 2010-08-13
% CRE-TIME   = 14:42:04   EXPIR-TIME = 00:00:00
% READ-PASS  = NONE
% WRITE-PASS = NONE
%SUM        00001 JV'S; JV-VALUE = 00000000 BYTES

```

- (1) Der Attribut-Guard „DEFPROT“ wird angelegt, dabei wird der Standardwert für das Schutzmerkmal ACCESS festgelegt.

- (2) Die Standardwerte für die Schutzmerkmale werden angezeigt. In der Spalte „SCOPE: CREATE OBJECT“ werden die Werte angezeigt, die beim Erzeugen einer Jobvariable gelten. Für das Schutzmerkmal USER-ACCESS ist als benutzerspezifischer Wert „*ALL-USERS“ eingetragen. Für alle übrigen Schutzmerkmale gelten die System-Standardwerte (Die Spalte „SCOPE: MODIFY-OBJECT-ATTR“ hat für Jobvariablen keine Bedeutung, sondern nur für Dateien).
- (3) Die Standardschutzregel „1STCHAREQUJ“ wird dem Regelbehälter „SYS.UDJ“ (J steht für Jobvariable) hinzugefügt.
Über „PROTECT-OBJ=*PAR(NAME=j*, ATTRIB-GUARD=DEFPROT“ wird festgelegt, dass diese Regel für alle Jobvariablen gilt, deren Namen mit J beginnt und dass die Jobvariablen für die diese Regel gilt, ihre Standardwerte aus dem Attribut-Guard „DEFPROT“ beziehen.
- (4) Die Eigenschaften des Regelbehälters und der enthaltenen Regel werden angezeigt.
- (5) Die Jobvariable „JV.DEFPROT“ wird angelegt. Ihr Name entspricht der Regel „1STCHAREQUJ“ und somit werden die Default-Werte aus dem Attribut-Guard „DEFPROT“ übernommen.
- (6) Dass die Werte aus dem Attribut-Guard „DEFPROT“ stammen, zeigt sich am Attribut USER-ACCESS: Der Wert „ALL-USERS“ ist kein System-Standardwert.
- (7) Die Jobvariable „NO.DEFPROT“ wird angelegt. Ihr Name entspricht *nicht* der Regel „1STCHAREQUJ“ und somit werden als Default-Werte die System-Standardwerte verwendet.
- (8) Dass die Default-Werte System-Standardwerte sind, zeigt sich am Attribut USER-ACCESS: Der Wert „OWNER-ONLY“ ist ein System-Standardwert.
- (9) Die Jobvariable „JV.STD-PROT“ wird angelegt. Ihr Name entspricht zwar der Regel „1STCHAREQUJ“, aber durch die Angabe „PROTECTION-ATTR=*STD“ wird die Verwendung der System-Standardwerte erzwungen.
- (10) Dass die Default-Werte System-Standardwerte sind, zeigt sich am Attribut USER-ACCESS: Der Wert „OWNER-ONLY“ ist ein System-Standardwert.

Default-Protection und Monitor-Jobvariablen

Die Default-Protection gilt auch für Monitor-Jobvariablen. Wird durch Default-Protection ein Schutzmerkmal gesetzt, das ein weiteres Zugreifen (z.B. durch Systemkommandos) verhindert, muss die Default-Protection-Einstellung vom Anwender geändert werden. Im MONJV-Handler wird z.B. eine neu anzulegende Monitor-Jobvariable nur dann auf USER-ACCESS=*ALL-USERS gesetzt, wenn die Default-Protection nicht explizit USER-ACCESS=*OWNER-ONLY vorsieht.

2.3.4 Schutz bei der Auftragsüberwachung und -steuerung

Auftragsüberwachung

Jobvariablen, die einen Auftrag überwachen, sind gegen Löschen und gegen die Änderung von Schutzmerkmalen geschützt.

Dieser Schutz besteht ab Erklärung zur überwachenden Jobvariablen bis zur Beendigung der Überwachungsfunktion.

Für einen Einzelauftrag endet die Überwachungsfunktion mit der Beendigung des Auftrags. Für den Folge-Job eines Repeatjobs beginnt die Überwachungsfunktion jeweils wieder mit dem LOGON des Folgeauftrags und endet mit der Auftragsbeendigung.

Bei einem Kalender-Job endet die Überwachungsfunktion erst, wenn der Kalender-Job insgesamt beendet ist.

Ein privilegierter Benutzer (Privileg TSOS) kann darüber hinaus die Überwachung eines Auftrags mit dem Kommando MODIFY-JOB-OPTIONS vorzeitig beenden oder auf eine andere Jobvariable übertragen. In diesem Fall endet der Schutz mit dem Verlust der Überwachungsfunktion.

Der Schutz wird in einer gesonderten Ausgabezeile mit dem Kommando SHOW-JV-ATTRIBUTES angezeigt. Mit der Angabe SELECT=*BY-ATTRIBUTES(MONJV=*YES) werden Jobvariablen angezeigt, für die dieser Schutz besteht. Der Schutz kann mit MODIFY-JV-ATTRIBUTES im Operanden MONJV-PROTECTION explizit aufgehoben werden. Wird der Schutz aufgehoben, kann die auftragsüberwachende Jobvariable nicht mehr mit aktuellen Werten versorgt werden. Einige Komponenten (z.B. JMS) setzen deshalb den Schutz bei jedem Zugriff neu.

Bedingungsabhängige Auftragssteuerung

Jobvariablen, die in bedingten Kommandos oder Makros bei bedingungsabhängiger Auftragssteuerung verwendet werden, sind gegen Löschen und gegen Änderung der Schutzmerkmale geschützt.

Der Schutz wird in der Ausgabe des Kommandos SHOW-JV-ATTRIBUTES nicht gesondert angezeigt. Aber mit der Angabe SELECT=*BY-ATTRIBUTES(CJC-PROTECTION=*YES) lässt sich die Ausgabe so reduzieren, dass nur Jobvariablen angezeigt werden, für die dieser Schutz besteht.

Der Schutz dauert, solange eine Jobvariable in einer Bedingung verwendet wird (z.B. von der Aktivierung bis zur Beendigung bzw. Löschen der CJC-Kommandofolge oder bis die Wartezeit eines WAIT-Kommandos beendet ist).

2.4 Zugriffsverlust auf eine Jobvariable

Die folgende Übersicht beschreibt die Reaktion des Systems, falls der Zugriff auf eine verwendete Jobvariable nicht möglich ist.

Situation		
Die Jobvariable befindet sich im Katalog eines anderen Rechners, zu dem die Verbindung abgebrochen ist. Der Katalog wurde im MRSCAT des eigenen Rechners eingetragen mit ADD-MASTER-CATALOG-ENTRY bzw. mit MODIFY-MASTER-CATALOG-ENTRY und folgenden Operanden:	Die Jobvariable befindet sich in einem Katalog, der mit EXPORT-PUBSET exportiert wurde. Folgende Operanden wurden angegeben:	
DIALOG-WAIT= wert1 BATCH-WAIT= wert2	TERMINATE-JOBS= NO	TERMINATE-JOBS = YES bzw. zusätzlich ein FORCE-PUBSET-EXPORT
Reaktion		
Das Kommando (oder auch Makroaufruf), das auf den Katalog zugreifen will, wartet die eingestellte Zeit, und danach wird die Ausführung mit Fehler (Spin-Off) abgebrochen. Eine Wartezeit mit dem Wert Null bewirkt den sofortigen Abbruch. Ohne Fehlerbehandlung führt dies zur Beendigung der Prozedur bzw. des Batch-Auftrags. Wenn eine Fehlerbehandlung stattfindet, führt der nächste Katalogzugriff erneut zu der vereinbarten Wartezeit.	ein laufender Auftrag bzw. Programm kann den Zugriff beenden	ein laufender Auftrag wird sofort abgebrochen
Standard: wert1=30 Sekunden wert2=28800 Sekunden	Der Katalog ist für noch startende Aufträge nicht verfügbar (Abbruch)	

Tabelle 11: Reaktion beim Zugriffsverlust auf eine Jobvariable

Weitere Informationen über die Verwendung von Jobvariablen in einem Mehrrechnerverbund mit den möglichen Situationen für einen Zugriffsverlust finden Sie im Handbuch „HIPLEX MSCF“ [8].

2.5 Die Werte der Benutzer-Jobvariablen

Der Benutzer kann einer Jobvariablen jegliche Zeichenkombination, im Zeichenformat oder in sedezimaler Form, als Wert zuweisen. Diese Information kann bis zu 256 Zeichen (Byte) lang sein. Die Jobvariable kann sowohl in ihrer gesamten Länge angesprochen werden, als auch mittels Teilbereichsspezifikation, wenn nur ein bestimmter Teil der Jobvariablen gebraucht wird. Ein Teilbereich der Jobvariable wird durch Angabe der Position des ersten Zeichens und der Länge des Teilbereichs bestimmt.

In den Kommandos werden die Jobvariablen-Werte im gleichen Format angegeben wie Zeichen- und Sedezimalkonstanten in der Assembler-Sprache, z.B. 'ABC' oder 'X'1C2C3'.

Sedezimalkonstanten ungerader Länge werden links mit einer binären Null ergänzt.

Hinweis

Bei Eingabe von Zeichenkonstanten wird Groß- und Kleinschreibung berücksichtigt.

Dies ist besonders zu beachten bei Verwendung von bedingten Ausdrücken in der Auftragssteuerung (siehe [Seite 83](#)) und bei der Eingabe von Kommandoteilen aus Jobvariablen (siehe [Seite 70](#)).

2.6 Sicherung von Benutzer-Jobvariablen

Die Datensicherungsfunktionen in BS2000/OSD-BC sowie in den Softwareprodukten ARCHIVE und HSMS unterstützen auch die Datensicherung von Jobvariablen. Bei Verlust von Jobvariablen (z.B. versehentliches Löschen oder Überschreiben) können die betreffenden Jobvariablen anhand einer Sicherungskopie wieder rekonstruiert werden.

Datensicherung mit Snapsets

Ein Snapset ist die Sicherungskopie eines SF- bzw. SM-Pubsets auf Basis von Snap-Units. Dabei sichert jede Snap-Unit des Snapsets die jeweils zugeordnete Original-Platte des Pubsets. Snapsets dienen als Pubset-Sicherungen gegen Ausfall von Dateien und Jobvariablen (Backup) und werden im laufenden Pubset-Betrieb von der Systembetreuung oder vom HSMS-Administrator erzeugt und später wieder gelöscht. Die Snapsets können als logische Sicherung aller Dateien und Jobvariablen eines Pubsets benutzt werden. Der neueste Snapset kann auch wie eine physikalische Sicherung zur Pubset-Restaurierung auf Plattenbasis verwendet werden.

Detaillierte Information zur Datensicherung mit Snapsets entnehmen Sie dem Handbuch „Einführung in die Systembetreuung“ [3].

Bei Verlust von Jobvariablen (z.B. versehentliches Löschen oder Überschreiben) kann der Benutzer die betreffenden Jobvariablen anhand eines Snapsets wieder rekonstruieren:

- Das Kommando LIST-JV-FROM-SNAPSET informiert über Jobvariablen, die auf einem Snapset gesichert wurden (wie SHOW-JV-ATTRIBUTES, das Informationen über Jobvariablen aus dem aktuellen Dateikatalog liefert).
Auf Programmebene ist die Abfrage mit dem Makroaufruf LJFSNAP möglich.
- Das Kommando RESTORE-JV-FROM-SNAPSET restauriert Jobvariablen auf dem Stand eines bestimmten Snapsets oder auf dem aktuellsten Stand auf Basis aller vorhandenen Snapsets. Das Umbenennen erfolgt hier über den Operanden NEW-JV-NAME. Die weiteren Optionen zum Überschreiben (Operand REPLACE) und Steuern der Protokollausgabe (Operanden OUTPUT, REPORTING) sind wie beim Restaurieren von Dateien vorhanden.
Auf Programmebene ist die Rekonstruktion mit dem Makroaufruf RJFSNAP möglich.

Eine genaue Beschreibung der Kommandos bzw. der Makroaufrufe finden Sie im Handbühren „Kommandos“ [1] bzw. „DVS Makros“ [5].

Datensicherung mit ARCHIVE und HSMS

In den automatischen Sicherungsläufen eines Data Center mit dem Dienstprogramm ARCHIVE bzw. HSMS kann die Systembetreuung oder der HSMS-Administrator neben Dateien optional auch von Benutzer-Jobvariablen sichern.

Der Benutzer kann auch selbst die Sicherung seiner Jobvariablen mit ARCHIVE oder HSMS vornehmen.

Bei Verlust von Jobvariablen (z.B. versehentliches Löschen oder Überschreiben) kann der Benutzer die betreffenden Jobvariablen anhand einer Sicherungskopie wieder rekonstruieren.

Eine genaue Beschreibung der nötigen Anweisungen finden Sie in den Handbüchern „ARCHIVE“ [11] bzw. „HSMS“ [12].

2.7 Sonder-Jobvariablen

Sonder-Jobvariablen sind Systeminformationen, die der Benutzer wie Jobvariablen lesen kann. Sonder-Jobvariablen sind über die Pseudo-Benutzer-Kennung SYSJV anzusprechen. Eine echte Benutzer-Kennung SYSJV darf daher im System nicht existieren.

Für Sonder-Jobvariablen existiert kein Katalogeintrag. Das Pfadnamenschema ist deshalb nicht anwendbar. Eine Katalogkennung darf nicht angegeben werden. Der Name einer Sonder-Jobvariablen wird in der Form „\$SYSJV.<jvname>“ angegeben.

Im Kommando SHOW-JV-ATTRIBUTES kann die Angabe „JV-NAME=\$SYSJV.<jvname>“ auch Musterzeichen (Wildcards) enthalten. Mit der Angabe „\$SYSJV.“ werden die Namen aller im System verfügbaren Sonder-Jobvariablen ausgegeben.

Mit INFORMATION=*ALL-ATTRIBUTES wird neben dem Namen der Sonder-Jobvariable in einer zusätzliche Ausgabezeile Format und Inhalt der Sonderjobvariable beschrieben. Der Beschreibungstext erscheint in der aktuell eingestellten Task-Sprache: Englisch bzw. Deutsch (kann task-spezifisch mit dem Kommando /MODIFY-MSG-ATTRIBUTES eingestellt werden, siehe Handbuch „Kommandos“ [1]).

Sonder-Jobvariablen können nur gelesen werden. Die Verwendung ist nur in den folgenden Kommandos möglich:

- MODIFY-JV im Operanden SET-VALUE (Wertübernahme)
- MODIFY-JV-CONDITIONALLY im Operanden SET-VALUE (Wertübernahme)
- SHOW-JV im Operanden JV-CONTENTS (Ausgabe des Wertes)
- SKIP-COMMANDS im Operanden CONDITION (im bedingten Ausdruck)

Daneben können Sonder-Jobvariablen auch zur Eingabe von Kommandoteilen aus Jobvariablen verwendet werden (siehe [Abschnitt „Eingabe aus Jobvariablen“ auf Seite 70](#)).

Der Zugriff auf eine Sonder-Jobvariable ist nur möglich, falls die gewünschte Information auch im System existiert. Es kann z.B. kein Programmname ausgegeben werden, wenn der abfragende Auftrag kein Programm geladen hat (\$SYSJV.PROGNAME). Der Benutzer wird in diesem Fall mit der Meldung JVS04B2 darauf hingewiesen, dass die gewünschte Sonder-Jobvariable „leer“ ist (wenn im nachfolgenden Ausgabeformat null Zeichen möglich sind). Die folgenden beiden Tabellen listen alle Sonderjobvariablen auf, wobei die erste Tabelle alphabetisch und die zweite (ab [Seite 49](#)) nach Kategorien sortiert ist. Die zweite Tabelle enthält keine Beschreibung der Sonderjobvariablen.

Sonderjobvariablen alphabetisch sortiert

Name	Ausgabeformat	Inhalt
\$\$SYSJV.ACCOUNT	1-8 Zeichen	Abrechnungsnummer der laufenden Task
\$\$SYSJV.COUNTER	4 Zeichen, 0001-9999	taskspezifischer Zähler ¹
\$\$SYSJV.DATE	jj-mm-ttiii	aktuelles Datum (ISO-Format) und dem Tag (iii) im laufenden Jahr
\$\$SYSJV.DATE4	jjjjj-mm-ttiii	aktuelles Datum (ISO4-Format) und dem Tag (iii) im laufenden Jahr
\$\$SYSJV.DATE-ISO4	jjjj-mm-tt	aktuelles Datum (ISO4-Format)
\$\$SYSJV.DATE-TIME-LONG	jjjj-mm-tt hh:mm:ss	aktuelles Datum und Uhrzeit (Format für Textausgaben)
\$\$SYSJV.DATE-TIME-SHORT	jjjj-mm-tt.hhmmss	aktuelles Datum und Uhrzeit (Format für Dateinamen)
\$\$SYSJV.DATUM	tt.mm.jjjj	aktuelles Datum (deutsches Format)
\$\$SYSJV.DAY	MON, TUE, WED, THU, FRI, SAT, SUN	aktueller Wochentag, englisch
\$\$SYSJV.HOME-CATID	1-4 Zeichen	Katalogkennung des Home-Pubsets
\$\$SYSJV.HOST	1-8 Zeichen	BCAM-Name des lokalen Rechners; z.B. D046ZE08
\$\$SYSJV.JOB-CLASS	1-8 Zeichen	Jobklasse der laufenden Task
\$\$SYSJV.JOB-ELAPSED-TIME	-ddddddddd-hh:mm:ss	Zeit, die seit dem Zeitpunkt des Jobstarts (siehe \$\$SYSJV.JOB-LOGON) vergangen ist, wobei d die Anzahl der Tage bezeichnet; die Sekundenangabe ist jedoch nicht genau, da der Jobstart nur minutengenau vorliegt: z.B. ergibt sich bei einer Startzeit von 12:31:34 nach 20 Sekunden der Wert -0000000000-00:00:54
\$\$SYSJV.JOB-LOGON	jjjjmmtt.hhmm	Zeitpunkt des Jobstarts z.B. 20010112.0901
\$\$SYSJV.JOB-LOGON-ISO	jjjj-mm-tt hh:mm	Jobstart-Zeitpunkt (ISO-Format) z.B. 2000-01-12 09:01
\$\$SYSJV.JOB-MONJV	0-54 Zeichen	Name der auftragsüberwachenden MONJV
\$\$SYSJV.JOBNAME	0-8 Zeichen	Auftragsname der laufenden Task

Tabelle 12: Sonder-Jobvariablen alphabetisch sortiert (Teil 1 von 4)

Name	Ausgabeformat	Inhalt
\$\$SYSJV.JOB-SOURCE	1-54 Zeichen	Pfadname der Original-Quelldatei (ENTER-bzw. Prozedurdatei, die auch im Feld <i>ORIGFILE</i> bei /SHOW-JOB-STATUS angezeigt wird)
\$\$SYSJV.JOB-SPOOLIN	jjjjmmtt.hhmm	Zeitpunkt des Job-Spoolin z.B. 20012212.0901
\$\$SYSJV.JOB-SPOOLIN-ISO	jjjj-mm-tt hh:mm	Zeitpunkt des Job-Spoolin (ISO-Format) z.B. 2000-01-12 09:01
\$\$SYSJV.MONAT	JAN, FEB, MRZ, APR, MAI, JUN, JUL, AUG, SEP, OKT, NOV, DEZ	aktueller Monat, deutsch
\$\$SYSJV.MONTH	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC	aktueller Monat, englisch
\$\$SYSJV.PERS-ID	0-8 Zeichen	Personal-Id bei personenbezogenem Logon
\$\$SYSJV.PRINCIPAL	1-256 Zeichen	Principal-Name der Kerberos-Identifikation beim Dialog-Logon
\$\$SYSJV.PRIO	1-3 Ziffern	Priorität der laufenden Task
\$\$SYSJV.PROCESSOR	0-8 Zeichen	Prozessorname der TIAM-Station; z.B. D046KR11
\$\$SYSJV.PROCESSOR-APPL	0-8 Zeichen	Prozessorname der TIAM-Station in einer Anwendung (z.B. OMNIS); z.B. D016ZE04
\$\$SYSJV.PROC-LEVEL	1-3 Zeichen	Prozedurlevel
\$\$SYSJV.PROC-SOURCE	1-54 Zeichen	Pfadname der Kopie der Quelldatei (kopierte ENTER-bzw. Prozedurdatei, die auch im Feld <i>CMDFILE</i> bei /SHOW-JOB-STATUS angezeigt wird)
\$\$SYSJV.PROG-MONJV	0-54 Zeichen	Name der programmüberwachenden MONJV
\$\$SYSJV.PROGNAME	0-41 Zeichen	Name des geladenen Programms, wenn dieses aus einer LMS-Bibliothek oder als *-Datei geladen wurde

Tabelle 12: Sonder-Jobvariablen alphabetisch sortiert (Teil 2 von 4)

Name	Ausgabeformat	Inhalt
\$SYSJV.REMAINING-BS2000-RUNTIME	5 Zeichen: UNDEF, UNLIM oder hh:mm	Restlaufzeit, die dem BS2000-System zum Herunterfahren zur Verfügung steht, bevor die reale bzw. virtuelle Maschine angehalten wird; unterstützt auf SQ-Servern, auf anderen BS2000-Servern ist der Wert immer UNDEF.
\$SYSJV.SNO	3 Zeichen	Systemlaufnummer
\$SYSJV.STARTUPTYPE	1 Zeichen	Startup-Typ z.B. Z, F, C, W, S
\$SYSJV.STATION	0-8 Zeichen	Stationsname der TIAM-Station; z.B. DSB11243
\$SYSJV.STATION-APPL	0-8 Zeichen	Stationsname der TIAM-Station in einer Anwendung (z.B. OMNIS); z.B. BT200175
\$SYSJV.STATIONTYPE	0-8 Zeichen	Gerätetyp der TIAM-Station z.B. DSS-9750
\$SYSJV.STD-CATID	1-4 Zeichen	Standard-Katalogkennung (Default-Pubset) des Anwenders
\$SYSJV.SYSCMD	1-54 Zeichen	Name der zugeordneten Datei oder logische Systemdatei in Klammern
\$SYSJV.SYSDTA	1-54 Zeichen	Name der zugeordneten Datei oder logische Systemdatei in Klammern
\$SYSJV.SYSID	1-3 Zeichen	Systemkennzeichen ²
\$SYSJV.SYSLST	1-54 Zeichen	Name der zugeordneten Datei oder logische Systemdatei in Klammern
\$SYSJV.SYSOUT	1-54 Zeichen	Name der zugeordneten Datei oder logische Systemdatei in Klammern
\$SYSJV.SYSTEM-VERSION	10 Zeichen	Systemversionsnummer z.B. V17.0A00pp mit pp=PVLU
\$SYSJV.TAG	MO, DI, MI, DO, FR, SA, SO	aktueller Wochentag, deutsch
\$SYSJV.TASK-CPU-USED	ssssss.ssss	seit LOGON verbrauchte CPU-Zeit der Task
\$SYSJV.TASK-CPU-USED-LONG	ssssssssss.ssss	seit LOGON verbrauchte CPU-Zeit der Task für sogenannte Langläufer
\$SYSJV.TASK-MODE	2-6 Zeichen: BATCH, DIALOG, SYSTEM, TP	Modus der laufenden Task

Tabelle 12: Sonder-Jobvariablen alphabetisch sortiert (Teil 3 von 4)

Name	Ausgabeformat	Inhalt
\$SYSJV.TEMP-PREFIX	1 Zeichen	Präfix für temporäre Jobvariablen und Dateien, z.B. #
\$SYSJV.TIME	hh:mm:ss	aktuelle Uhrzeit
\$SYSJV.TIME-ZONE	+hh:mm i	Zeitdifferenz der lokalen Zeitzone gegenüber der UTC-Zeit, wobei der Indikator i die aktuelle Zeitzählung anzeigt: – W (Winterzeit≙Normalzeit) – S (Sommerzeit)
\$SYSJV.TIMESTAMP	jjjj-mm-tthhmmss	Zeitstempel mit Datum und Uhrzeit (UTC) im Format wie in einer MONJV
\$SYSJV.TOMORROW	jjjj-mm-tt	Datum des Folgetags
\$SYSJV.TSN	4 Zeichen	TSN der laufenden Task
\$SYSJV.USERID	1-8 Zeichen	Benutzerkennung der laufenden Task
\$SYSJV.VIRTUAL-HOST	0-8 Zeichen	BCAM-Name des virtuellen Rechners
\$SYSJV.YESTERDAY	jjjj-mm-tt	Datum des Vortags
\$SYSJV.ZEIT	hh-mm-ss	aktuelle Uhrzeit

Tabelle 12: Sonder-Jobvariablen alphabetisch sortiert (Teil 4 von 4)

- ¹ Wird mit Null initialisiert und vor jedem taskeigenen Lesezugriff um eins erhöht, enthält also nur innerhalb einer Task eindeutige Werte, z.B. für Labelgenerierung.
- ² Für einen Home-Pubset vergebenes Kennzeichen.

Sonderjobvariablen nach Kategorie

Kategorie	Name	Ausgabeformat
Datum/Zeit	\$\$SYSJV.DATE	jj-mm-ttiii
	\$\$SYSJV.DATE4	jjjj-mm-ttiii
	\$\$SYSJV.DATE-ISO4	jjj-mm-tt
	\$\$SYSJV.DATE-TIME-LONG	jjj-mm-tt hh:mm:ss
	\$\$SYSJV.DATE-TIME-SHORT	jjj-mm-tt.hhmmss
	\$\$SYSJV.DATUM	tt.mm.jjjj
	\$\$SYSJV.DAY	MON, TUE, WED, THU, FRI, SAT, SUN
	\$\$SYSJV.MONAT	JAN, FEB, MRZ, APR, MAI, JUN, JUL, AUG, SEP, OKT, NOV, DEZ
	\$\$SYSJV.MONTH	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
	\$\$SYSJV.TAG	MO, DI, MI, DO, FR, SA, SO
	\$\$SYSJV.TIME	hh:mm:ss
	\$\$SYSJV.TIME-ZONE	+hh:mm i
	\$\$SYSJV.TIMESTAMP	jjjj-mm-tthhmmss
	\$\$SYSJV.TOMORROW	jjjj-mm-tt
	\$\$SYSJV.YESTERDAY	jjjj-mm-tt
	\$\$SYSJV.ZEIT	hh-mm-ss
Job-/Auftragsinformation	\$\$SYSJV.ACCOUNT	1-8 Zeichen
	\$\$SYSJV.COUNTER	4 Zeichen, 0001-9999
	\$\$SYSJV.JOB-CLASS	1-8 Zeichen
	\$\$SYSJV.JOB-ELAPSED-TIME	20 Zeichen
	\$\$SYSJV.JOB-LOGON	13 Zeichen
	\$\$SYSJV.JOB-LOGON-ISO	16 Zeichen
	\$\$SYSJV.JOBNAME	0-8 Zeichen
	\$\$SYSJV.JOB-SOURCE	1-54 Zeichen
	\$\$SYSJV.JOB-SPOOLIN	13 Zeichen
	\$\$SYSJV.JOB-SPOOLIN-ISO	16 Zeichen
	\$\$SYSJV.PERS-ID	0-8 Zeichen

Tabelle 13: Sonder-Jobvariablen nach Kategorien sortiert (Teil 1 von 2)

Kategorie	Name	Ausgabeformat
Job-/Auftragsinformation (Forts.)	\$\$SYSJV.PRINCIPAL	1-256 Zeichen
	\$\$SYSJV.PRIO	1-3 Ziffern
	\$\$SYSJV.PROC-SOURCE	1-54 Zeichen
	\$\$SYSJV.PROGNAME	0-41 Zeichen
	\$\$SYSJV.STD-CATID	1-4 Zeichen
	\$\$SYSJV.TASK-CPU-USED	11 Zeichen, ssssss.ssss
	\$\$SYSJV.TASK-CPU-USED-LONG	15 Zeichen, ssssssssss.ssss
	\$\$SYSJV.TASK-MODE	2-6 Zeichen: BATCH, DIALOG, SYSTEM, TP
	\$\$SYSJV.TSN	4 Zeichen
	\$\$SYSJV.USERID	1-8 Zeichen
Information über Jobvariablen	\$\$SYSJV.JOB-MONJV	0-54 Zeichen
	\$\$SYSJV.PROG-MONJV	0-54 Zeichen
	\$\$SYSJV.TEMP-PREFIX	1 Zeichen
Systemdateien und Prozeduren	\$\$SYSJV.PROC-LEVEL	1-3 Zeichen
	\$\$SYSJV.SYSCMD	1-54 Zeichen
	\$\$SYSJV.SYSDTA	1-54 Zeichen
	\$\$SYSJV.SYSLST	1-54 Zeichen
	\$\$SYSJV.SYSOUT	1-54 Zeichen
TIAM-Information	\$\$SYSJV.HOST	1-8 Zeichen
	\$\$SYSJV.PROCESSOR	0-8 Zeichen
	\$\$SYSJV.PROCESSOR-APPL	0-8 Zeichen
	\$\$SYSJV.STATION	0-8 Zeichen
	\$\$SYSJV.STATION-APPL	0-8 Zeichen
	\$\$SYSJV.STATIONTYPE	0-8 Zeichen
	\$\$SYSJV.VIRTUAL-HOST	0-8 Zeichen
Systemdaten	\$\$SYSJV.HOME-CATID	1-4 Zeichen
	\$\$SYSJV.REMAINING-BS2000-RUNTIME	5 Zeichen: UNDEF, UNLIM oder hh:mm
	\$\$SYSJV.SNO	3 Zeichen
	\$\$SYSJV.STARTUPTYPE	1 Zeichen
	\$\$SYSJV.SYSID	1-3 Zeichen
	\$\$SYSJV.SYSTEM-VERSION	10 Zeichen

Tabelle 13: Sonder-Jobvariablen nach Kategorien sortiert (Teil 2 von 2)

Exit 033 für Sonder-Jobvariablen

Die Exit-Routine wird aufgerufen, wenn eine vom System nicht angebotene Sonder-Jobvariable angesprochen wird. Der Zugriff kann durch die Kommandos SHOW-JV und MODIFY-JV oder durch die Makros GETJV und SETJV erfolgen. In der Exit-Routine können Werte für die Sonder-Jobvariable übergeben werden, die der Anwender des Kommandos erhalten soll. Auf diese Weise können eigene Sonder-Jobvariablen realisiert werden.

Die Adresse des Parameterbereichs EX033 wird in Register 1 übergeben. Das Feld JVSXJV des Parameterbereichs enthält den Namen der vom Benutzer gewünschten Sonder-Jobvariable.

Folgende Informationen werden an die Exit-Routine übergeben:

```
R1    = A(EX033-Parameterliste)
R12   = A(TPR Program Manager)
R13   = A(Sicherungsbereich der rufenden Komponente)
R14   = A(indirekter Rücksprung)
R15   = A(Exit-Routine)
```

Rückkehrinformationen

Die Rückinformation an die rufende Systemkomponente kann in der Parameterliste übergeben werden. Im Feld JVSXLEN kann die Länge der Information (1-256) und im Feld JVSXVAL der Inhalt der Sonder-Jobvariablen eingetragen werden. Der Maincode im Standardheader muss auf 'X'0000' gesetzt sein.

Ist keine Exit-Routine angemeldet oder der Maincode ungleich 'X'0000', erhält der Benutzer die Meldung:

```
% JVS0472 NAME DER SONDER-JV UNZULAESSIG. KOMMANDO KORRIGIEREN
```

DSECT

Eine DSECT für die Parameterliste (Adresse in Register 1) kann mit dem Makro EX033 erzeugt werden.

```
EX033    EX033 MF=D
EX033    MFTST MF=D,PREFIX=J,MACID=VSX,ALIGN=F,                                C
          DMACID=VSX,SUPPORT=(D,C,M,L),DNAME=VSXPL
EX033    DSECT ,
          *,##### PREFIX=J, MACID=VSX #####
JVSXMIN  EQU    1                                MIN. LENGTH
*
JVSXMAX  EQU    256                              MAX. LENGTH
*
*    end parameterarea
```

```

JVSXHDR  FHDR  MF=(C,JVSX),EQUATES=NO                STANDARDHEADER
JVSXHDR  DS    0A
JVSXFHE  DS    0XL8                0  GENERAL PARAMETER AREA HEADER
*
JVSXIFID DS    0A                0  INTERFACE IDENTIFIER
JVSXFCTU DS    AL2                0  FUNCTION UNIT NUMBER
*
*                                BIT 15  HEADER FLAG BIT,
*                                MUST BE RESET UNTIL FURTHER NOTICE
*                                BIT 14-12 UNUSED, MUST BE RESET
*                                BIT 11-0  REAL FUNCTION UNIT NUMBER
JVSXFCT  DS    AL1                2  FUNCTION NUMBER
JVSXFCTV DS    AL1                3  FUNCTION INTERFACE VERSION NUMBER
*
JVSXRET  DS    0A                4  GENERAL RETURN CODE
JVSXSRET DS    0AL2               4  SUB RETURN CODE
JVSXSR2  DS    AL1                4  SUB RETURN CODE 2
JVSXSR1  DS    AL1                5  SUB RETURN CODE 1
JVSXMRET DS    0AL2               6  MAIN RETURN CODE
JVSXMR2  DS    AL1                6  MAIN RETURN CODE 2
JVSXMR1  DS    AL1                7  MAIN RETURN CODE 1
JVSXFHL  EQU    8                8  GENERAL OPERAND LIST HEADER LENGTH
*
*   main return codes
JVSXSUCC EQU    0                NO ERROR DETECTED
JVSXREJE EQU    1                INVALID SPECIAL JV
*
JVSXJV   DS    CL54                NAME OF THE SPECIAL JV
JVSXLEN  DS    H                    RETURNED LENGTH OF JV-VALUE
JVSXUNU  DS    XL2                  UNUSED
JVSXVAL  DS    CL256               RETURNED VALUE OF THE SPECIAL
*                                JV
JVSXUNU1 DS    XL2                  UNUSED
JVSX#    EQU    *-JVSXHDR

```

2.8 Auftrags-/Programmüberwachung mit Jobvariablen

Das Jobvariablen-Konzept dient primär dem Austausch von Informationen zwischen Aufträgen (Jobs) des Benutzers. Mit ihrer Hilfe können so genannte Jobketten oder Jobnetze aufgebaut werden. Es können beliebige Abhängigkeiten zwischen Aufträgen auf Jobvariablen abgebildet werden, sodass Auftragsstart, Überwachung des Verarbeitungsfortschritts (Jobtrace), Jobrestart usw. praktisch automatisch erfolgen können.

Über die Makroschnittstelle können auf Programmebene Jobvariablen erzeugt, geändert und abgefragt werden. Diese Informationen können ebenfalls zur Auftragssteuerung verwendet werden.

Eine zusätzliche Anwendungsmöglichkeit der Jobvariablen innerhalb des BS2000 ist die Überwachung des Auftrags-/Programmablaufs im System. Dazu muss der Benutzer im Operanden MONJV in den folgenden Kommandos eine Jobvariable zur überwachenden Jobvariable erklären (Weitere Informationen, siehe [Seite 64](#)).

Operand MONJV in Kommandos der Auftragsüberwachung:

- ENTER-JOB
- ENTER-PROCEDURE
- PRINT-DOCUMENT
- SET-LOGON-PARAMETERS
- TRANSFER-FILE (siehe „openFT“ [15])

Operand MONJV in Kommandos der Programmüberwachung:

- LOAD-EXECUTABLE-PROGRAM (bzw. LOAD-PROGRAM)
- RESTART-PROGRAM
- START-EXECUTABLE-PROGRAM (bzw. START-PROGRAM)

Der **privilegierte** Benutzer kann in den folgenden Kommandos eine überwachende Jobvariable definieren.

- CLEAR-VOLUME (siehe „SPACEOPT“ [16])
- CREATE-VM (siehe „VM2000“ [17])
- EXPORT-PUBSET
- EXTEND-VM-MEMORY (siehe „VM2000“ [17])
- FORCE-PUBSET-EXPORT
- IMPORT-PUBSET
- MODIFY-JOB-OPTIONS
- REDUCE-VM-MEMORY (siehe „VM2000“ [17])
- START-SPACEOPT-JOB (siehe „SPACEOPT“ [16])
- START-SUBSYSTEM

Das Betriebssystem weist dann dieser Jobvariablen zu bestimmten Zeitpunkten feste, vordefinierte Werte zu, die vom Benutzer innerhalb einer Kommandofolge abgefragt werden können, um die Verarbeitung des Auftrags/Programms zu verfolgen und gegebenenfalls zu beeinflussen.

In den nachfolgenden Kommandos kann eine überwachende Jobvariable zur Identifikation eines Auftrags verwendet werden.

- CANCEL-FILE-TRANSFER (siehe „openFT“ [15])
- CANCEL-JOB
- CANCEL-PRINT-JOB
- CHANGE-TASK-PRIORITY
- CHANGE-TASK-CPU-LIMIT
- FORCE-JOB-CANCEL
- HOLD-JOB
- HOLD-TASK
- MODIFY-JOB
- MODIFY-JOB-OPTIONS
- MODIFY-PRINT-JOB-ATTRIBUTES
- MOVE-TASK-TO-CATEGORY
- RESUME-JOB
- RESUME-PRINT-JOB
- RESUME-TASK
- SHOW-JOB-STATUS
- SHOW-FILE-TRANSFER (siehe „openFT“ [15])
- SHOW-PRINT-JOB-ATTRIBUTES
- SHOW-RESOURCE-ALLOCATION

Bis zur Beendigung des zu überwachenden Auftrags enthält der Jobvariablen-Eintrag den Vermerk, dass die Jobvariable eine Überwachungsfunktion hat. Der Jobvariablen-Eintrag kann nur dann geändert werden, wenn im Kommando MODIFY-JV-ATTRIBUTES im Operanden MONJV-PROTECTION der Schutz aufgehoben wird. Wird der Schutz aufgehoben, kann die auftragsüberwachende Jobvariable nicht mehr mit aktuellen Werten versorgt werden. Einige Komponenten (z.B. JMS) setzen deshalb den Schutz bei jedem Zugriff neu.

2.8.1 Werte für überwachende Jobvariablen

Eine überwachende Jobvariable besteht aus einem Systemteil (Byte 1-128) und einem Benutzerteil (Byte 129-256).

Während des Ablaufs eines Auftrags oder Programms werden in den Systemteil der überwachenden Jobvariablen zu bestimmten Zeitpunkten festgelegte Werte eingetragen.

Die durch den MONJV-Handler und die Systemkomponenten versorgten Felder (Systemteil) sind vom Auftragsstart bis zum Auftragsende gegen jeden anderen Schreibzugriff geschützt.

2.8.1.1 Auftragsüberwachung

Auftragsüberwachende Jobvariablen sollen unabhängig von der Schnittstelle, über die sie bedient werden (z.B. JMS, SPOOL, BCAM, DSSM), eine einheitliche Struktur und soweit möglich einheitliche Inhalte besitzen. Auftragsüberwachend im Sinne dieser Konvention sind alle überwachenden Jobvariablen, die nicht programmüberwachend sind.

Für den Systemteil wird folgende Struktur unterstützt:

Byte	Bedeutung/mögliche Werte	Eintrag durch
1-3	Zustand des Auftrags: – 1. Zeichen „\$“ – 2.-3. Zeichen: „A“ bis „Z“ Ab JV V13.0B ist neben den bisherigen Werten „\$R“, „\$T“, „\$A“, „\$S“ und „\$M“ z.B. auch „\$XY“ möglich.	Systemkomponente ¹
4	Enthält den Wert Null (0).	MONJV-Handler
5-8	TSN des Auftrags.	Systemkomponente ¹⁾
9-12	Katalogkennung des Home-Pubsets.	MONJV-Handler
13-16	Enthält Leerzeichen.	MONJV-Handler
17	Typ der MONJV: Zeichen „A“ bis „Z“ frei wählbar	Systemkomponente ¹⁾
18-20	Aktuelle Systemlaufnummer	MONJV-Handler
21-36	Auftragsbeginn (Öffnen der MONJV) in UTC-Zeit. Format: yyyy-mm-ddhhmmss	MONJV-Handler
37-52	Zeitstempel, um einen Aktivitäts-Zeitpunkt während eines Auftrages in UTC-Zeit protokollieren zu können. Format: yyyy-mm-ddhhmmss	Anwender ²
53-60	Name der zu überwachenden Anwendung.	Anwender ²⁾
61-70	Reserviert.	
71-128	Anwendungsspezifische Informationen. Die Bytes 105 bis 128 werden ggf. vom Auftragsverwaltungssystem AVAS genutzt.	Anwender ²⁾

Tabelle 14: Struktur einer auftragsüberwachenden Jobvariablen

¹ Systemkomponente, die die Überwachung durchführt (z.B. JMS, SPOOL, DMS).

² Der Anwender kann das Element über das Kommando MODIFY-MONJV bzw. den Makro TIMJV versorgen.

Über TSN und Katalogkennung kann bei Angabe des Namens der Jobvariablen der überwachte Auftrag identifiziert werden.

Beide Werte werden von den entsprechenden Systemkomponenten gesetzt, wenn der Auftrag eingeleitet wird, also entweder zum LOGON- oder zum ENTER-Zeitpunkt.

Die Werte für den Zustand des Auftrags und die Katalogkennung werden immer linksbündig angezeigt und mit Leerzeichen (X'40') ausgefüllt, während die TSN-Anzeige rechtsbündig mit führenden Nullen eingetragen wird. Innerhalb der ersten 128 Bytes (Systemteil) werden alle nicht belegten Bytes mit Leerzeichen (X'40') gefüllt.

Bei Spool- und Cancel-Aufträgen werden jeweils Teilbereiche aus den ersten 128 Bytes benötigt. Näheres über den Aufbau der Jobvariablen ist für Spool-Aufträge im Handbuch „SPOOL“ [13] bzw. für Cancel-Aufträge im Handbuch „Kommandos“ [1] beschrieben. Wurde der Auftrag abnormal beendet, wird die Zustandsanzeige einer überwachenden Jobvariablen auf den Wert \$A gesetzt. Zusätzlich werden die Informationen über den Urheber in verkürzter Form und der Kommentar aus dem Operanden TEXT des Kommandos CANCEL-JOB in den Systemteil (die ersten 128 Bytes) der überwachenden Jobvariable übernommen.

Wird der Auftrag für eine andere Benutzerkennung gestartet, muss die Jobvariable auch für diese Benutzerkennung zugreifbar sein. Ist dies nicht der Fall, so kann nach der Auftragsannahme keine Zustandsänderung protokolliert werden. Der Wert der Zustandsanzeige bleibt bei „\$S“, und die Jobvariable ist geschützt (MONJV-Schutz), obwohl der Auftrag beendet sein kann.

Eine aktive überwachende Jobvariable kann nicht gleichzeitig einem anderen Auftrag oder Programm als überwachende Jobvariable zugeordnet werden.

Geht der Zugriff auf die überwachende Jobvariable während der Auftragsbearbeitung verloren, hängt die Reaktion des Systems vom Zustand des Katalogs ab, der die Jobvariable enthält.

Aus Gründen der Ablaufsicherheit wird empfohlen, die überwachende Jobvariable auf dem gleichen Rechner einzurichten, auf dem der Auftrag bearbeitet wird.

Hinweis

Die Angabe einer temporären Jobvariablen, eines JV-Kettungsnamens oder eines JV-Teilbereichs als überwachende Jobvariable ist nicht erlaubt und wird abgewiesen.

Die Zustandswerte bei Dialogaufträgen:

Wert	Bedeutung/Anlass der Wertzuweisung
\$R	Der LOGON-Vorgang wurde erfolgreich durchgeführt und ein Auftrag eingeleitet.
\$T	Der Auftrag wurde normal beendet (EXIT-JOB MODE=NORMAL bzw. LOGOFF)
\$A	Der Auftrag wurde abnormal beendet (EXIT-JOB MODE=ABNORMAL, CANCEL-JOB oder Systembeendigung) oder die Auftragsüberwachung wurde vorzeitig beendet oder auf eine andere Jobvariable übertragen (MODIFY-JOB-OPTIONS).

Tabelle 15: Zustandswerte bei Dialogaufträgen

Die Zustandswerte bei Batchaufträgen:

Wert	Bedeutung/Anlass der Wertzuweisung
\$S	Der Auftrag wurde auf Grund eines ENTER-JOB-Kommandos nach erfolgreich abgeschlossenem „SPOOL-IN“ in die Auftragswarteschlange eingereicht. Bei dem Folgeauftrag eines Repeatjobs tritt dieser Zustand nicht auf, da dessen Überwachung erst mit dem LOGON beginnt.
\$R	Der Auftrag wurde aus der Auftragswarteschlange ausgewählt und gestartet.
\$M	Der Auftrag wurde mit dem Kommando MOVE-JOBS aus dem aktuellen Jobpool exportiert
\$T	Der Auftrag wurde normal beendet (EXIT-JOB MODE=NORMAL bzw. LOGOFF)
\$A	Der Auftrag wurde abnormal beendet (EXIT-JOB MODE=ABNORMAL, CANCEL-JOB oder Systembeendigung) oder die Auftragsüberwachung wurde vorzeitig beendet oder auf eine andere Jobvariable übertragen (MODIFY-JOB-OPTIONS).

Tabelle 16: Zustandswerte bei Batchaufträgen

Beispiele

Mit dem Kommando SHOW-JV wird der Inhalt der auftragsüberwachenden Jobvariable MON.JOB ausgegeben. Der Buchstabe **J** (Byte 17) bedeutet, dass mit der Jobvariablen MON.JOB ein **Job** (Auftrag) überwacht wird.

```
/SHOW-JV JV=MON.JOB
```

Ausgabe:	%	\$R	0	1ULW	10SB	J	068	2010-11-14161559	2010-11-14161807	Appl	Info		
ab Position		1	4	5	9	17	18	21		37		53	71

Mit dem Kommando SHOW-JV wird der Inhalt der auftragsüberwachenden Jobvariable MON.SPOOL ausgegeben. Der Buchstabe **S** (Byte 17) bedeutet, dass die Jobvariable MON.SPOOL einen **Spool**-Auftrag überwacht. Die Erklärung aller weiteren und zusätzlichen Feldinhalte finden Sie im Handbuch „SPOOL“ [13].

/SHOW-JV JV=MON.SPOOL

Ausgabe:	%	\$S	0	1ULW	10SB	S	068	2010-11-14161559	2010-11-14161807	Appl	Info
ab Position		1	4	5	9	17	18	21	37	53	71

2.8.1.2 Überwachung von IMPORT-/EXPORT-PUBSET

Die Funktionen „Importieren“ bzw. „Exportieren“ von gemeinschaftlichen Datenträgern (Pubsets) und Katalogen werden durch die Kommandos IMPORT- bzw. EXPORT-PUBSET vom **privilegierten** Benutzer (siehe [Seite 237](#)) eingeleitet. Hierbei wird ein eigener Auftrag für die gewünschte Verarbeitung erzeugt. Der Verarbeitungszustand kann durch Jobvariablen überwacht werden. Nur die Zustandsanzeige wird mit Werten versorgt. Informationen über die Auswirkungen in einem MSCF-Verbund finden Sie im Handbuch „HIPLEX MSCF“ [8].

Die Zustandswerte bei IMPORT-PUBSET:

Wert	Bedeutung/Anlass der Wertzuweisung
\$I	Importauftrag erfolgreich eingeleitet.
\$R	Pubset erfolgreich importiert.
\$A	Import fehlerhaft abgebrochen.
\$W	Bei Import eines Shared-Pubsets wartet der Rechner auf die Bestätigung des Master-Rechners.

Tabelle 17: Zustandswerte bei IMPORT-PUBSET

Die Zustandswerte bei EXPORT-PUBSET: Überwachung bei Shared-Pubset-Betrieb

Wert	Bedeutung/Anlass der Wertzuweisung
\$E	Exportauftrag erfolgreich gestartet.
\$T	Exportieren eines Pubsets erfolgreich abgeschlossen.
\$A	Exportieren fehlerhaft oder mit CANCEL-PUBSET-EXPORT abgebrochen.

Tabelle 18: Zustandswerte bei EXPORT-PUBSET

Bei Einsatz von HIPLEX MSCF ist der gleichzeitige und gemeinsame Zugriff über mehrere Rechner hinweg auf einen Pubset möglich. Einer der Verbundteilnehmer („Sharer“) wird zum Eigentümer-Rechner („Master“) dieses mehrbenutzbaren Pubsets ernannt, der die Verwaltungsaufgaben dieses Verbundes übernimmt.

Auf dem Home-Pubset jedes Sharers befindet sich für jeden Shared-Pubset eine shared-pubset-spezifische auftragsüberwachende Jobvariable, welche beim Importieren des Pubsets eingerichtet wird (wenn sie noch nicht vorhanden ist).

Diese überwachende Jobvariable kann folgende Zustandswerte annehmen:

Wert	Bedeutung/Anlass der Wertzuweisung
\$R	Shared-Pubset verfügbar. Master-Wechsel erfolgreich durchgeführt.
\$T	Shared-Pubset wird nicht mehr überwacht; er wird bzw. ist bereits exportiert.
\$A	Master-Wechsel fehlerhaft abgebrochen, z.B. bei Ausfall des Masters und einer nicht erfolgreichen Bestimmung eines neuen Masters aus der Reihe der Slave-Rechner.
\$C	Wegen Ausfall bzw. Abschalten des Masters wird ein Master-Wechsel auf dem gemeinschaftlichen Datenträger durchgeführt.

Tabelle 19: Zustandswerte bei Shared-Pubset-Betrieb

2.8.1.3 Programmüberwachung

Programmüberwachende Jobvariablen sind *nicht* gegen Überschreiben geschützt. Die ersten 128 Bytes sollten aber auch hier nicht für Benutzerzwecke verwendet werden, da ein Systemeintrag überschrieben werden kann bzw. bei einer Zustandsänderung dieser Teil vom System überschrieben wird.

In Jobvariablen zur Programmüberwachung setzt das Betriebssystem zwei Werte:

Zustandsanzeige Länge: 3 Zeichen
 Rückkehrcode-Anzeige Länge: 4 Zeichen

Die Zustandsanzeige kann die Werte „\$R“, „\$T“ oder „\$A“ annehmen.

Wert	Bedeutung/Anlass der Wertzuweisung
\$R	Das Programm wurde gestartet bzw. geladen oder wieder gestartet.
\$T	Das Programm wurde erfolgreich beendet (Makro TERM MODE=NORMAL)
\$A	Das Programm wurde vorzeitig durch einen Programmfehler oder durch einen definierten Fehlerausgang (Makro TERM MODE=ABNORMAL) beendet oder per Kommando CANCEL-PROGRAM abgebrochen.

Tabelle 20: Zustandswerte bei Programmüberwachung

Die Rückkehrcode-Anzeige dient dazu, einen vom Benutzer auf Programmebene definierten Rückkehrcode in die Ebene der Auftragssteuerung weiterzugeben. Die Anzeige kann im Programm durch den TERM-Makro gesetzt werden, wobei die Abspeicherung linksbündig erfolgt. Standardwert ist `UUUU` bei fehlerfreier Beendigung.

Beispiel

Mit dem Kommando wird der Inhalt der programmüberwachenden Jobvariablen MONITOR ausgegeben. Der Buchstabe P (Byte 17) bedeutet, dass die Jobvariable ein **P**rogramm überwacht.

```
/SHOW-JV JV=MONITOR
```

Ausgabe:	%	\$R		P	
ab Position		1		17	

2.8.2 Identifizierung von Aufträgen mit überwachenden Jobvariablen

Eine wichtige Funktion von überwachenden Jobvariablen ist die Identifizierung eines Auftrags in den Kommandos zur Auftragsverwaltung (Kommandoliste siehe [Seite 54](#)).

In diesen Kommandos können Aufträge sowohl über die TSN als auch über eine MONJV angesprochen werden.

Einige dieser Kommandos (SHOW-JOB-STATUS, CHANGE-TASK-PRIORITY, CANCEL-JOB und FORCE-JOB-CANCEL) können rechnerübergreifend abgesetzt werden. Die Definition einer überwachenden Jobvariablen in einem Rechnerverbund ist von Nutzen, da die TSN eines Auftrags in einem MSCF-Verbund normalerweise nicht eindeutig ist. Mithilfe einer netzweit eindeutig benannten überwachenden Jobvariablen kann jedoch ein Auftrag identifiziert werden, der irgendwo im jeweiligen MSCF-Verbund läuft (siehe auch Handbuch „HIPLEX MSCF“ [8]).

Die Eigenschaften der Bearbeitung bei Angabe der überwachenden Jobvariablen ändern sich dabei nicht. Insbesondere kann sich ein Benutzer nicht auf den Auftrag eines anderen Benutzers beziehen, auch dann nicht, wenn er auf dessen überwachende Jobvariable zugreifen kann.

2.8.3 Schutz von überwachenden Jobvariablen

Jobvariablen, die als überwachende Jobvariable verwendet werden, können ebenso gegen Zugriffe geschützt werden wie andere Jobvariablen. Insbesondere kann der Benutzer Kennwörter für den erweiterten Lese- und/oder Schreibschutz vergeben.

Das System umgeht diese Schutzmechanismen nicht, wenn es versucht, den überwachenden Jobvariablen Werte zuzuweisen. Daraus folgt, dass ein Benutzer selbst dafür Sorge zu tragen hat, dass seine Jobvariablen zugreifbar sind. Kennwörter müssen dem System beim ersten Zugriff bekannt sein (z.B. Kommando ADD-PASSWORD).

Kann die überwachende Jobvariable nicht zugeordnet werden, wird der Auftrag bzw. das Programm nicht gestartet.

Die Überwachung eines Auftrags oder Programms erfordert Schreibzugriffe auf die überwachende Jobvariable, um die entsprechenden Werte zu setzen. Wenn beim ersten Zugriff auf eine überwachende Jobvariable das Zugriffsrecht gegeben ist, dann bleibt dem System dieses Zugriffsrecht solange erhalten, wie die Jobvariable zur Überwachung benutzt wird.

Diese Vererbung der Zugriffsberechtigung beschränkt sich auf den Kennwortschutz (READ-PASSWORD/WRITE-PASSWORD).

Die Änderung der Schutzmerkmale (z.B. Mehrbenutzbarkeit oder BASIC-ACL) bei auftragsüberwachenden Jobvariablen ist nur möglich bei Aufhebung des MONJV-Schutzes.

Das System schützt die ersten 128 Bytes (Systembereich) einer auftragsüberwachenden Jobvariablen vor Schreibzugriffen. Bestimmte Felder des Systemteils (siehe [Tabelle 14 auf Seite 55](#)) können jedoch über das Kommando MODIFY-MONJV bzw. den Makroaufruf TIMJV geändert werden.

Der Systembereich einer programmüberwachenden Jobvariablen ist zwar nicht geschützt, sollte aber trotzdem nicht genutzt werden, da dabei die vom System hinterlegten Informationen überschrieben werden bzw. das System bei jeder Zustandsänderung diese 128 Bytes selbst neu überschreibt.

Zusätzlich ist der JV-Eintrag einer auftragsüberwachenden Jobvariablen selbst gegen Änderung geschützt. Dieser Schutz beginnt mit ENTER-JOB, LOGON bzw. SET-LOGON-PARAMETERS und wird wieder aufgehoben:

- bei Beendigung des überwachten Auftrags
- wenn TSOS die Überwachung eines Auftrags mit dem Kommando MODIFY-JOB-OPTIONS vorzeitig beendet
- bei expliziter Freigabe durch den Benutzer:

```
/MOD-JV-ATTR JV-NAME=jvname,PROTECTION=*PAR(MONJV-PROTECTION=*NO)
```

Während der Schutzdauer kann die Jobvariable keinem anderen Auftrag oder Programm als überwachende Jobvariable zugewiesen werden.

Bei der Systemeinleitung sind in folgenden Fällen Zugriffe auf überwachende Jobvariablen notwendig:

- Ein überwachter Auftrag wird aus der Auftragswarteschlange entfernt. Die überwachende Jobvariable wird auf \$A gesetzt, der Schreibschutz wird aufgehoben.
- Ein Auftrag wurde mit RERUN-AFTER-CRASH=*YES oder FLUSH-AFTER-SHUTDOWN=*NO gestartet und befindet sich noch in der Auftragswarteschlange. Die überwachende Jobvariable bleibt geschützt und wird ggf. auf \$\$ gestellt.

Zur Absicherung korrekter Werte in überwachenden Jobvariablen ist demnach notwendig, dass sie während der Systemeinleitung zugreifbar sind. Da mehrere Pubsets gleichzeitig betrieben werden können (siehe Handbuch „Einführung in die Systembetreuung“ [3]), ist Folgendes zu beachten:

- Eine überwachende Jobvariable sollte auf dem Home-Pubset des Rechners liegen, auf dem der Auftrag ausgeführt wird. Andere Pubsets sind zum Zeitpunkt der Systemeinleitung nicht zugreifbar.
- Im Fall fehlerhafter Beendigung einer BS2000-Session bleiben überwachende Jobvariablen den Zustand des überwachten Auftrags oder Programms entsprechend der letzten ausgeführten Eintragung an.

Besonderheiten bei Repeatjobs

Die MONJV enthält die TSN und den Auftragszustand des ersten Auftrags. Die TSN und der Auftragszustand einer nachfolgenden Wiederholung wird erst zum Startzeitpunkt des Folgeauftrages, d.h. zum LOGON-Zeitpunkt, aktualisiert. Eine Abfrage nach dem Zustand „\$\$“ ist für Folgeaufträge nicht möglich.

Dies ist auch bei Verwendung der MONJV zur Identifizierung von Aufträgen zu beachten (z.B. CANCEL-JOB). Dabei ist zu überlegen, ob der gerade laufende bzw. bereits beendete Auftrag oder der bereits in Type 1 befindliche Folgeauftrag angesprochen werden soll. Dieser Folgeauftrag kann erst über die MONJV angesprochen werden, wenn er gestartet ist.

Außerdem ist die MONJV von LOGOFF bis zum Start des Folgeauftrages nicht geschützt. Die Gefahr des Zugriffsverlustes bei langen Wartezeiträumen zwischen zwei Wiederholungen ist entsprechend groß.

Besonderheiten bei Kalenderjobs

Die MONJV ist während der gesamten Laufzeit geschützt, also auch für sich in Type 1 befindliche Folgeaufträge.

Die TSN bleibt während der gesamten Laufzeit gleich. Die MONJV enthält ab der Beendigung einer Jobausprägung bis zum Start der Nächsten den Terminierungszustand \$T oder \$A des Vorgängers. Eine Abfrage nach dem Zustand „\$S“ ist für Folgeaufträge nicht möglich.

2.8.4 Kettungsnamen überwachender Jobvariablen

Bei Erklärung einer Jobvariablen zur überwachenden Jobvariablen wird automatisch ein JV-LINK-Eintrag mit einem Standard-Kettungsnamen erzeugt. Die nochmalige Verwendung der Standard-Kettungsnamen innerhalb des überwachten Auftrags bzw. des programmaufrufenden Auftrags überschreibt den Eintrag. Der Zugriff auf die überwachende Jobvariable über den Standard-Kettungsnamen geht somit verloren.

Auftragsüberwachung

Eine auftragsüberwachende Jobvariable (siehe [Abschnitt „Kommandos zur Vereinbarung einer Auftragsüberwachung“ auf Seite 77](#)) wird bei LOGON-Verarbeitung mit dem Standard-Kettungsnamen **SMONJVJ** in die JV-LINK-Tabelle des zu überwachenden Auftrags eingetragen. Mit diesem Kettungsnamen ist die überwachende Jobvariable in Kommandos bzw. Makroaufrufen innerhalb des Auftrags ansprechbar.

Die Verwendung einer temporären Jobvariablen ist nicht möglich.

Programmüberwachung

Eine programmüberwachende Jobvariable wird beim Start bzw. Laden eines Programms (siehe [Abschnitt „Kommandos zur Programmüberwachung“ auf Seite 82](#)) mit dem Standard-Kettungsnamen **SMONJVP** in die JV-LINK-Tabelle des aufrufenden Auftrags eingetragen. Mit diesem Kettungsnamen ist die überwachende Jobvariable in Kommandos innerhalb des Auftrags ansprechbar. Das Programm kann ebenfalls über den Kettungsnamen auf die Jobvariable zugreifen. Der JV-LINK-Eintrag existiert von Beginn der Überwachung bis Programmende (TERM oder Kommando CANCEL-PROGRAMM).

Die Verwendung einer temporären Jobvariablen ist möglich. Andere Aufträge können dann aber nicht darauf zugreifen.

2.9 Bedingungsabhängige Auftragssteuerung

Die bedingungsabhängige Auftragssteuerung (CJC = Conditional Job Control) verwendet Jobvariablen und ihre Wertänderungen zur ereignisabhängigen Steuerung von Aufträgen. In bedingten Kommandos kann der Anwender sich auf die Werte von Jobvariablen beziehen. Wenn irgendeine Jobvariable verändert wird (Ereignis tritt ein), benachrichtigt das System alle Aufträge, in denen der geänderte Bereich dieser Jobvariablen in bedingten Kommandos verwendet wird. Als Änderung gilt bereits die Ersetzung eines Wertes durch den gleichen Wert. Ist die angegebene Bedingung durch das eingetretene Ereignis erfüllt, wird die gewünschte CJC-Funktion ausgeführt.

Mithilfe der im Folgenden beschriebenen Funktionen kann eine komplexe Steuerung für Jobketten und Jobnetze realisiert werden.

2.9.1 Bedingungen und Ereignisse

Der Benutzer kann in den Kommandos ADD-CJC-ACTION, SKIP-COMMANDS, WAIT-EVENT sowie in dem Makroaufruf ONEVT eine Bedingung mit Jobvariablen und Konstanten formulieren. Einfache Bedingungen können mit logischen Operatoren zu komplexen Bedingungen zusammengefasst werden.

Vom jeweiligen Zustand der Bedingung („erfüllt“ oder „nicht erfüllt“) ist die Wirkung des Kommandos abhängig. Dieser Bedingungsstatus wird durch Ereignisse, das sind Wertänderungen der dort angegebenen Jobvariablen, verändert. Bei jedem für die Bedingung relevanten Ereignis (z.B. ein MODIFY-JV-Kommando für eine darin verwendete Jobvariable) benachrichtigt das System den betreffenden Auftrag. Die Bedingung wird nun ausgewertet, und im „erfüllt“-Fall werden die hierfür vorgesehenen Aktionen ausgeführt. Die Wertänderung kann von jedem Auftrag erfolgen, der die entsprechende Zugriffsberechtigung für die Benutzer-Jobvariable besitzt. Sonder-Jobvariablen, die nur in Bedingungen des Kommandos SKIP-COMMANDS zulässig sind, können nur vom System geändert werden (z.B. Datum oder Zeit).

Die Behandlung von Ereignissen (d.h. die Bedingungsauswertung) innerhalb des Benutzerauftrags erfolgt in der Reihenfolge, in welcher der Auftrag darüber unterrichtet wird. Es ist nicht möglich, Aufträge über die Erzeugung von Jobvariablen zu benachrichtigen. Bedingte Kommandos mit noch nicht existenten Jobvariablen werden daher abgewiesen. Gegen erlöschen eingerichtete bedingte Kommandos oder Makros, falls der diese Jobvariable enthaltende Katalog endgültig exportiert wird, bzw. die Bedingungen werden durch das Kommando REMOVE-CJC-ACTION bzw. den Makroaufruf DONEVT explizit gelöscht.

2.9.2 Synchronisierung von Ereignissen

Ändert ein Auftrag mehrere Jobvariablen nacheinander, so werden diese Änderungen aufeinander folgend vermerkt und in derselben Reihenfolge von allen betroffenen Aufträgen ausgewertet. In einem Mehrrechnerverbund ist nicht automatisch sichergestellt, dass unsynchronisierte Wertänderungen derselben Jobvariablen durch verschiedene Aufträge von allen betroffenen Aufträgen in allen Rechnern in der gleichen Reihenfolge verarbeitet werden. Im Bedarfsfall muss hier der Benutzer die Wertänderungen mit den Kommandos MODIFY-JV-CONDITIONALLY oder WAIT-EVENT selbst synchronisieren.

2.9.3 Kommandos/Makros der Bedingungsabhängigen Auftragssteuerung

SHOW-CJC-STATUS

Mit diesem Kommando kann sich der Benutzer über Aufträge mit momentan wirksamen Anwendungen der Bedingungsabhängigen Auftragssteuerung (CJC = Conditional Job Control) informieren. Die Abfrage kann für Rechner, Kataloge oder Jobvariablen formuliert werden.

SKIP-COMMANDS

Dieses Kommando verzweigt innerhalb der Kommandofolge abhängig von der Auswertung einer als Operand angegebenen Bedingung.

Tritt in der Kommandofolge ein SKIP-COMMANDS auf, so werden die Werte aller im Bedingungsoperanden angegebenen Jobvariablen unmittelbar gelesen und der bedingte Ausdruck ausgewertet. Ergibt die Auswertung „Bedingung erfüllt“, so wird zur vorgegebenen Stelle in der Kommandofolge verzweigt. Das Sprungziel kann in der Kommandofolge vor oder hinter dem SKIP-COMMANDS-Kommando liegen. Ergibt die Auswertung „Bedingung nicht erfüllt“, so wird die Verarbeitung mit dem Kommando fortgesetzt, das auf SKIP-COMMANDS folgt.

WAIT-EVENT

Mit dem Kommando WAIT-EVENT kann ein Auftrag in den Wartezustand versetzt werden, bis eine angegebene Bedingung erfüllt ist oder bis ein vorgegebenes Zeitintervall verstrichen ist. Im letzteren Fall kann der Benutzer durch eine Marke angeben, wo der Auftrag nach Zeitablauf fortgesetzt werden soll. Das Sprungziel kann in der Kommandofolge vor oder hinter dem WAIT-EVENT-Kommando liegen. Ohne Angabe eines Sprungziels wird zum nächsten SET-JOB-STEP, EXIT-JOB, LOGOFF, END-PROCEDURE bzw. IF-BLOCK-ERROR in der Kommandofolge verzweigt.

ADD-CJC-ACTION

Mit dem Kommando ADD-CJC-ACTION kann ein Auftrag während eines bestimmten Zeitintervalls mehrfach auf die Erfüllung einer angegebenen Bedingung asynchron warten, während die normale Abarbeitung der nachstehenden Kommandofolge fortgesetzt wird. Für die Fälle „Bedingung erfüllt“ und Auftreten von „timeout“ können jeweils unterschiedliche Aktionen festgelegt werden. Erlaubte Aktionen sind das Starten von Aufträgen (ENTER-JOB), das Starten von Prozeduren (ENTER-PROCEDURE) und das Setzen von Jobvariablen (MODIFY-JV).

Eine CJC-Kommandofolge beginnt mit dem Kommando ADD-CJC-ACTION und endet mit dem Kommando END-CJC-ACTION. Dazwischen können zwei Kommandofolgen für „Bedingung erfüllt“ und „timeout“ mit beliebig vielen erlaubten Aktionen (ENTER-JOB-, ENTER-PROCEDURE- bzw. MODIFY-JV-Kommandos) stehen. Die Kommandofolgen werden vom System für eine spätere Ausführung gespeichert. Tritt das Ereignis „Bedingung erfüllt“ ein, so wird ein gerade bearbeitetes Kommando noch ausgeführt (WAIT-EVENT-Wartezustand wird sofort unterbrochen!) und die erste CJC-Kommandofolge abgearbeitet. Danach wird der Auftrag normal am Unterbrechungspunkt fortgesetzt. Entsprechend wird bei Auftreten von „timeout“ die zweite CJC-Kommandofolge abgearbeitet. Befindet sich der Auftrag im Programm-Modus (incl. TU-Contingency), so wird er sofort unterbrochen, und die vorgegebene CJC-Kommandofolge wird ausgeführt. Danach erfolgt die Rückverzweigung zur Unterbrechungsstelle im Programm.

Hinweis

Tritt das Ereignis „Bedingung erfüllt“ ein, wird die CJC-Kommandofolge durchlaufen. Werden mehrere Wiederholungen vereinbart, wird vor jedem erneuten Durchlauf geprüft, ob die Bedingung erfüllt ist.

Das Zeitintervall, in dem das System die Bedingung überprüfen soll (timeout), kann ebenfalls im Kommando angegeben werden.

Das Kommando ADD-CJC-ACTION bleibt im System solange wirksam, bis

- der Auftrag beendet (Kommando EXIT-JOB bzw. LOGOFF) bzw. abgebrochen (Kommando CANCEL-JOB) wird,
- die maximale Anzahl Ereignisse „Bedingung erfüllt“ eingetreten ist,
- die Lebensdauer „timeout“ abgelaufen ist,
- das ADD-CJC-ACTION gelöscht wird (REMOVE-CJC-ACTION), oder
- ein Katalog, der eine beteiligte Jobvariable enthält, exportiert wird

Hinweis

Solange das Kommando ADD-CJC-ACTION wirksam ist, kann kein Fixpunkt (Makro WRCPT) geschrieben werden. Ein Restart (Kommando RESTART-PROGRAM) beendet alle noch wirksamen Kommandos ADD-CJC-ACTION.

END-CJC-ACTION

Das Kommando END-CJC-ACTION dient in der Kommandofolge als Abschluss der CJC-Kommandofolge. Eine CJC-Kommandofolge beginnt mit einem ADD-CJC-ACTION-Kommando und endet mit einem END-CJC-ACTION-Kommando. Zwischen diesen begrenzenden Anweisungen stehen alle Kommandos, die im Fall von „Bedingung erfüllt“ bzw. nach Ablauf der angegebenen Zeit (timeout) ausgeführt werden sollen.

Nach END-CJC-ACTION wird die CJC-Kommandofolge unter einer auftragsbezogenen lokalen Nummer abgespeichert. Mit der Angabe dieser Nummer kann eine bestimmte CJC-Kommandofolge wieder gelöscht werden (REMOVE-CJC-ACTION).

REMOVE-CJC-ACTION

Mit diesem Kommando erlischt die Wirksamkeit einer bestimmten CJC-Kommandofolge oder aller noch wirksamen CJC-Kommandofolgen (Operand ALL).

Alle im Zusammenhang mit den gelöschten CJC-Kommandofolgen gespeicherten Informationen werden gelöscht.

ONEVT - Makroaufruf

Mit dem Makroaufruf ONEVT kann der Benutzer auf Programmebene Wertänderungen von Jobvariablen zur Steuerung seines Programmlaufs ausnutzen. Dazu kann als Operand eine Bedingung angegeben werden. Bei jeder Wertänderung einer dort verwendeten Jobvariablen wird vom System geprüft, ob die Bedingung erfüllt ist. Der Fall „Bedingung erfüllt“ hat die Eigenschaft eines Ereignisses im Sinne der TU-Ereignissteuerung über so genannte Event Items. Das bedeutet, dass vorher mit dem Makroaufruf ENAEI eine entsprechende Ereigniskennung eingerichtet und entweder das erwartete Ereignis mit SOLSIG angefordert oder dafür eine Contingency-Routine definiert werden muss.

DONEVT - Makroaufruf

Mit dem Makroaufruf DONEVT wird die Wirkung des Makros ONEVT aufgehoben. Da das Warten auf eine ONEVT-Bedingung (Jobvariablen-Ereignis) auch durch den TIMEOUT-Wert im SOLSIG aufgehoben werden kann, ist es sinnvoll (logisch zwingend), eine „verspätete“ Auswirkung des ONEVT mittels DONEVT und DISEI zu verhindern.

Andernfalls kann, wenn mit derselben Ereigniskennung ONEVT und SOLSIG wiederholt durchlaufen werden, mit diesem SOLSIG ein verspätetes Ereignis des alten ONEVT angefordert werden.

Alternativ hierzu kann in jedem Aufruf des Makros ONEVT dem Operanden POST ein anderer Wert zugewiesen werden, um Ereignisse verschiedener ONEVT unterscheiden und alte verwerfen zu können.

Nähere Informationen zum Thema Ereignissteuerung können dem Handbuch „Makroaufrufe an den Ablaufteil“ [4] entnommen werden.

Nachfolgende Tabelle zeigt den Anwendungsbereich von bedingten Kommandos und Makros.

Kommando / Makro	Programmebene	Kommandoebene			
		Batchverarbeitung		Dialogverarbeitung	
		Proz.Modus	Kom.Modus	Proz.Modus	Kom.Modus
SHOW-CJC-STATUS	--	x	x	x	x
SKIP-COMMANDS	--	x	x	x	--
WAIT-EVENT	--	x	x	x	x
ADD-CJC-ACTION	--	x	x	x	x
END-CJC-ACTION	--	x	x	x	x
REMOVE-CJC-ACTION	--	x	x	x	x
ONEVT	x	--	--	--	--
DONEVT	x	--	--	--	--

Tabelle 21: Anwendungsbereich bedingter Kommandos und Makroaufrufe

Hinweis

Temporäre Jobvariablen in den Kommandos/Makros der Bedingungsabhängigen Auftragssteuerung sind nur innerhalb des erzeugenden Auftrags möglich!
Der Zugriff auf temporäre Jobvariablen eines anderen Auftrags ist nicht möglich.

2.10 Eingabe aus Jobvariablen

Eine Jobvariable kann Kommando- bzw. Anweisungsteile ersetzen. SDF ersetzt die Jobvariable vor der Ausführung durch ihren Wert. Die daraus resultierende Eingabe wird von SDF syntaktisch geprüft.

Die Ersetzung von Jobvariablen ist im ungeführten Dialog, in Prozeduren und im Batchbetrieb möglich. Im (temporär) geführten Dialog ist sie nur in der NEXT-Zeile und in der Eingabe für Operandenwerte erlaubt.

Die Jobvariable wird angegeben:

- direkt über ihren Namen in der Form „&(jv-name)“.
- indirekt über ihren Jobvariablenkettungsname in der Form „&(*jv-link)“. Die Verknüpfung von Kettungsname mit der Jobvariablen erfolgt vorher durch das Kommando /SET-JV-LINK LINK-NAME = jv-link, JV-NAME = jv-name.

Hinweis

Vor der Jobvariablenersetzung ersetzt ACS ggf. den Aliasnamen einer JV durch ihren realen Pfadnamen (gemäß Aliaskatologeintrag).

Für die Ersetzung von Jobvariablen bestehen folgende Einschränkungen:

- Eine Jobvariable kann nur in ihrer gesamten Länge einen Ausdruck ersetzen.
- Für die einzusetzende Jobvariable muss Lesezugriff bestehen, sonst wird die Eingabe als Syntaxfehler abgewiesen.
- Innerhalb von CJC-Kommandofolgen ist das Ersetzen nicht möglich.
- Jobvariablen können keine Eingabedaten ersetzen. Anweisungen an Programme mit SDF-Oberfläche werden von SDF nicht wie Eingabedaten, sondern wie Kommandos behandelt.
- Jobvariablen können in Prozeduren oder ENTER-Dateien den führenden Schrägstrich vor Kommandos bzw. die zwei führenden Schrägstriche vor Anweisungen, den führenden Punkt von Nicht-S-Marken, den Strichpunkt als Kommandotrenner und das Fortsetzungszeichen nicht ersetzen.
- Jobvariablen können nicht als Prozedurparameter eingesetzt werden. Diese Einschränkung kann z.B. durch Verwendung eines Kettungsnamens umgangen werden (siehe *Beispiel 2*).
- Ausdrücke können nicht geschachtelt werden.
- Im Dialog wird eine Jobvariablen-Ersetzung in der oben gezeigten Form nur durchgeführt, wenn keine gleich lautende S-Variable bzw. Builtin-Funktion bekannt ist. Die Ersetzung wird in der Reihenfolge S-Variable, Builtin-Funktion und dann erst Jobvariable

durchgeführt (entspricht der Einstellung `JV-REPLACEMENT=*AFTER-BUILTIN-FUNCTION` in S-Prozeduren). Mit Angabe von Benutzer- und/oder Katalogkennung im Jobvariablen-Namen wird eindeutig eine Jobvariable referenziert.

Die Jobvariablen-Ersetzung in S-Prozeduren wird durch die Einstellung `JV-REPLACEMENT` im Kommando `SET-PROCEDURE-OPTIONS` auf folgende Weise beeinflusst:

- In S-Prozeduren ist `JV-REPLACEMENT=*NONE` voreingestellt, d.h. es werden nur S-Variablen bzw. Builtin-Funktionen ersetzt.
- Die Einstellung von `JV-REPLACEMENT` kann mit dem gleichnamigen Operanden im Kommando `SET-PROCEDURE-OPTIONS` in einer S-Prozedur geändert werden. Bei der Einstellung `JV-REPLACEMENT=*AFTER-BUILTIN-FUNCTION` können Inkompatibilitäten vermieden werden, wenn bei der bisherigen Form der Jobvariablen-Ersetzung der JV-Name mit Benutzer- und/oder Katalogkennung angegeben wird.

Die Jobvariablen-Ersetzung kann auch durch die Builtin-Funktion `JV()` erreicht werden. Die Angabe besitzt dann die Form „`&(JV(JV-NAME=string_ausdruck,...))`“. Hierbei ist auch die wahlweise Angabe eines Teilbereichs möglich. Siehe auch Builtin-Funktion `JV()` im Handbuch „Kommandos“ [1].

Beispiel 1: Ersetzung im Dialog

```
/cre-jv jv=cmd _____ (1)
```

```
/mod-jv jv=cmd,set-value='SHOW-FILE-ATTRIBUTES' _____ (2)
```

```
/&(cmd) _____ (3)
```

```
%      3 :20SG:$USER1.ALT.SYS.LOGON.USERPROC.X1
%     51 :20SG:$USER1.ALT.SYSSDF.USER.EXAMPLE.1
%     21 :20SG:$USER1.DATEI.1
%     48 :20SG:$USER1.DATEI.2
%     84 :20SG:$USER1.DATEI.3
%     66 :20SG:$USER1.OUT.SORT1-2
%      3 :20SG:$USER1.PROC.JV
%:20SG: PUBLIC:      7 FILES RES=      276 FRE=      39 REL=      21 PAGES
```

```
/mod-jv jv==cmd,set-value='-FILE-ATTRIBUTES FILE-NAME=PROC.' _____ (4)
```

```
/set-jv-link link-name=cmdlink,jv-name=cmd _____ (5)
```

```
/sh&(*cmdlink) _____ (6)
```

```
%      3 :20SG:$USER1.PROC.JV _____
%:20SG: PUBLIC:      1 FILE RES=      3 FRE=      2 REL=      0 PAGES
```

(1) Die Jobvariable mit dem Namen „CMD“ wird angelegt.

(2) Die Jobvariable CMD erhält den Wert „SHOW-FILE-ATTRIBUTES“

- (3) Nach Absenden des Kommandos wird der Variablenstring durch das in der Jobvariablen definierte Kommando ersetzt und das Kommando ausgeführt.

Hinweis

Die Jobvariable wird hier ersetzt, da keine S-Variable bzw. Built-in-Funktion mit Namen „CMD“ existiert. Mit Angabe von Benutzer- und/oder Katalogkennung im Jobvariablen-Namen (z.B. „&(\$user1.cmd)“ statt „&(cmd)“) kann die Jobvariable „CMD“ eindeutig referenziert werden.

- (4) Der Wert der Jobvariablen CMD wird geändert. Er enthält jetzt nur noch einen Teil des Kommandonamens „-FILE-ATTRIBUTES“ und den teilqualifizierten Dateinamen „PROC.“.
- (5) Der Jobvariablen CMD wird der Kettungsname „CMDLINK“ zugeordnet.
- (6) Nach Absenden des Kommandos wird der Variablenstring durch den der Jobvariablen zugeordneten Kommandoteil ersetzt und das Kommando ausgeführt. Der Bezug zur Jobvariablen wird über den Kettungsnamen hergestellt.

Beispiel 2: Ersetzung in einer Nicht-S-Prozedur

```

/BEGIN-PROC PAR=YES(PROC-PAR=(&PARAM1)) _____ (1)
.
.
/SET-JV-LINK LINK-NAME=PARAM1 ,JV-NAME=&PARAM1 _____ (2)
/&(*PARAM1) FILE-NAME=LST.JOB _____ (3)
.
.
/END-PROC

```

- (1) Die durch den Prozedurparameter PARAM1 spezifizierte Jobvariable soll das jeweils auszuführende Kommando enthalten. Da die Angabe „&(&PARAM1)“ nicht zulässig ist, wird im Folgenden der Umweg über einen Kettungsnamen beschritten.
- (2) Der aktuelle Jobvariablenname wird für den Prozedurparameter PARAM1 eingesetzt und mit dem Kettungsnamen PARAM1 verknüpft.
- (3) Für den Kettungsnamen PARAM1 wird der Inhalt der vereinbarten Jobvariablen eingesetzt. Enthält die Jobvariable z.B. den Wert PRINT-DOCUMENT, so wird die Datei LST.JOB ausgedruckt.

*Beispiel 3: Ersetzung in einer S-Prozedur***S-Prozedur *SPROC*:**

```

/          SET-PROC-OPT  JV-REPLACE=*AFTER-BUILTIN _____ (1)
/          DECL-PAR      JV-1(INIT=*PROMPT) _____ (2)
/          &(JV(JV-NAME=JV-1))  FILE-NAME=LST.JOB _____ (3)
/FEHL:    IF-BLOCK-ERROR
/          WRITE-TEXT  C '** Fehler &MC **'
/          ELSE
/          WRITE-TEXT  C '** Kommando &(&(JV-1)) ausgefuehrt **'
/          END-IF _____ (4)
/ENDE:    EXIT-PROC

```

Aufruf der Prozedur *SPROC*:

```

/create-jv  jv=jv.kommando
/modify-jv  jv=jv.kommando,set-val='SHOW-FILE-ATTRIBUTES'
/call-proc  sproc
%JV-1:  jv.kommando
% DMS0533 REQUESTED FILE NOT CATALOGED IN PUBSET '10SN'. COMMAND TERMINATED
% SDP0004 ERROR DETECTED AT COMMAND LINE: 3 IN PROCEDURE':10SN:$USER1.SPROC'
** FEHLER DMS0533 ** _____ (5)

```

- (1) Mit dem Operanden **JV-REPLACEMENT** des Kommandos **SET-PROCEDURE-OPTIONS** wird vereinbart, dass Jobvariablen-Ersetzung zulässig ist.
- (2) Vereinbart den Prozedurparameter **JV-1**. Der Wert für diesen Parameter soll im Dialog abgefragt werden.
- (3) Jobvariablen-Ersetzung wird mit der Builtin-Funktion **JV()** realisiert (unabhängig von der **JV-REPLACEMENT**-Einstellung). Der Ausdruck **&(JV(JV-NAME=JV-1))** wird bei Prozeduraufruf durch den Wert der Jobvariable ersetzt, deren Name im Prozedurparameter **JV-1** übergeben wird.
- (4) Tritt ein Fehler bei der Jobvariablen-Ersetzung auf, wird der Text „** Fehler“ mit dem entsprechenden Meldungsschlüssel (**&MC**) ausgegeben. Bei fehlerfreiem Ablauf wird der Text der **ELSE**-Schleife angezeigt. Auch hier wird der Ausdruck **&(&(JV-1))** durch den Wert der definierten Jobvariable ersetzt. Während der erste Schritt der Ersetzung von **&(JV-1)** unabhängig von der **JV-REPLACEMENT**-Einstellung erfolgt (**JV-1** ist eindeutig eine S-Variable), funktioniert der zweite Schritt der Ersetzung nur mit der Einstellung **JV-REPLACEMENT=*AFTER-BUILTIN**.
- (5) Die angeforderte Datei ist im gewünschten Pubset nicht vorhanden.

3 Kommandos

Im folgenden Kapitel werden die im Zusammenhang mit Jobvariablen benötigten BS2000-Kommandos aufgelistet. Sie sind in vier Gruppen aufgeteilt:

- Verwaltung von Jobvariablen
- Auftragsüberwachung
- Programmüberwachung
- Bedingungsabhängige Auftragssteuerung

Die Kommandos zur Auftrags- bzw. Programmüberwachung sind mit den relevanten Operanden jeweils in einer Tabelle zusammengefasst.



Eine vollständige Darstellung der Kommandosyntax und die Beschreibung aller Operanden finden Sie im Handbuch „Kommandos“ [1].

Die funktionsgleichen Makroaufrufe für die Verwaltung von Jobvariablen werden im Abschnitt „Makroaufrufe“ (siehe [Seite 87](#)) beschrieben.

3.1 Kommandos zur Verwaltung von Jobvariablen

Kommandoübersicht

Kommando	Makro	Funktion
COPY-JV	COPJV	Jobvariable kopieren
CREATE-JV	CATJV	Jobvariable neu erzeugen
DELETE-JV	ERAJV	Jobvariable löschen
MODIFY-JV	SETJV	Jobvariableninhalt ändern
MODIFY-JV-ATTRIBUTES	CATJV	Katalogeintrag für Jobvariable ändern
MODIFY-JV-CONDITIONALLY	CSWJV	Jobvariableninhalt bedingt ändern
MODIFY-MONJV	TIMJV	Elemente im Systemteil einer auftragsüberwachenden JV setzen
REMOVE-JV-LINK	RELJV	JV-LINK-Eintrag löschen
SET-JV-LINK	DCLJV	Jobvariablen-Kettungsnamen definieren
SHOW-JV	GETJV	Jobvariableninhalt ausgeben
SHOW-JV-ATTRIBUTES	STAJV	Jobvariablen-Attribute ausgeben
SHOW-JV-LINK	LNKJV	JV-LINK-Eintrag ausgeben

Tabelle 22: Kommandos zur Verwaltung von Jobvariablen

Die Kommandos zur Verwaltung von Jobvariablen können im Dialog- und Batchbetrieb verwendet werden.

Eine vollständige Beschreibung der Kommandos ist im Handbuch „Kommandos“ [1] enthalten.

3.2 Kommandos zur Auftragsüberwachung

Dieser Abschnitt gibt einen Überblick über die Kommandos zur Auftragsverwaltung:

- Das sind zunächst Kommandos, die eine Auftragsüberwachung mit Jobvariablen ermöglichen (siehe [Tabelle 23](#)).
- Daneben gibt es Kommandos, die einen Auftrag verwalten oder über einen Auftrag informieren. Wenn ein Auftrag überwacht wird, kann er über den Namen seiner überwachenden Jobvariablen (MONJV) bezeichnet werden (siehe [Tabelle 24](#)).

Die nachfolgend aufgelisteten Kommandos sind entweder vollständig im Handbuch „Kommandos“ [1] oder in dem jeweils explizit genannten Produkt-Handbuch beschrieben.

3.2.1 Kommandos zur Vereinbarung einer Auftragsüberwachung

Die folgenden Kommandos leiten einen Auftrag ein (nur MODIFY-JOB-OPTIONS bezieht sich auf einen bereits existierenden Auftrag) und bieten die Möglichkeit, im Operanden MONJV eine Jobvariable zur Überwachung dieses Auftrags zu definieren. Zusätzlich kann im Operanden JV-PASSWORD ein für den Jobvariablen-Zugriff erforderliches Kennwort angegeben werden.

Kommando	Funktion
ENTER-JOB	beauftragt das System eine Kommandofolge als eigenständigen Auftrag (Batchauftrag) auszuführen (siehe auch Hinweise zu ENTER-JOB)
ENTER-PROCEDURE	startet eine Kommandofolge, die in einer Prozedurdatei gespeichert ist, als Batchauftrag
MODIFY-JOB-OPTIONS	ändert die Parameter (Protokollierung und Auftragsüberwachung) eines bestehenden Auftrags (siehe auch Hinweis zu MODIFY-JOB-OPTIONS)
PRINT-DOCUMENT	leitet einen Druckauftrag ein
SET-LOGON-PARAMETERS.	leitet einen Dialog- oder Batchauftrag ein (siehe auch Hinweise zu ENTER-JOB)
TRANSFER-FILE	überträgt eine Datei vom lokalen System zum fernen System oder umgekehrt (siehe „openFT“ [15])

Tabelle 23: Kommandos zur Vereinbarung der Auftragsüberwachung

Hinweise zur Vereinbarung einer MONJV

Die im Operanden MONJV angegebene Jobvariable darf nicht gegen Überschreiben geschützt sein.

Falls die Jobvariable zum Zeitpunkt der Kommandoverarbeitung nicht zugreifbar ist, wird eine Fehlermeldung nach SYSOUT ausgegeben und das Kommando abgewiesen. Existiert die angegebene Jobvariable in der eigenen Benutzerkennung noch nicht, wird sie vom System erzeugt (nur wenn der Auftrag vom System angenommen wird) und ist mehrbenutzbar.

Ist die Jobvariable mit einem Kennwort geschützt, kann die Erklärung zur überwachenden Jobvariablen nur erfolgen, wenn der Auftraggeber das Kennwort bereits in die Kennwort-Tabelle eingetragen hat (Kommando ADD-PASSWORD) oder explizit im Operanden JV-PASSWORD angibt (Der Operand JV-PASSWORD wird ignoriert, wenn keine MONJV vereinbart wird.)

Hinweise zu ENTER-JOB

Bei Überwachung eines Repeatjobs kann die MONJV jeweils nur eine TSN und den zugehörigen Auftragszustand enthalten. Die Wiederholung des Repeatjobs (nächster Auftrag) kann erst ab dem Startzeitpunkt der MONJV entnommen werden. Dies ist besonders zu beachten, wenn die MONJV zur Identifizierung eines Auftrags in Kommandos verwendet werden soll (z.B. CANCEL-JOB).

Der Zustand „\$S“ kann für Auftragswiederholungen nicht abgefragt werden, da die Protokollierung erst mit dem Startzeitpunkt beginnt (mit „\$R“). Außerdem ist zu beachten, dass die MONJV nach Beendigung eines Auftrags bis zum Start des nachfolgenden Auftrags *nicht* geschützt ist (Gefahr des Zugriffsverlustes).

Operandenangaben im SET-LOGON-PARAMETERS- bzw. LOGON-Kommando zu Beginn der katalogisierten Kommandofolge werden nur ausgewertet, wenn der jeweilige Operand im ENTER-JOB nicht angegeben wird und der Auftrag entweder an der Bedienstation oder unter Angabe von DEFAULT-FROM-FILE=*YES gestartet wird. In allen anderen Fällen kann eine MONJV deshalb nur im Kommando ENTER-JOB vereinbart werden.

Ein Batchauftrag, der auf einem Remote-Rechner ablaufen soll, ist über eine MONJV nur zugreifbar, wenn im MRSCAT der beteiligten Rechner jeweils die Katalogkennung des Home-Pubsets des Partnerrechners eingetragen ist.

In einem Batchauftrag ist es das erste Kommando innerhalb der abzuarbeitenden Kommandofolge. Der Batchauftrag wird mit ENTER-JOB in Auftrag gegeben.

Hinweise zu SET-LOGON-PARAMETERS

In einem Batchauftrag ist es das erste Kommando innerhalb der abzuarbeitenden Kommandofolge. Der Batchauftrag wird mit ENTER-JOB in Auftrag gegeben.

Bei Ausführung eines Batchauftrages, der nicht an der Bedienstation und nicht unter Angabe von DEFAULT-FROM-FILE=*YES gestartet wurde, werden die Operandenangaben im Kommando SET-LOGON-PARAMETERS bzw. LOGON nicht ausgewertet. Das Vereinbaren einer überwachenden JV ist in diesem Fall nur über das ENTER-JOB-Kommando möglich.

Hinweis zu MODIFY-JOB-OPTIONS

Nur ein privilegierter Benutzer (Privileg TSOS) kann die Überwachung eines bestehenden Auftrags mit dem Kommando MODIFY-JOB-OPTIONS verändern. Dabei kann er im Operanden MONJV eine bestehende Überwachung vorzeitig beenden oder die Überwachung mit der angegebenen Jobvariable starten bzw. bei bestehender Überwachung auf die angegebene Jobvariable übertragen.

3.2.2 Kommandos der Auftragsüberwachung

Die folgenden Kommandos verwalten einen Auftrag oder informieren. Falls der Auftrag von einer Jobvariablen überwacht wird, kann dieser Auftrag anstelle der TSN wahlweise über seine überwachende Jobvariable (Name oder Linkname) identifiziert werden. Die Angabe erfolgt im Operanden MONJV (bis auf wenige Ausnahmen in der Struktur JOB-IDENTIFICATION=*MONJV(...)).

Kommando	Funktion
CANCEL-FILE-TRANSFER	bricht einen Dateiübertragungs-Auftrag ab, der unter der eigenen Benutzerkennung läuft oder auf diese zugreift (siehe „openFT“ [15])
CANCEL-JOB	bricht einen Dialog-, Batch- oder Druckauftrag ab, der unter der eigenen Benutzerkennung läuft (siehe auch Hinweis zu CANCEL-JOB)
CANCEL-PRINT-JOB	Bricht einen Druckauftrag ab, der unter der eigenen Benutzerkennung läuft
CHANGE-TASK-PRIORITY	ändert die Ablaufpriorität eines gestarteten Auftrags
FORCE-JOB-CANCEL	bricht den angegebenen Auftrag unverzüglich ab
HOLD-JOB	versetzt einen noch nicht gestarteten Auftrag in den Wartezustand
MODIFY-JOB	ändert die Eigenschaften des Auftrags (nur in Typ 1)
MODIFY-JOB-OPTIONS	ändert die Parameter (Protokollierung und Auftragsüberwachung) eines bestehenden Auftrags
RESUME-JOB	hebt den Wartezustand für den angegebenen Auftrag auf
SET-SYSLST-READ-MARK	setzt eine Lesemarke in der SYSLST-Datei des Auftrags
SET-SYSOUTREAD-MARK	setzt eine Lesemarke in der SYSOUT-Datei des Auftrags
SHOW-FILE-TRANSFER	gibt Informationen über einen Dateiübertragungs-Auftrag aus (siehe „openFT“ [15])
SHOW-JOB-STATUS	gibt Informationen über einen Auftrag aus
SHOW-RESOURCE-ALLOCATION	gibt Informationen über Betriebsmittel aus, die der Auftrag belegt

Tabelle 24: Kommandos zur Vereinbarung der Auftragsüberwachung

Die Kommandos werden abgewiesen, wenn die angegebene Jobvariable keinen Auftrag überwacht oder wenn auf die überwachende Jobvariable nicht zugegriffen werden kann (Schutz durch ein Kennwort).

In der JV-LINK-Tabelle des überwachten Auftrags wird die überwachende Jobvariable automatisch mit dem Kettungsname **SMONJVJ** eingetragen. Der Auftrag kann mit diesem Standard-Kettungsname auf seine überwachende Jobvariable und die enthaltenen Infor-

mationen zugreifen, ohne dass der Name vorher bekannt ist. Dies ist vor allem für katalogisierte Kommandofolgen von Bedeutung, wenn dafür mehrere Batchaufträge gleichzeitig laufen sollen. Es ist dabei zu beachten, dass der Standard-Kettungsname in der Kommandofolge des Auftrags nicht explizit vergeben wird und somit die Zuordnung überschreibt.

Hinweis zu CANCEL-JOB

Wurde der Auftrag abnormal beendet, d.h. das Kommando CANCEL-JOB wurde mit STEPS=*ALL (Voreinstellung) abgesetzt, wird die Zustandsanzeige einer überwachenden Jobvariablen auf den Wert \$A gesetzt. Zusätzlich werden die Informationen über den Urheber in verkürzter Form und der Kommentar aus dem Operanden TEXT des Kommandos in den Systemteil (die ersten 128 Bytes) der überwachenden Jobvariable übernommen:

- Die Urheberinformation beginnt ab Byte 37 und besteht aus der Zeichenfolge *CAN:info*, wobei *info* die ersten 27 Bytes der nach SYSOUT gelieferten Urheberinformation enthält.
- Der Kommentar beginnt ab Byte 70 und besteht aus der Zeichenfolge *TEXT:text*, wobei *text* die ersten 51 Bytes des im Operanden angegebenen Kommentars enthält.

3.3 Kommandos zur Programmüberwachung

Dieser Abschnitt gibt einen Überblick über Kommandos, in denen Jobvariablen zur Programmüberwachung definiert werden können.

Kommandoübersicht

Kommando	Funktion
LOAD-EXECUTABLE-PROGRAM (bzw. LOAD-PROGRAM)	lädt ein Programm (LLM, Lade- oder Bindemodul) in den Speicher
RESTART-PROGRAM	startet ein Programm an seinem Fixpunkt (Wiederanlauf)
START-EXECUTABLE-PROGRAM (bzw. START-PROGRAM)	lädt und startet ein Programm (LLM, Lade- oder Bindemodul)

Tabelle 25: Kommandos zur Programmüberwachung

Eine vollständige Beschreibung der Kommandos ist im Handbuch „Kommandos“ [1] zu finden. Zusätzliche Informationen zum Laden und Starten von Programmen enthält das Handbuch „Bindelader-Starter“ [14].

Hinweise zur Verwendung einer MONJV

- Die Jobvariable darf nicht gegen Überschreiben geschützt sein.
- Wird eine temporäre Jobvariable verwendet, so ist zu beachten, dass nur der aufrufende Auftrag Zugriff zu der überwachenden Jobvariablen hat.
- Eine nicht vorhandene Jobvariable wird neu angelegt.
- Existiert die Jobvariable zum Zeitpunkt der Kommandoabgabe und ist sie wegen eines schwer wiegenden Fehlers nicht zugreifbar (z.B. Katalog zerstört), wird eine Meldung auf SYSOUT ausgegeben und das Kommando abgewiesen.
- Das Programm kann mit diesem Standard-Kettungsnamen **SMONJVP** auf seine überwachende Jobvariable zugreifen, ohne dass der Name bei der Programmerstellung bekannt ist (siehe „Kettungsnamen überwachender Jobvariablen“ auf Seite 64).

3.4 Kommandos zur bedingungsabhängigen Auftragssteuerung

Im folgenden Abschnitt werden die Kommandos der bedingungsabhängigen Auftragssteuerung aufgelistet und die Regeln zur Bildung von bedingten Ausdrücken beschrieben.

Die Abkürzung CJC steht für Conditional Job Control = Bedingungsabhängige Auftragssteuerung.

3.4.1 Kommandoübersicht

Kommando	Funktion
ADD-CJC-ACTION	Leitet eine CJC-Kommandofolge ein
END-CJC-ACTION	Kennzeichnet das Ende einer CJC-Kommandofolge
REMOVE-CJC-ACTION	Beendet die Wirksamkeit einer CJC-Kommandofolge
SHOW-CJC-STATUS	Gibt Informationen aus über Aufträge, die momentan mit bedingungsabhängiger Auftragssteuerung arbeiten
SKIP-COMMANDS	Verzweigt bedingt innerhalb einer Kommandofolge
WAIT-EVENT	Leitet einen bedingten Wartezustand ein

Tabelle 26: Kommandos zur bedingungsabhängigen Auftragssteuerung

Bei den Kommandos ADD-CJC-ACTION, SKIP-COMMANDS und WAIT-EVENT kann im Operanden CONDITION eine komplexe Bedingung mithilfe von Jobvariablen-Namen und -Werten angegeben werden. Dieser zentrale Metabegriff „Bedingter Ausdruck“ ist im nachfolgenden [Abschnitt „Bedingte Ausdrücke“](#) ausführlich dargestellt.

Eine vollständige Beschreibung der Kommandos ist im Handbuch „Kommandos“ [1] enthalten.

3.4.2 Bedingte Ausdrücke

Mit einem bedingten Ausdruck kann die Verarbeitung der Kommandos ADD-CJC-ACTION, SKIP-COMMANDS oder WAIT-EVENT von Jobvariablenwerten abhängig gemacht werden. Ein bedingter Ausdruck kann „wahr“ oder „falsch“ sein. Er besteht aus einem oder mehreren relationalen Ausdrücken <relation.ausdr>, die wie folgt aufgebaut werden können. Alternative Möglichkeiten sind durch „/“ getrennt; wahlfreie Angaben sind in eckige Klammern eingeschlossen.

<relation.ausdr>	::=	(<term> <vergleichts-op> <term>)
<term>	::=	<jvid> / <jv-teilfolgen-def> / <konst>
<vergleichts-op>	::=	< / > / = / <= / >= / <> / LT / GT / EQ / LE / GE / NE
<jvid>	::=	<jvname> / <*jvlink> / <#jvname> / <sonder-jvname>
<jvname>	::=	Name einer permanenten Benutzer-Jobvariablen
<*jvlink>	::=	Kettungsname einer Benutzer-Jobvariablen mit vorangestelltem Stern (*)
<#jvname>	::=	Name einer temporären Benutzer-Jobvariablen
<sonder-jvname>	::=	Name einer Sonder-Jobvariablen ohne Katalogkennung, aber mit Angabe der Benutzerkennung SYSJV
<jv-teilfolgen-def>	::=	(<jvid> [, [<start>], [<länge>]])
<start>	::=	<integer 1..256> Default-Wert = 1
<länge>	::=	<integer 1..64> Default-Wert = 64 (<start> + <länge> ≤ 257)
<konst>	::=	<c-string 1..64> / <x-string 1..128> zum Beispiel: C'HALLO' bzw. 'HALLO' oder X'00FF' bzw. X'0FF'

Die Operatoren LT, GT, EQ, LE, GE, NE müssen mit einem Leerzeichen von anderen Buchstaben (in Konstanten, Jobvariablen- oder Kettungsname) getrennt werden.

Besonders zu beachten ist außerdem, dass Groß- und Kleinschreibung in bedingten Ausdrücken erhalten bleibt.

Komplexe bedingte Ausdrücke ergeben sich durch Kombination von relationalen Ausdrücken mithilfe von logischen Operatoren:

<bed.ausdr>	::=	(<relation.ausdr> <logisch.op> <relation.ausdr>) / (<bed.ausdr> <logisch.op> <bed.ausdr>) / (NOT <bed.ausdr>)
<logisch.op>	::=	AND / OR / XOR logisches UND logisches ODER exklusives ODER

Beispiel

(JV1=C'START')

((JV2=C'O.K.') AND (*LINK<=C'12'))

((((JV4,10,3)=C'NEU') OR (#TEMP.JV=C'Y') AND (*LINK2=X'00'))

Hinweis

Die Auswertung eines bedingten Ausdrucks erfolgt „von innen nach außen“ entsprechend der Klammerung. Auf der gleichen Klammerstufe werden die logischen Operationen in folgender Reihenfolge durchgeführt:

1. NOT
2. AND
3. OR
4. XOR

So wird z. B. der Ausdruck

```
(NOT (JV1=C'ABC') OR (JV2=C'Z') AND (JV3<>JV4))
```

ausgewertet wie

```
((NOT (JV1=C'ABC')) OR ((JV2=C'Z') AND (JV3<>JV4)))
```

Beim Vergleich von Jobvariablen-Werten sind folgende Hinweise zu beachten:

- die Auswertung relationaler Ausdrücke erfolgt byteweise von links nach rechts. Für das Vergleichsergebnis ist das jeweilige Bitmuster im EBCDIC-Code maßgebend. So sind z.B. die abdruckbaren Ziffern 0 bis 9 (X'F0' bis X'F9') „größer“ als Buchstaben A ... Z (X'C1' ... X'E9'), und Großbuchstaben größer als Kleinbuchstaben.
- Liegt der Vergleichswert außerhalb des Definitionsbereichs einer Jobvariablen oder ist die Jobvariable leer (z.B. nach Deklaration), ist das Vergleichsergebnis in jedem Fall „falsch“, auch wenn die Abfrage mit '<>' (ungleich) erfolgt.
- Besteht der Inhalt einer Jobvariablen aus binären Nullen, so ist die Jobvariable nicht leer. Es gilt des Weiteren, dass bei zwei ansonsten gleichen Jobvariablen-Werten derjenige Wert als der größere gilt, der um eine binäre Null länger ist.

Beispiel

Die folgenden Ausdrücke sind „wahr“

```

C' ' < C'A'
C'a' < C'A'
C'A' < C'B'
C'B' < C'BB'
C'GUTEN ABEND' < C'GUTEN MORGEN'
C'ZZZZZZZZ' < C'0' (!)
C'8' < C'9'
C'899999999' < C'9' (!)

X'81' < X'C1'
X'0123' < X'0124'
X'C1' < X'C100'
X'FOFOFOFO' < X'F1'
X'3FFF' < C' '

```

Die folgenden Abfragen werden mit „falsch“ beantwortet.

```
(IN)      CALL-PROCEDURE NAME=PROC.JV
(IN)      /.ANF BEGIN-PROCEDURE LOGGING=*ALL
(IN)      /CREATE-JV JV=JV1
(IN)      /SKIP-COMMANDS TO-LABEL=END,IF=*JV(CONDITION=((JV1,11,1) NE '2'))
(OUT)     % CJC0011 SKIP COMMAND: CONDITION = FALSE
(IN)      /MODIFY-JV JV-CONTENTS=*SUBSTRING
( )       (JV-NAME=JV1,POSITION=1,LENGTH=5),SET-VALUE=C'12345'
(IN)      /SKIP-COMMANDS TO-LABEL=END,IF=*JV(CONDITION=((JV1,6,2) EQ 'A1'))
(OUT)     % CJC0011 SKIP COMMAND: CONDITION = FALSE
(IN)      /SKIP-COMMANDS TO-LABEL=END,IF=*JV(CONDITION=((JV1,6,2) NE 'A1'))
(OUT)     % CJC0011 SKIP COMMAND: CONDITION = FALSE
(IN)      /.END END-PROCEDURE
```

Ein bedingter Ausdruck wird in folgenden Fällen zurückgewiesen:

- Eine JV, auf die Bezug genommen wird, ist nicht zugreifbar (zerstörter Katalog).
- Eine JV, auf die Bezug genommen wird, ist im angegebenen Katalog nicht vorhanden.
- Der Benutzer ist zum Lesezugriff nicht berechtigt bzw. das für den Lesezugriff erforderliche Kennwort ist nicht in der Kennwort-Tabelle des Auftrags eingetragen.

4 Makroaufrufe

In diesem Kapitel ist die Makroschnittstelle der Jobvariablen-Funktionen für Assembler-Programmierer beschrieben. Die Makros haben im Wesentlichen die gleichen Funktionen wie die entsprechenden Kommandos.

4.1 Makroübersicht

Makro	Funktion
CATJV	Jobvariable katalogisieren
COPJV	Jobvariable kopieren
CSWJV	Jobvariable prüfen und setzen
DCLJV	Jobvariablen-Kettungsnamen definieren
DONEVT	Bedingung für Jobvariablen-Ereignis löschen
ERAJV	Jobvariable löschen
GETJV	Jobvariablenwert ausgeben
JVSEL	Jobvariablenauswahl des STAJV-Makros auf bestimmte Merkmale beschränken
LNKJV	JV-LINK-Einträge ausgeben
ONEVT	Bedingung für Jobvariablen-Ereignis setzen
RELJV	JV-LINK-Eintrag löschen
SETJV	Jobvariable setzen
STAJV	Merkmale von Jobvariablen ausgeben
TERM	Programm und Prozedurabschnitt beenden
TIMJV	Elemente im Systemteil einer auftragsüberwachenden JV versorgen

Tabelle 27: Makroübersicht

S-Typ-Makros können die Operanden MF, PREFIX, PARAM und MACID verwenden. Nachfolgend sind die möglichen Werte und ihre Bedeutung beschrieben. Bei der Beschreibung der einzelnen Makroaufrufe ist angegeben, welche Werte für den jeweiligen Makroaufruf zulässig bzw. Voreinstellung sind.

4.2 Allgemeine Makro-Operanden

4.2.1 Der Operand MF

Der Operand MF bestimmt die Art der Makrogenerierung.

MF-Angabe	Bedeutung/Auswirkung
MF=S	ist Voreinstellung: Generiert zuerst den Befehlssteil, dann den Datenbereich. Der Datenbereich enthält keine Feldnamen. Der Standardheader ist initialisiert.
MF=D	Erzeugt einen Datenbereich mit Feldnamen und erläuternde Equates. Die Bildung der Feldnamen kann im Aufruf mit den Operanden PREFIX bzw. MACID beeinflusst werden. Der Datenbereich beginnt mit dem Statement DSECT. Diese Art der Makroauflösung wird im Folgenden als Erzeugung einer DSECT bezeichnet. Eine DSECT beschreibt die Struktur eines Speicherbereiches, ohne selbst Speicherplatz zu belegen. Der Adresspegel wird auf null gesetzt.
MF=C	Erzeugt einen Datenbereich mit Feldnamen und erläuternden Equates. Die Bildung der Feldnamen kann im Aufruf mit den Operanden PREFIX bzw. MACID beeinflusst werden.
MF=L	Erzeugt nur den Datenbereich. Der Datenbereich enthält keine Feldnamen. Der Standardheader ist initialisiert. Auf den erzeugten Datenbereich kann in einem Aufruf mit MF=E verwiesen werden.
MF=E	Es werden nur die zum Aufruf des Funktionsbausteins notwendigen Befehle generiert. Der Befehlssteil endet i.d.R. mit einem SVC. Im Makroaufruf muss die Adresse des Datenbereichs mit den Parameterwerten angegeben sein. MF=(E,adr) MF=(E,(r)) Nur der Verarbeitungscode (SVC) wird generiert. adr bzw. das Register r enthalten die Adresse des Datenbereichs (Parameterliste) für die MF=L-Auflösung. MF=E[,PARAM=adr] MF=E[,PARAM=(r)] Der Operand PARAM bezeichnet die Adresse der Operandenliste. r = Register, das die Adresse des Datenbereichs enthält. Vor dem Makroaufruf muss das Register mit diesem Adresswert geladen werden.
MF=M	Der Aufruf erzeugt Befehle, die in der bereits vorhandenen Operandenliste alle explizit angegebenen Funktionsoperanden mit den neuen Werten versorgen.

Tabelle 28: MF-Operand in Makroaufrufen

Bei getrennter Generierung von Befehlssteil und Datenbereich müssen die in den Übersetzungsparametern (VERSION, PARMOD) angegebenen Werte übereinstimmen.

Die Verwendung eines Standardheaders ist von dem Übersetzungsparameter VERSION abhängig. Ohne Standardheader wird im Befehlssteil der SVC 133 generiert, mit Standardheader der SVC 190.

Parameterlisten müssen auf Wortgrenze ausgerichtet sein.

Die Versorgung der Parameterliste sollte nur über den entsprechenden Makroaufruf mit den explizit angegebenen Operanden erfolgen. Die einzelnen Bits besitzen oft (besonders bei CATJV) eine komplexe Abhängigkeit. Deshalb ist das direkte Verändern der Parameterliste nicht zu empfehlen.

Register R1 wird bei der Makroverarbeitung mit der Adresse der Parameterliste geladen (bei MF=E und MF=S).

4.2.2 Der Operand PREFIX

In den Makros ist es möglich, in verschiedenen Formen der Auflösung den Operanden PREFIX anzugeben. Dieser steuert die Generierung der symbolischen Namen für die Parameterliste. Der Präfix setzt sich aus maximal drei Buchstaben zusammen. Bei welchen Makroauflösungen ein Präfix angegeben werden kann, ist im Abschnitt „Allgemeines“ einer jeden Makrobeschreibung vermerkt. Die mögliche Eingabe einer beliebigen dreistellige Buchstabenkombination ist mit PREFIX=pre angegeben.

4.2.3 Der Operand PARAM

Der Operand PARAM bezeichnet die Adresse der Operandenliste.

PARAM dient zur Adressierung der Operandenliste.

=adr symbolische Adresse der Operandenliste.

=(r) Register, das die symbolische Adresse der Operandenliste enthält.

4.2.4 Der Operand MACID

Mit dem Operanden MACID kann der Anwender das zweite bis einschließlich vierte Zeichen der Namen in einer Parameterliste bestimmen. Bei welchen Makroauflösungen eine MACID angegeben werden kann, ist im Abschnitt „Allgemeines“ einer jeden Makrobeschreibung vermerkt. Die mögliche Eingabe einer beliebigen dreistellige Buchstabenkombination ist mit MACID=macid angegeben.

Weitere Informationen zu den Operanden MF, PREFIX, PARAM und MACID siehe Handbuch „Makroaufrufe an den Ablaufteil“ [4].

4.3 Erläuterung der verwendeten Metazeichen

Formale Darstellung	Erläuterung	Beispiel
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Konstanten, die in dieser Form vom Benutzer eingegeben werden müssen.	ACCESS=READ Anzugeben ist ACCESS=READ
Kleinbuchstaben	Kleinbuchstaben bezeichnen Variablen, die bei der Eingabe vom Benutzer durch aktuelle Werte ersetzt werden müssen, d.h. ihr Inhalt kann von Fall zu Fall verschieden sein.	RDPASS=kennwort Anzugeben ist z.B. RDPASS=C 'OTTO' oder RDPASS=54321 etc.
{ }	Geschweifte Klammern schließen Alternativen ein, d.h. aus den eingeschlossenen Größen muss eine Angabe ausgewählt werden. Ausnahme: Standardwerte	ACCESS={ READ } WRITE } Anzugeben ist ACCESS=READ oder ACCESS=WRITE
/	Der Schrägstrich trennt alternativ zu verwendende Angaben und hat dieselben Funktionen wie geschweifte Klammern.	WRITE=YES/NO Anzugeben ist WRITE=YES oder WRITE=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!)	bereich[,länge] Anzugeben ist z.B. BADR,60 oder BADR
<u>ABC..</u>	Die Unterstreichung hebt den Standardwert hervor. Das ist der Wert, den das System einsetzt, wenn der Benutzer keine Angabe macht (= Voreinstellung). Hat ein Operand keinen Standardwert, so ist die Angabe dieses Operanden Pflicht.	ESTATE={ <u>NEW</u> UPDATE }] Anzugeben ist: STATE=NEW oder STATE=UPDATE (keine Angabe, impliziert STATE=NEW)

Tabelle 29: Verwendete Metazeichen bei den Makrobeschreibungen (Teil 1 von 2)

Formale Darstellung	Erläuterung	Beispiel
...	Punkte bedeuten eine Wiederholung. Sie zeigen an, dass die davor stehende Einheit mehrmals hintereinander wiederholt werden kann.	(protect1,..4) Anzugeben ist z.B. (ACCESS) oder (ACCESS, EXDATE) etc.
□	Dieses Zeichen kennzeichnet ein Leerzeichen (X'40')	STD□ Anzugeben ist 'STD '

Tabelle 29: Verwendete Metazeichen bei den Makrobeschreibungen (Teil 2 von 2)

4.4 Beschreibung der Makros

CATJV Jobvariable katalogisieren

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **CATJV** erstellt oder ändert den Katalogeintrag einer Jobvariablen.

Das explizite Ausschalten der Default-Protection mit CATJV ...,PROTECT=STD wird nur mit gleichzeitiger Angabe des Operanden VERSION=4 unterstützt.

Die Verwendung von Makroversionen < 4 kann im Zusammenhang mit Default-Protection problematisch sein: Erhält eine Jobvariable über Default-Protection das Schutzmerkmal SHARE=YES oder ACCESS=READ oder Kennwörter, kann der nichtprivilegierte Anwender das Schutzmerkmal zwar auf SHARE=NO oder ACCESS=WRITE oder RDPASS bzw. WRPASS=NONE setzen. Diese Angaben werden aber nicht ausgewertet, da die Einstellungen der Default-Protection eine höhere Priorität haben.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
CATJV	jvname1 [,jvname2] [,STATE={ $\frac{\text{NEW}}{\text{UPDATE}}$ }] [,PROTECT= { $\frac{\text{DEFAULT}}{\text{STD}}$ }]

Operation	Operanden
CATJV (Forts.)	$[, ACCESS = \left\{ \begin{array}{l} \text{WRITE} \\ \text{READ} \end{array} \right\}]$ $[, SHARE = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}]$ $[, OWNERAR = \left\{ \begin{array}{l} \text{NO-ACCESS} \\ \left([\begin{array}{l} \text{READ} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \\ \text{R} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \end{array} \right] [, \left\{ \begin{array}{l} \text{WRITE} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \\ \text{W} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \end{array} \right\}] \right) \end{array} \right\}]$ $[, GROUPAR = \left\{ \begin{array}{l} \text{NO-ACCESS} \\ \left([\begin{array}{l} \text{READ} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \\ \text{R} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \end{array} \right] [, \left\{ \begin{array}{l} \text{WRITE} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \\ \text{W} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \end{array} \right\}] \right) \end{array} \right\}]$ $[, OTHERAR = \left\{ \begin{array}{l} \text{NO-ACCESS} \\ \left([\begin{array}{l} \text{READ} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \\ \text{R} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \end{array} \right] [, \left\{ \begin{array}{l} \text{WRITE} = \left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \\ \text{W} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \end{array} \right\}] \right) \end{array} \right\}]$ $[, BASACL = \left\{ \begin{array}{l} \text{NONE} \\ \text{STD} \end{array} \right\}]$

Operation	Operanden
CATJV (Forts.)	$\left[\text{,GUARDS} = \left\{ \begin{array}{l} \text{NONE} \\ \left(\left[\text{,READ} = \left\{ \begin{array}{l} * \text{NONE} \\ \text{readguard} \end{array} \right\} \right] \left[\text{,WRITE} = \left\{ \begin{array}{l} * \text{NONE} \\ \text{writeguard} \end{array} \right\} \right] \right) \end{array} \right\} \right]$ $\left[\text{,MANCLAS} = \left\{ \begin{array}{l} * \text{NONE} \\ \text{manclas} \end{array} \right\} \right]$ $\left[\text{,RDPASS} = \left\{ \begin{array}{l} \text{NONE} \\ \text{kennwort} \end{array} \right\} \right] \left[\text{,WRPASS} = \left\{ \begin{array}{l} \text{NONE} \\ \text{kennwort} \end{array} \right\} \right]$ $\left[\text{,RETPD} = \left\{ \begin{array}{l} 0 \\ \text{tage} \end{array} \right\} \right]$ $\left[\text{,MONJV} = \left\{ \begin{array}{l} \text{UNCHANGED} \\ \text{NO} \end{array} \right\} \right]$ $\left[\text{,VERSION} = \left\{ \begin{array}{l} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \right\} \right] \left[\text{,MF} = \left\{ \begin{array}{l} \text{S} \\ \text{C} \\ (\text{E}, \dots) \\ \text{D} \\ \text{L} \end{array} \right\} \right] \left[\text{,PREFIX} = \left\{ \begin{array}{l} \text{IDJ} \\ \text{pre} \end{array} \right\} \right]$

jvname1 bezeichnet den vollqualifizierten Pfadnamen, unter dem die permanente oder temporäre Jobvariable katalogisiert wird. Die Angabe einer fremden Benutzerkennung ist nur der Systembetreuung erlaubt.

jvname2 vereinbart einen neuen Namen für die Jobvariable jvname1. Bis Version < 2 dürfen Benutzer- und Katalogkennung nicht angegeben werden, da dies eine Änderung des Besitzers oder des Katalogs bedeuten könnte. Ab Version = 2 ist die Angabe eines vollqualifizierten Pfadnamens möglich, jedoch müssen Benutzer- und Katalogkennung mit den Angaben in jvname1 übereinstimmen. Dieser Operand ist nur zusammen mit STATE=UPDATE wirksam.

STATE	gibt an, ob für eine noch nicht vorhandene Jobvariable ein Katalogeintrag erstellt werden soll oder ein vorhandener Katalogeintrag geändert werden soll.
= <u>NEW</u>	ist Voreinstellung: Es soll ein Katalogeintrag erstellt werden.
=UPDATE	Ein vorhandener Katalogeintrag soll geändert werden.
	<i>Hinweis</i>
	Im UPDATE-Modus werden weggelassene Operanden nicht auf Standardwert gesetzt. Im UPDATE-Modus kann mit den Operanden ACCESS, SHARE, RDPASS, WRPASS und RETPD der entsprechende Wert im JV-Eintrag nur modifiziert werden, wenn er explizit angegeben ist.
PROTECT	gibt an, woher die Schutzmerkmale der neu zu katalogisierenden Jobvariable übernommen werden sollen, deren Wert nicht explizit angegeben wird. Die Angabe dieses Operanden ist nur zusammen mit VERSION=4 erlaubt.
	Zur Wertevergabe siehe Tabelle „System-Standardwerte für Jobvariablen-Schutzmerkmale“ auf Seite 37 .
= <u>DEFAULT</u>	ist Voreinstellung: Die Schutzmerkmale der neuen Jobvariable werden mit Werten der Funktion Default-Protection katalogisiert.
=STD	Die Schutzmerkmale der neuen Jobvariable werden mit System-Standardwerten katalogisiert.
ACCESS	legt fest, ob auf die Jobvariable nur lesend oder auch schreibend zugegriffen werden darf.
=WRITE	ist Voreinstellung bei STATE=NEW: Lese- und Schreibzugriff zur Jobvariablen sind erlaubt.
=READ	Die Jobvariable darf nur gelesen werden.

SHARE	bestimmt, ob die Jobvariable von einer fremden Benutzerkennung aus benutzt werden darf.
=NO	ist Voreinstellung bei STATE=NEW: Die Jobvariable darf von anderen Benutzerkennungen aus nicht benutzt werden.
=YES	Die Benutzung der Jobvariablen ist auch aus fremden Benutzerkennungen heraus erlaubt.
OWNERAR	bestimmt die Zugriffsrechte des Eigentümers der Jobvariablen (und der Systembetreuung). Lese- und Schreibberechtigung müssen jeweils explizit vergeben werden. Die Angabe dieses Operanden ist nicht zusammen mit VERSION=0 erlaubt.
=NO-ACCESS	Der Eigentümer besitzt explizit weder Lese- noch Schreibberechtigung. Dies entspricht der Angabe (READ=NO,WRITE=NO).
=(READ=...,WRITE=...)	Lese- und Schreibberechtigung werden - wie explizit angegeben - erteilt (READ/WRITE=YES) oder nicht erteilt (READ/WRITE=NO).
GROUPAR	bestimmt die Zugriffsrechte, die alle Benutzerkennungen aus der Gruppe des Eigentümers besitzen (außer Eigentümer und Systembetreuung). Die Angabe dieses Operanden ist nicht zusammen mit VERSION=0 erlaubt. Die Definition von Benutzergruppen ist erst bei Einsatz des Software-Produktes SECOS möglich. Im Hinblick auf den möglichen Einsatz von SECOS sollten für GROUP (Operand GROUPAR) die gleichen Rechte wie für OTHERS (Operand OTHERAR) vergeben werden.
=NO-ACCESS	Benutzerkennungen der Eigentümergruppe besitzen explizit weder Lese- noch Schreibberechtigung. Dies entspricht der Angabe (READ=NO,WRITE=NO).
=(READ=...,WRITE=...)	Lese- und Schreibberechtigung werden (wie explizit angegeben) erteilt (READ/WRITE=YES) oder nicht erteilt (READ/WRITE=NO).

OTHERAR	<p>Bestimmt die Zugriffsrechte der anderen Benutzer, die nicht der Eigentümergruppe angehören. Die Angabe dieses Operanden ist nicht zusammen mit VERSION=0 erlaubt.</p> <p>Falls SECOS nicht eingesetzt wird, sollten die Zugriffsrechte jedoch im Hinblick auf einen zukünftigen Einsatz von SECOS genauso wie für die Eigentümergruppe GROUP (Operand GROUPAR) gesetzt werden.</p>
=(READ=...,WRITE=...)	<p>Lese- und Schreibberechtigung werden (wie explizit angegeben) erteilt (READ/WRITE=YES) oder nicht erteilt (READ/WRITE=NO).</p>
BASACL	<p>bestimmt, ob für die Jobvariable eine BASIC-ACL gelöscht, aktiviert oder neu eingerichtet werden soll. Die Angabe dieses Operanden ist nicht zusammen mit VERSION=0 erlaubt.</p>
=NONE	<p>Eine aktivierte BASIC-ACL gelöscht. Bei Neueinrichten einer Jobvariablen (STATE=NEW) ist NONE Voreinstellung und bedeutet, dass keine BASIC-ACL vereinbart wird. Die Zugriffskontrolle erfolgt damit entsprechend der Werte ACCESS und SHARE (Standard-Zugriffskontrolle).</p>
=STD	<p>Beim Neueinrichten der Jobvariable (STATE=NEW) wird eine BASIC-ACL eingerichtet, in der für die Benutzergruppe OWNER Lese- und Schreibzugriff zugelassen wird und die Benutzergruppen GROUP und OTHERS keine Zugriffsrechte besitzen. Bei Änderung des Jobvariablen-Eintrags (STATE=UPDATE) wird die Angabe STD nur ausgewertet, wenn keine BASIC-ACL aktiviert war. In diesem Fall wird eine BASIC-ACL aktiviert, in der die Zugriffsrechte entsprechend der Standard-Zugriffskontrolle im Jobvariablen-Eintrag (SHARE und ACCESS) gesetzt sind:</p>

Standard-Zugriffskontrolle		BASIC-ACL-Schutz					
SHARE	ACCESS	OWNER		GROUP		OTHERS	
		R	W	R	W	R	W
NO	WRITE	Y	Y	N	N	N	N
NO	READ	Y	N	N	N	N	N
YES	WRITE	Y	Y	Y	Y	Y	Y
YES	READ	Y	N	Y	N	Y	N

Tabelle 30: Standard-Zugriffskontrolle/BASIC-ACL (CATJV-Makro)

GUARDS	bestimmt, ob für die Jobvariable ein Guard gelöscht, aktiviert oder neu eingerichtet werden soll. Die Angabe dieses Operanden ist nur ab VERSION=3 erlaubt.
=NONE	Ein früher definierter Guard wird aufgehoben. Bei Neueinrichten einer Jobvariablen (STATE=NEW) ist NONE Voreinstellung und bedeutet, dass kein Guard vereinbart wird.
=(READ=...,WRITE=...)	Die Zugriffskontrolle erfolgt über den angegebenen Lese- und/oder Schreibguard (READ/WRITE=readguard/writeguard) oder Lese- und/oder Schreibzugriff sind nicht erlaubt (READ/WRITE= <u>*NONE</u>). Der Name des Guards darf maximal 8 Zeichen lang sein, wenn keine Benutzerkennung angegeben wird. Eine Katalogkennung darf nicht angegeben werden.
MANCLAS	<i>Der Operand wird nur für SM-Pubsets ausgewertet.</i> Gibt an, ob die HSMS-Funktionen JV-Sicherung, Archivierung und Langzeitarchivierung über eine mit HSMS definierte Management-Klasse gesteuert werden. Näheres siehe Handbuch „HSMS“ [12]. Die Angabe dieses Operanden ist nur ab VERSION=3 erlaubt.
=*NONE	Eine früher vereinbarte Steuerung von HSMS-Funktionen über eine Management-Klasse wird aufgehoben. Bei Neueinrichten einer Jobvariablen (STATE=NEW) ist NONE Voreinstellung und bedeutet, dass keine Management-Klasse vereinbart ist.
=manclas	Name, der mit HSMS definierten Management-Klasse.
RDPASS	gibt an, ob ein Lesekennwort für die Jobvariable vereinbart und ein früher definiertes Lesekennwort aufgehoben wird.
=NONE	Ein früher definiertes Lesekennwort wird aufgehoben. Bei Neueinrichten einer Jobvariablen (STATE=NEW) ist NONE Voreinstellung und bedeutet, dass kein Lesekennwort vereinbart wird.
=kennwort	Definiert das Lesekennwort, das für den Zugriff auf die Jobvariable angegeben werden muss. Länge ≤ 4 Byte. Angabe als C-String, X-String oder Dezimalzahl. C-String: C'Zeichenkonstante'; maximal 4 Zeichen. X-String: X'Sedezimalkonstante'; maximal 8 Zeichen. Dezimalzahl: -2147483648 ≤ zahl ≤ 2147483647 Ein Kennwort X'00000000' bzw. '0' wird ignoriert. Ist eine Jobvariable nur durch ein Lesekennwort geschützt, muss dieses Kennwort auch zum Verändern der Jobvariablen angegeben werden.

WRPASS	gibt an, ob ein Schreibkennwort für die Jobvariable vereinbart und ein früher definiertes Schreibkennwort aufgehoben wird.
=NONE	Ein früher definiertes Schreibkennwort wird aufgehoben. Bei Neueinrichten einer Jobvariablen (STATE=NEW) ist NONE Voreinstellung und bedeutet, dass kein Schreibkennwort vereinbart wird.
=kennwort	Definiert das Schreibkennwort, das für einen Schreibzugriff auf die Jobvariable angegeben werden muss. Länge ≤ 4 Bytes. Angabe als C-String, X-String oder Dezimalzahl (siehe Operand RDPASS).
RETPD	legt die Dauer der Schutzfrist für die Jobvariable fest.
=0	ist Voreinstellung bei STATE=NEW: Die Dauer der Schutzfrist beträgt null Tage. Das bedeutet, dass im Jobvariablen-Eintrag als Expiration Date das aktuelle Tagesdatum steht (siehe Feld IDJEEEXD in der DSECT des Makros STAJV, Seite 180). Die Jobvariable kann verändert werden, wenn die anderen Schutzmerkmale einen schreibenden Zugriff zulassen.
=tage	tage = Anzahl der Tage, die die Jobvariable unverändert bleiben soll, auch wenn die anderen Schutzmerkmale schreibenden Zugriff zulassen. Das Expiration Date im Jobvariablen-Eintrag enthält das Datum, ab dem der Schutz aufgehoben ist. Expiration Date ist das aktuelle Tagesdatum plus angegebene Anzahl von Tagen. Dieser Operand wird nur berücksichtigt, wenn STATE=UPDATE angegeben ist, d.h. die Jobvariable muss bereits katalogisiert sein. Der Maximalwert für RETPD beträgt 32767 Tage.
MONJV	bestimmt, ob die Schutzmerkmale einer überwachenden Jobvariablen bestehen bleiben sollen.
= <u>UNCHANGED</u>	ist Voreinstellung: Der Schutz für eine auftragsüberwachende Jobvariable bleibt unverändert bestehen.
=NO	Hebt den Schutz des Systembereichs (Byte 1 bis 128) einer ehemaligen überwachenden Jobvariablen auf. Die Angabe ist nur wirksam mit STATE=UPDATE. Vor dem Aufruf ist sicherzustellen, dass der überwachte Auftrag tatsächlich aus der Warteschlange entfernt wurde (SHOW-JOB-STATUS).
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.

VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Der Operand MF=D/C zur Erzeugung einer DSECT bzw. CSECT wird dabei nicht unterstützt (siehe Hinweis DSECT).
=1	Die Makroauflösung entspricht dem Stand JV V10.0.
=2	Die Makroauflösung entspricht dem Stand JV V11.0 und V11.2.
=3	Die Makroauflösung entspricht dem Stand JV V12.0.
=4	Die Makroauflösung entspricht dem Stand $JV \geq V13.0C$.

Hinweise zur DSECT

- Der Aufruf des Makros mit den Operanden MF=D und VERSION=1/2/3/4 erzeugt eine DSECT für die Operandenliste des CATJV-Makros der entsprechenden Version.
- Eine DSECT für den Makro mit VERSION=0 wird mit Aufruf des Makros IDJCA [D][,prefix] erzeugt.

Hinweis zur Vergabe von Zugriffsrechten mit BASIC-ACL

BASIC-ACL wird aktiviert, wenn für mindestens einen Berechtigten BASIC-ACL-Rechte (im Operanden OWNERAR, GROUPAR oder OTHERAR) angegeben werden. Bei Aktivierung einer BASIC-ACL werden die Rechte für eine nicht angegebene Benutzergruppe wie bei der Angabe von BASIC-ACL=NO-ACCESS (weder Schreib- noch Lese-Recht) für diese Benutzergruppe gesetzt.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

CATJV    CATJV MF=D,VERSION=4
1 *****
1 *          VERSION 410
1 *****
1 *          C A T J V   P A R A M E T E R   L I S T          *
1 *****
1          #INTF REFTYPE=REQUEST,                               C
1          INTNAME=CATJV,INTCOMP=004
1 CATJV    DSECT
1 *****
1 *          UNIT=41, FUNCTION=4,  VERSION=<PARAMETER VERSION>  *
1 *****
1          FHDR    MF=(C, IDJC)

```

```

2          DS      0A
2 IDJCFHE DS      OXL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 IDJCIFID DS      0A          0  INTERFACE IDENTIFIER
2 IDJCFCTU DS      AL2         0  FUNCTION UNIT NUMBER
2 *
2 *                                     BIT 15  HEADER FLAG BIT,
2 *                                     MUST BE RESET UNTIL FURTHER NOTICE
2 *                                     BIT 14-12 UNUSED, MUST BE RESET
2 *                                     BIT 11-0  REAL FUNCTION UNIT NUMBER
2 IDJCFCT  DS      AL1          2  FUNCTION NUMBER
2 IDJCFCTV DS      AL1          3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJCRET  DS      0A           4  GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJCSRET DS      0AL2         4  SUB RETURN CODE
2 IDJCSR2  DS      AL1          4  SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJCR2OK EQU  X'00'           All correct, no additional info
2 IDJCR2NA EQU  X'01'           Successful, no action was necessary
2 IDJCR2WA EQU  X'02'           Warning, particular situation
2 IDJCSR1  DS      AL1          5  SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A      X'00'           FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B      X'01' - X'1F'   PARAMETER SYNTAX ERROR
2 * CLASS C      X'20'           INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D      X'40' - X'7F'   NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E      X'80' - X'82'   WAIT AND RETRY
2 *
2 IDJCRFSP EQU  X'00'           FUNCTION SUCCESSFULLY PROCESSED
2 IDJCRPER EQU  X'01'           PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJCRFNS EQU  X'01'           CALLED FUNCTION NOT SUPPORTED
2 IDJCRFNA EQU  X'02'           CALLED FUNCTION NOT AVAILABLE
2 IDJCRVNA EQU  X'03'           INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJCRAER EQU  X'04'           ALIGNMENT ERROR
2 IDJCRIER EQU  X'20'           INTERNAL ERROR
2 IDJCRCAR EQU  X'40'           CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJCRECR EQU  X'41'           SUBSYSTEM (SS) MUST BE CREATED
2 *                                     EXPLICITELY BY CREATE-SS
2 IDJCRECN EQU  X'42'           SS MUST BE EXPLICITELY CONNECTED

```

```

2 *
2 IDJCRWAR EQU X'80'          WAIT FOR A SHORT TIME AND RETRY
2 IDJCRWLR EQU X'81'          "      LONG      "
2 IDJCRWUR EQU X'82'          WAIT TIME IS UNCALCULABLY LONG
2 *                          BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJCRTNA EQU X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJCRDH EQU X'82'          SS IN DELETE / HOLD
2 *
2 IDJCMRET DS OAL2           6 MAIN RETURN CODE
2 IDJCMR2 DS AL1            6 MAIN RETURN CODE 2
2 IDJCMR1 DS AL1            7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XYYYYY')
2 *
2 IDJCRLNK EQU X'FFFF'       LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJCFHL EQU 8              8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL CATJV PARAMETER LIST *
1 *****
1 IDJCHDRI EQU X'00290404',4
1 IDJRD PAS DS CL4           READ PASSWORD
1 IDJWR PAS DS CL4           WRITE PASSWORD
1 IDJRETPD DS H              RETENTION PERIOD
1 IDJCFLAG DS X              FLAGS
1 IDJSTATE EQU X'80'         7-7 1=UPDATE, 0=NEW (STATE)
1 IDJACCES EQU X'40'         6-6 1=READ, 0=WRITE (ACCESS)
1 IDJSHARE EQU X'20'         5-5 1=YES, 0=NO (SHARE)
1 IDJRPN EQU X'10'           4-4 1=RD PASS NULL, STATE=U
1 IDJWPN EQU X'08'           3-3 1=WR PASS NULL, STATE=U
1 IDJRETPN EQU X'04'         2-2 1=RETPD NULL, STATE=U /
1 *                          RETPD NOT NULL, STATE=NEW
1 IDJACCEN EQU X'02'         1-1 1=ACCESS NULL, STATE=U
1 IDJSHARN EQU X'01'         0-0 1=SHARE NULL, STATE=U
1 *
1 IDJCFLG1 DS X              FLAGS
1 IDJCENCR EQU X'80'         7-7 0=YES, 1=NO
1 *                          (ENCRYPTION)
1 IDJTYPE EQU X'00'         6-6 NOT USED (DEL. V11.2)
1 IDJMONJV EQU X'20'         5-5 1=MONJV=NO, STATE=UPDATE
1 IDJCP2 EQU X'10'           4-4 1=P2 CALLER, 0=P1 CALLER
1 IDJCECT EQU X'08'         3-3 1=SET BY CMD PROCESSING
1 IDJBACLN EQU X'04'         2-2 1=SET BASIC-ACL = NONE
1 IDJBACLS EQU X'02'         1-1 1=SET BASIC-ACL = STD
1 IDJNSTEX EQU X'01'         0-0 1=SET NOSTEP=EXISTING

```

```

1 *
1 IDJJAR      DS    OX          ACCESS RIGHTS
1 IDJOWNER    DS    X          OWNER
1 IDJGROUP    DS    X          GROUP
1 IDJOTHER    DS    X          OTHERS
1 *
1 IDJJAUS     EQU   X'80'      7-7 1=USER CLASS SPECIFIED
1 IDJJARS     EQU   X'40'      6-6 1=READ   SPECIFIED
1 IDJJAWS     EQU   X'20'      5-5 1=WRITE  SPECIFIED
1 IDJJARO     EQU   X'08'      3-3 1=SET   READ
1 IDJJAWO     EQU   X'04'      2-2 1=SET   WRITE
1 *
1 IDJCFLG2    DS    X          FLAGS
1 IDJGRDN     EQU   X'80'      7-7 :S: GUARDS = *NONE
1 IDJPVSS     EQU   X'10'      4-4 :S: GUARD PUBSET SPECIFIED
1 IDJRDGS     EQU   X'08'      3-3 :S: READ  GUARD SPECIFIED
1 IDJWRGS     EQU   X'04'      2-2 :S: WRITE GUARD SPECIFIED
1 IDJMANSP    EQU   X'02'      1-1 :S: MANCLAS SPECIFIED
1 *
1 IDJPRFLG    DS    X          PROTECT FLAG
1 IDJPRNSP    EQU   X'80'      7-7 :S: PROTECT NOT SPEC(DEFAULT)
1 IDJPRSTD    EQU   X'40'      6-6 :S: PROTECT STD SPECIFIED
1 IDJACCSP    EQU   X'08'      3-3 :S: ACCESS SPECIFIED
1 IDJSHASP    EQU   X'04'      2-2 :S: SHARE  SPECIFIED
1 IDJRDSPSP   EQU   X'02'      1-1 :S: RDPASS SPECIFIED
1 IDJWRPSP    EQU   X'01'      0-0 :S: WRPASS SPECIFIED
1          DS    X          RESERVED
1 *
1 IDJRDG      DS    CL18      READ  GUARD
1 IDJWRG      DS    CL18      WRITE GUARD
1 IDJPUBS     DS    CL4       GUARD PUBSET
1 IDJMANCL    DS    CL8       MANCLAS
1 *
1 IDJJV1      DS    CL54      JVNAME
1 IDJJV2      DS    CL54      RENAME JVNAME
1          DS    CL54      RESERVED
1          DS    A          RESERVED
1          DS    CL16      RESERVED
1 IDJCJVS     DS    A          RESERVED
1 IDJPLLEN    EQU   *-CATJV   LENGTH
1 *****
1          SPACE

```

COPJV

Jobvariable kopieren

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro COPJV kopiert den Inhalt einer Jobvariablen (Sende-JV) in eine andere Jobvariable (Empfangs-JV). Wahlweise können neben dem Inhalt auch die Schutzmerkmale der Sende-JV mit Ausnahme eines bestehenden MONJV- bzw. CJC-Schutzes kopiert werden.

Es können sowohl permanente als auch temporäre Jobvariablen kopiert werden. Für die Sende-JV muss Lesezugriff und für die Empfangs-JV Schreibzugriff erlaubt sein. Eine nicht existierende Empfangs-JV wird neu angelegt, wobei beim Anlegen einer permanenten JV die im Benutzereintrag maximal erlaubte Anzahl nicht überschritten werden kann.

Privilegierte Funktionen

Die Systembetreuung (Privileg TSOS) ist standardmäßig Mit-Eigentümer aller Jobvariablen (kann somit also auch Jobvariablen unter jeder Benutzerkennung anlegen bzw. kopieren).

Diese Mit-Eigentümerschaft kann für permanente Jobvariablen bei Einsatz von SECOS eingeschränkt werden.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
COPJV	<p><jvname1>,<jvname2></p> <p>,SAME=$\left\{ \begin{array}{l} *NO \\ *YES \end{array} \right\}$</p> <p>,WRITE=$\left\{ \begin{array}{l} *REPLACE \\ *NEW \end{array} \right\}$</p> <p>,MF=$\left\{ \begin{array}{l} S \\ L \\ D \\ C \\ (E, (1)) \\ (E, (<r>)) \\ (E, <re1exp>) \end{array} \right\}$,PREFIX=$\left\{ \begin{array}{l} IDJ \\ pre \end{array} \right\}$</p>

<jvname1>

Name der Jobvariable, die zu kopieren ist (Sende-JV).

Der Lesezugriff muss erlaubt sein (bei einer JV unter fremder Benutzererkennung muss entweder USER-ACCESS=ALL-USERS oder Leserecht über BASIC-ACL oder GUARDS bzw. Miteigentümerschaft bestehen).

<jvname2>

Name der Jobvariable, in die kopiert werden soll (Empfangs-JV).

Ist die Empfangs-JV noch nicht katalogisiert, wird sie neu angelegt. In diesem Fall darf nur die eigene Benutzererkennung oder eine Benutzererkennung, für die der Benutzer Miteigentümer ist, angegeben werden.

Ist die Empfangs-JV katalogisiert, muss der Schreibzugriff erlaubt sein (bei einer JV unter fremder Benutzererkennung muss entweder Standardzugriffkontrolle mit USER-ACCESS=*ALL-USERS oder Schreibrecht über BASIC-ACL oder GUARDS bzw. Miteigentümerschaft bestehen).

Die Empfangs-JV wird jedoch nur bei Angabe von REPLACE=*YES (ist Voreinstellung) überschrieben.

SAME	Gibt an, ob die Schutzmerkmale der Sende-JV auch für die Empfangs-JV gelten sollen.
=*NO	Die Schutzmerkmale werden nicht in die Empfangs-JV übernommen. Bei einer neu zu erstellenden Empfangs-JV werden bei den Schutzmerkmalen die System-Standardwerte gesetzt (s.a. Voreinstellungen des Kommandos CREATE-JV). Für eine bereits bestehende Empfangs-JV bleiben die bisherigen Schutzmerkmale erhalten.
=*YES	Die Empfangs-JV erhält die gleichen Schutzmerkmale wie die Sende-JV (bzgl. ACCESS, USER-ACCESS, OWNER, GROUP, OTHERS, EXPIR-DATE, EXPIR-TIME, MAN-CLASS, vereinbarte GUARDS, sowie die gleichen Kennwörter; siehe auch Ausgabefelder des Kommandos SHOW-JV-ATTRIBUTES). Ein bestehender MONJV- bzw. CJC-Schutz wird jedoch nicht übernommen. Die Angabe SAME=*YES wird in folgenden Fällen ignoriert (d.h. es gilt *NO): <ul style="list-style-type: none"> – Die Empfangs-JV ist eine temporäre JV. – Die Empfangs-JV wird von CJC verwendet. – Die Empfangs-JV liegt unter einer fremden Benutzerkennung und der Aufrufer ist nicht Miteigentümer. Liegt die Sende-JV auf einer fremden Userid und ist mit BASIC-ACL oder GUARDS geschützt, werden die Merkmale USER-ACCESS, BASIC-ACL und GUARDS der Empfangs-JV auf Standardwerte gesetzt.
WRITE	Gibt an, ob eine bereits existierende Empfangs-JV überschrieben werden soll.
=*REPLACE	Eine bereits existierende Empfangs-JV wird ohne Meldung überschrieben.
= *NEW	Eine bereits existierende Empfangs-JV wird nicht überschrieben. Der Aufruf wird mit einem Fehler (Maincode X'0444') abgewiesen.
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

COPJV    COPJV MF=D
1 *****
1 *      VERSION 410
1 *****
1 *      C O P J V    P A R A M E T E R    L I S T          *
1 *****
1          #INTF REFTYPE=REQUEST,                          C
1              INTNAME=COPJV,INTCOMP=001
1 COPJV    DSECT
1 *****
1 *      UNIT=41, FUNCTION=35, VERSION=1                    *
1 *****
1          FHDR    MF=(C,IDJP)
2          DS      OA
2 IDJPFHE DS    OXL8          0    GENERAL PARAMETER AREA HEADER
2 *
2 IDJPIFID DS    OA          0    INTERFACE IDENTIFIER
2 IDJPFCTU DS    AL2         0    FUNCTION UNIT NUMBER
2 *
2 *          BIT 15    HEADER FLAG BIT,
2 *          MUST BE RESET UNTIL FURTHER NOTICE
2 *          BIT 14-12 UNUSED, MUST BE RESET
2 *          BIT 11-0  REAL FUNCTION UNIT NUMBER
2 IDJPFCT  DS    AL1         2    FUNCTION NUMBER
2 IDJPFCTV DS    AL1         3    FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJPRET  DS    OA          4    GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJPSRET DS    OAL2         4    SUB RETURN CODE
2 IDJPSR2  DS    AL1         4    SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJPR2OK EQU  X'00'          All correct, no additional info
2 IDJPR2NA EQU  X'01'          Successful, no action was necessary
2 IDJPR2WA EQU  X'02'          Warning, particular situation
2 IDJPSR1  DS    AL1         5    SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'          FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'  PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'          INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'  NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'  WAIT AND RETRY

```

```

2 *
2 IDJPRFSP EQU X'00' FUNCTION SUCCESSFULLY PROCESSED
2 IDJPRPER EQU X'01' PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJPRFNS EQU X'01' CALLED FUNCTION NOT SUPPORTED
2 IDJPRFNA EQU X'02' CALLED FUNCTION NOT AVAILABLE
2 IDJPRVNA EQU X'03' INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJPRAER EQU X'04' ALIGNMENT ERROR
2 IDJPRIER EQU X'20' INTERNAL ERROR
2 IDJPCAR EQU X'40' CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJPRECR EQU X'41' SUBSYSTEM (SS) MUST BE CREATED
2 * EXPLICITELY BY CREATE-SS
2 IDJPRECN EQU X'42' SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJPRWAR EQU X'80' WAIT FOR A SHORT TIME AND RETRY
2 IDJPRWLR EQU X'81' " LONG "
2 IDJPRWUR EQU X'82' WAIT TIME IS UNCALCULABLY LONG
2 * BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJPRTNA EQU X'81' SS TEMPORARILY NOT AVAILABLE
2 IDJPRDH EQU X'82' SS IN DELETE / HOLD
2 *
2 IDJPMRET DS OAL2 6 MAIN RETURN CODE
2 IDJPMR2 DS AL1 6 MAIN RETURN CODE 2
2 IDJPMR1 DS AL1 7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XYYYY')
2 *
2 IDJPRLNK EQU X'FFFF' LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJPFHL EQU 8 8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL COPJV PARAMETER LIST *
1 *****
1 IDJPHDRI EQU X'00292301',4
1 IDJPFGL DS XL1 FLAG
1 IDJPSAME EQU X'80' 7-7 1=SAME=YES 0=NO
1 IDJPWRT EQU X'40' 6-6 1=WRITE=NEW 0=REPLACE
1 * 0=DESCRIPTOR GIVEN
1 IDJPRES DS XL3 RESERVED
1 IDJPJV1 DS CL54 JVNAME1
1 IDJPJV2 DS CL54 JVNAME2
1 IDJPJVS DS A RESERVED
1 IDJPPLLN EQU *-COPJV LENGTH OF DSECT
1 *****
1 SPACE

```

CSWJV

Jobvariable prüfen und setzen

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Mit dem Makro **CSWJV** kann auf Programmebene ein bedingtes Setzen einer Jobvariablen durchgeführt werden (siehe auch Kommando MODIFY-JV-CONDITIONALLY).

Eine angegebene Jobvariable wird mit einem Feldinhalt verglichen. Bei Gleichheit wird die Jobvariable mit dem Inhalt eines anderen Feldes überschrieben, bei Ungleichheit wird der Jobvariablen-Wert in das Vergleichsfeld übertragen.

Die zu prüfende Jobvariable ist während der Bearbeitung durch den Makro gegen Zugriffe anderer Aufträge geschützt.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
CSWJV	$\left\{ \begin{array}{l} \text{jvid} \\ (\text{jvid} [\text{,start} [\text{,länge}]]) \end{array} \right\}$ <p>,bereich1, bereich2</p> <p>[,PASS=kennwort]</p> $,\text{VERSION}=\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\} ,\text{MF}=\left\{ \begin{array}{l} \text{S} \\ \text{C} \\ (\text{E}, \dots) \\ \text{D} \\ \text{L} \end{array} \right\} ,\text{PREFIX}=\left\{ \begin{array}{l} \text{IDJ} \\ \text{pre} \end{array} \right\}$

jvid	<p>identifiziert die zu prüfende oder zu setzende Jobvariable. jvid kann sein:</p> <p>jvname ein vollqualifizierter Pfadname einer permanenten oder temporären Jobvariablen</p> <p>*jvlink ein gültiger Jobvariablen-Kettungsname</p>		
start	<p>gibt die Position des ersten zu prüfenden oder zu setzenden Bytes innerhalb des Jobvariablen-Wertes an. Es gilt: $1 \leq \text{start} \leq 256$. Standardwert = 1.</p>		
länge	<p>legt die Anzahl der zu prüfenden oder zu setzenden Bytes des Jobvariablen-Wertes fest. Die Summe aus start und länge darf höchstens 257 betragen. Ist länge nicht gegeben, gelten die in den jeweiligen Satzlängenfeldern von bereich1 und bereich2 eingetragenen Längen.</p>		
bereich1	<p>symbolische Adresse des Vergleichsfeldes im Benutzerprogramm.</p> <p>Format</p> <table border="1" style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">1111xxxx</td> <td style="padding: 2px 10px;">vergleichswert</td> </tr> </table> <p style="margin-left: 40px;"><-4Byte-></p> <p style="margin-left: 40px;"><- 1111=Gesamtlänge -----></p> <p>Der aktuelle Wert der Jobvariablen jvid wird mit vergleichswert verglichen. Bei Ungleichheit wird vergleichswert mit dem Jobvariablen-Wert überschrieben (Returncode 0456 bzw. Meldung JVS0456).</p>	1111xxxx	vergleichswert
1111xxxx	vergleichswert		
bereich2	<p>symbolische Adresse des Setzwertes der Jobvariablen.</p> <p>Format</p> <table border="1" style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">1111xxxx</td> <td style="padding: 2px 10px;">setzwert</td> </tr> </table> <p style="margin-left: 40px;"><-4Byte-></p> <p style="margin-left: 40px;"><- 1111=Gesamtlänge -----></p> <p>Ist der Jobvariablen-Wert von jvid gleich dem Vergleichswert in bereich1, wird der Jobvariablen-Wert durch setzwert überschrieben.</p>	1111xxxx	setzwert
1111xxxx	setzwert		

PASS=kennwort	ermöglicht die Eingabe des zum Schreibzugriff auf die Jobvariable erforderlichen Kennworts (siehe Makro CATJV). Ist die Jobvariable kennwortgeschützt und der Operand PASS nicht gegeben, muss das Kennwort mit dem Kommando ADD-PASSWORD (z.B. über den CMD -Makro) vor dem ersten Aufruf von CSWJV dem System bekannt gemacht werden.
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.
VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Der Operand MF=D/C zur Erzeugung einer DSECT/CSECT wird dabei nicht unterstützt (siehe Hinweis DSECT).
=1	Die Makroauflösung entspricht $JV \geq V10.0$.

Hinweise zur DSECT

- Der Aufruf des Makros CSWJV mit den Operanden MF=D und VERSION=1 erzeugt eine DSECT für die Operandenliste des CSWJV-Makros (VERSION=1).
- Eine DSECT für den Makro mit VERSION=0 wird mit Aufruf des Makros IDJCS [D][,prefix] erzeugt.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

CSWJV   CSWJV MF=D,VERSION=1
1 *****
1 *           VERSION 203
1 *****
1 *           C S W J V   P A R A M E T E R   L I S T           *
1 *****
1           #INTF REFTYPE=REQUEST,                               C
1           INTNAME=CSWJV,INTCOMP=001
1 CSWJV   DSECT
1 *****
1 *           UNIT=41, FUNCTION=5,   VERSION=<PARAMETER VERSION>   *
1 *****
1           FHDR   MF=(C, IDJW)

```

```

2          DS      0A
2 IDJWFHE DS      OXL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 IDJWIFID DS      0A          0  INTERFACE IDENTIFIER
2 IDJWFCTU DS      AL2          0  FUNCTION UNIT NUMBER
2 *
2 *                                BIT 15  HEADER FLAG BIT,
2 *                                MUST BE RESET UNTIL FURTHER NOTICE
2 *                                BIT 14-12 UNUSED, MUST BE RESET
2 *                                BIT 11-0  REAL FUNCTION UNIT NUMBER
2 IDJWFCT  DS      AL1          2  FUNCTION NUMBER
2 IDJWFCTV DS      AL1          3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJWRET  DS      0A          4  GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJWSRET DS      0AL2          4  SUB RETURN CODE
2 IDJWSR2  DS      AL1          4  SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJWR2OK EQU  X'00'          All correct, no additional info
2 IDJWR2NA EQU  X'01'          Successful, no action was necessary
2 IDJWR2WA EQU  X'02'          Warning, particular situation
2 IDJWSR1  DS      AL1          5  SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A      X'00'          FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B      X'01' - X'1F'  PARAMETER SYNTAX ERROR
2 * CLASS C      X'20'          INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D      X'40' - X'7F'  NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E      X'80' - X'82'  WAIT AND RETRY
2 *
2 IDJWRFSP EQU  X'00'          FUNCTION SUCCESSFULLY PROCESSED
2 IDJWRPER EQU  X'01'          PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJWRFNS EQU  X'01'          CALLED FUNCTION NOT SUPPORTED
2 IDJWRFNA EQU  X'02'          CALLED FUNCTION NOT AVAILABLE
2 IDJWRVNA EQU  X'03'          INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJWRAER EQU  X'04'          ALIGNMENT ERROR
2 IDJWRIER EQU  X'20'          INTERNAL ERROR
2 IDJWRCAR EQU  X'40'          CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJWRECR EQU  X'41'          SUBSYSTEM (SS) MUST BE CREATED
2 *                                EXPLICITELY BY CREATE-SS
2 IDJWRECN EQU  X'42'          SS MUST BE EXPLICITELY CONNECTED

```



```

2 *
2 IDJRRWAR EQU X'80'          WAIT FOR A SHORT TIME AND RETRY
2 IDJRWLR EQU X'81'          "      LONG      "
2 IDJRWUR EQU X'82'          WAIT TIME IS UNCALCULABLY LONG
2 *                          BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJRTNA EQU X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJRDH EQU X'82'          SS IN DELETE / HOLD
2 *
2 IDJWMRET DS OAL2          6 MAIN RETURN CODE
2 IDJWMR2 DS AL1           6 MAIN RETURN CODE 2
2 IDJWMR1 DS AL1           7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 IDJWRLNK EQU X'FFFF'      LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJWFHL EQU 8             8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL CSWJV PARAMETER LIST *
1 *****
1 IDJWHDR1 EQU X'00290501',4
1 IDJWAREA DS A             AREA 1 ADDRESS
1 IDJWARE2 DS A             AREA 2 ADDRESS
1 IDJWJV DS CL54           JVNNAME
1 IDJWPOS DS H             SUBSTRING START POSITION
1 IDJWLEN DS H             SUBSTRING LENGTH
1 IDJWPASS DS XL2          RESERVED
1 IDJWPASS DS CL4          PASSWORD
1 IDJWFLAG DS X            RESERVED
1 IDJWENCN EQU X'80'       7-7 0=YES, 1=NO
1 *                        (ENCRYPTION)
1 IDJWFGNV EQU X'00'       6-6 NOT USED (DEL. V12)
1 IDJWJVIX EQU X'20'       5-5 0=NO, 1=YES
1 *                        (JVID INDEXED (SUBSTRING) )
1 IDJWFEX EQU X'10'       4-4 0=NO, 1=YES
1 *                        (PASSWORD GIVEN)
1 IDJWP2 EQU X'08'        3-3 0=P1 CALLER,1=P2 CALLER
1 IDJWP2 DS XL3           RESERVED
1 IDJWJVS DS A            RESERVED
1 IDJWPLLN EQU *-CSWJV    LENGTH OF DSECT
1 *****
1 SPACE

```

DCLJV

Jobvariablen-Kettungsnamen definieren

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **DCLJV** ordnet einer Jobvariablen einen Kettungsnamen zu und erstellt für die Jobvariable - falls sie noch nicht vorhanden ist - einen Katalogeintrag.

Hinweise

- Die Zuordnung zwischen Jobvariablen-Kettungsnamen und Jobvariablen-Namen wird in der JV-LINK-Tabelle des Auftrags festgehalten.
- Pro Kettungsnamen kann nur ein Eintrag in der JV-LINK-Tabelle vorhanden sein, eine Jobvariable kann dagegen mehreren Kettungsnamen zugeordnet sein.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
DCLJV	jvname [,LINK=*jvlink] $,VERSION = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}, MF = \left\{ \begin{array}{c} S \\ C \\ (E, \dots) \\ D \\ L \end{array} \right\}, PREFIX = \left\{ \begin{array}{c} IDJ \\ pre \end{array} \right\}$

jvname	ist der vollqualifizierte Pfadname einer permanenten oder temporären Jobvariablen. Existiert die Jobvariable bereits, muss der Benutzer die Zugriffsberechtigung haben. Handelt es sich um eine neue Jobvariable, wird sie unter dem angegebenen Namen katalogisiert (entspricht dem Makroaufruf CATJV mit Standardwerten). In diesem Fall darf nur die Systembetreuung eine andere Benutzerkennung angeben.
LINK	bezeichnet den Kettungsnamen (erstes Zeichen ist ein „*“), unter dem die Jobvariable innerhalb des Auftrags ab sofort angesprochen werden kann. Existiert für den angegebenen Kettungsnamen bereits eine Zuordnung, so wird die alte Zuordnung des Kettungsnamens aufgehoben und durch die neue Zuordnung ersetzt.
=*jvlink	Kettungsname der Jobvariablen. Die maximale Länge des Kettungsnamens ist 8 Zeichen einschließlich „*“.
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.
VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Der Operand MF=D/C zur Erzeugung einer DSECT/CSECT wird dabei nicht unterstützt (siehe Hinweis DSECT).
=1	Die Makroauflösung entspricht $JV \geq V10.0$.

Hinweise zur DSECT

- Der Aufruf des Makros DCLJV mit den Operanden MF=D und VERSION=1 erzeugt eine DSECT für die Operandenliste des DCLJV-Makros (VERSION=1).
- Eine DSECT für den Makro mit VERSION=0 wird mit Aufruf des Makros IDJDC [D][,prefix] erzeugt.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

DCLJV    DCLJV MF=D,VERSION=1
1 *****
1 *      VERSION 203
1 *****
1 *      D C L J V    P A R A M E T E R    L I S T          *
1 *****
1          #INTF REFTYPE=REQUEST,                          C
1              INTNAME=DCLJV,INTCOMP=001
1 DCLJV    DSECT
1 *****
1 *      UNIT=41, FUNCTION=6,    VERSION=<PARAMETER VERSION>    *
1 *****
1          FHDR    MF=(C,IDJD)
2          DS      OA
2 IDJDFHE DS    OXL8          0    GENERAL PARAMETER AREA HEADER
2 *
2 IDJDIFID DS    OA          0    INTERFACE IDENTIFIER
2 IDJDFCTU DS    AL2         0    FUNCTION UNIT NUMBER
2 *
2 *                                BIT 15    HEADER FLAG BIT,
2 *                                MUST BE RESET UNTIL FURTHER NOTICE
2 *                                BIT 14-12 UNUSED, MUST BE RESET
2 *                                BIT 11-0    REAL FUNCTION UNIT NUMBER
2 IDJDFCT DS    AL1          2    FUNCTION NUMBER
2 IDJDFCTV DS    AL1         3    FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJDRET DS    OA          4    GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJDSRET DS    OAL2         4    SUB RETURN CODE
2 IDJDSR2 DS    AL1          4    SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJDR2OK EQU   X'00'          All correct, no additional info
2 IDJDR2NA EQU   X'01'          Successful, no action was necessary
2 IDJDR2WA EQU   X'02'          Warning, particular situation
2 IDJDSR1 DS    AL1          5    SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'          FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'  PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'          INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'  NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'  WAIT AND RETRY

```

```

2 *
2 IDJDRFSP EQU X'00'          FUNCTION SUCCESSFULLY PROCESSED
2 IDJDRPER EQU X'01'          PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJDRFNS EQU X'01'          CALLED FUNCTION NOT SUPPORTED
2 IDJDRFNA EQU X'02'          CALLED FUNCTION NOT AVAILABLE
2 IDJDRVNA EQU X'03'          INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJDRAER EQU X'04'          ALIGNMENT ERROR
2 IDJDRIER EQU X'20'          INTERNAL ERROR
2 IDJDRCAR EQU X'40'          CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJDRECR EQU X'41'          SUBSYSTEM (SS) MUST BE CREATED
2 *                               EXPLICITELY BY CREATE-SS
2 IDJDRECN EQU X'42'          SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJDRWAR EQU X'80'          WAIT FOR A SHORT TIME AND RETRY
2 IDJDRWLR EQU X'81'          "      LONG      "
2 IDJDRWUR EQU X'82'          WAIT TIME IS UNCALCULABLY LONG
2 *                               BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJDRTNA EQU X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJDRDH EQU X'82'          SS IN DELETE / HOLD
2 *
2 IDJDMRET DS OAL2           6 MAIN RETURN CODE
2 IDJDMR2 DS AL1            6 MAIN RETURN CODE 2
2 IDJDMR1 DS AL1            7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XYYYY')
2 *
2 IDJDRLNK EQU X'FFFF'          LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJDFHL EQU 8                8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL DCLJV PARAMETER LIST *
1 *****
1 IDJDHRI EQU X'00290601',4
1 IDJJVL DS CL8                JV LINKNAME
1 IDJJV DS CL54                JVNAME
1 IDJDFLG DS XL2                FLAG RESERVED
1 IDJJDVS DS A                  RESERVED
1 IDJDPLLN EQU *-DCLJV          LENGTH OF DSECT
1 *****
1 SPACE

```

DONEVT

Bedingung eines Jobvariablen-Ereignisses löschen

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ (bei Version=0): Standardform/D-/E-/L-Form
S-Typ (bei Version=1): Standardform/C-/D-/E-/L-/M-Form
siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)

Makrobeschreibung

Mit dem Makro **DONEVT** werden eine oder mehrere Bedingungen gelöscht, die zuvor durch einen **ONEVT**-Makroaufruf an eine mit dem **ENAEI**-Makro deklarierte Ereigniskennung gebunden wurden. Der Anwender kann auch alle zurzeit existierenden Bedingungen für Jobvariablen-Ereignisse löschen.

Der Anwender muss den **DONEVT**-Makro vor dem **DISEI**-Makro absetzen, d.h., eine Bedingung löschen, bevor er die Ereigniskennung dazu aufgibt, andernfalls ist mit fehlerhaften Ergebnissen zu rechnen.

Über folgende drei Wege können Bedingungen für Jobvariablen-Ereignisse gelöscht werden:

- Der ONEVT-Zähler (Operand COUNT) erreicht den Wert null.
- Das Programm wird beendet.
- Ein DONEVT-Makro löscht die Bedingung.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
DONEVT	$EIID = \left\{ \begin{array}{l} *ALL \\ adr \\ (r) \end{array} \right\}$ $\left. \begin{array}{l} ,VERSION = \underline{0} \\ ,MF = \left\{ \begin{array}{l} \underline{S} \\ (D, pre) \\ D \\ (E, \dots) \\ L \end{array} \right\} \\ ,PREFIX = \left\{ \begin{array}{l} \underline{DON} \\ pre \end{array} \right\} \end{array} \right\}$ $\left. \begin{array}{l} ,VERSION = \underline{1} \\ [,MF = \left\{ \begin{array}{l} \underline{S} \\ C \\ D \\ E \\ L \\ M \end{array} \right\} \\ [,PARAM = \left\{ \begin{array}{l} adr \\ (r) \end{array} \right\} \\ ,PREFIX = \left\{ \begin{array}{l} \underline{J} \\ pre \end{array} \right\} \\ ,MACID = \left\{ \begin{array}{l} \underline{VSC} \\ macid \end{array} \right\} \\ [,POST = \left\{ \begin{array}{l} wert \\ r \end{array} \right\} \end{array} \right\}$

EIID	bezeichnet die zu löschenden Bedingungen, die zuvor mit dem ONEVT-Makro für ein Jobvariablen-Ereignis gesetzt wurden.
=*ALL	alle noch existierenden Bedingungen für Jobvariablen-Ereignisse werden gelöscht.
=adr	symbolische Adresse der Ereigniskurzbezeichnung.
=(r)	r = Register, das die symbolische Adresse der Ereigniskurzbezeichnung enthält.
MF PREFIX MACID PARAM	Zur Beschreibung der Operanden MF, PREFIX, MACID und PARAM siehe Seite 88 . Die gültigen Werte und Voreinstellungen von MF für diesen Makro sind zu Beginn der Makrobildbeschreibung, für PREFIX und MACID bei der Beschreibung des Operanden VERSION angegeben.

VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Bei der D-Form des Makroaufrufs kann ein Präfix (pre=1..3 Buchstaben) angegeben werden. Voreinstellung: pre=DON
=1	Die Makroauflösung entspricht dem Stand $\geq JV V10.0$. Bei der C-Form, D-Form oder M-Form des Makroaufrufs kann ein Präfix PREFIX (p=1 Buchstabe) und bei der C-Form und M-Form zusätzlich eine Macid MACID angegeben werden.
POST	bezeichnet einen 2 Byte langen Wert, der auf Programmebene zusammen mit der Ereigniskurzennung zur Identifizierung des DONEVT verwendet werden kann und zuvor mit dem ONEVT-Makro gesetzt wurde. Die Angabe dieses Operanden ist nur ab VERSION=1 erlaubt.
=wert	2 Byte lange Assembler-Konstante beliebigen Formats.
=r	r = Register, das die Adresse des 2 Byte langen Feldes enthält, in dem der Benutzer den Wert hinterlegt hat.

Rückinformation und Fehleranzeigen

bei VERSION=0:

R15

b	b					a	a
---	---	--	--	--	--	---	---

 Über die Ausführung des Makros DONEVT (VERSION=0) wird im Register R15 ein Returncode übergeben:
 (aa = Rücksprungschalter, bb = Sekundärindikator)

X'bb'	X'aa'	Erläuterung
X'00'	X'00'	Ausführung normal beendet.
X'00'	X'04'	Funktion nicht ausgeführt: ungültige Adresse für die Ereigniskennung.
X'04'	X'04'	Funktion nicht ausgeführt: Ereigniskennung nicht gefunden.
X'08'	X'04'	Funktion nicht ausgeführt: Systemfehler (Speicherplatzverwaltung).
	X'FF'	Funktion nicht ausgeführt: CJC im System nicht verfügbar.

bei VERSION=1:

Nach Initialisierung des Standardheaders (bei Aufruf mit MF=S/L) wird die Rückkehrinformation an der symbolischen Adresse <PREFIX><MACID>RET zur Verfügung gestellt (4Byte).

Standard-
header

c	c	b	b	a	a	a	a
---	---	---	---	---	---	---	---

Über die Ausführung des Makros DONEVT (VERSION=1) wird im Standardheader ein Returncode übergeben:
(aaaa=MAINCODE,bb=SUBCODE1,cc=SUBCODE2)

X'cc'	X'bb'	X'aaaa'	Erläuterung
X'00'	X'00'	X'0000'	Ausführung normal beendet.
X'00'	X'01'	X'0004'	Funktion nicht ausgeführt: ungültige Adresse der Ereigniskennung.
X'04'	X'01'	X'0004'	Funktion nicht ausgeführt: Ereigniskennung nicht gefunden.
X'08'	X'20'	X'0004'	Funktion nicht ausgeführt: Systemfehler.
		X'FFFF'	Funktion nicht ausgeführt: Fehler bei der Initialisierung des Standardheaders, siehe Seite 229 .

Zusätzlich zu den bei VERSION=0 bzw. 1 genannten Returncodes können bei grundlegenden Fehlern (z.B. ungültige Adresse der Operandenliste) auch allgemeine JV-Returncodes auftreten. Deren Bedeutung kann dem Kommentar zum entsprechenden Returncode im Makro IDEJVS ([Seite 229](#)) entnommen werden.

DSECT

```

DONEVT  DONEVT MF=D,VERSION=1,PREFIX=A
1 DONEVT  MFCHK MF=D,                                     C
1          SUPPORT=(C,D,E,L,M,S),                       C
1          PREFIX=A,                                     C
1          MACID=VSC,                                    C
1          DMACID=VSC,                                   C
1          DNAME=DONPL,                                  C
1          PARAM=,                                       C
1          SVC=190
2 DONEVT  DSECT ,
2          *,##### PREFIX=A, MACID=VSC #####
1 AVSCFHDR  FHDR  MF=(C,AVSC),EQUATES=NO
2 AVSCFHDR DS    0A
2 AVSCFHE  DS    0XL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 AVSCIFID DS    0A          0  INTERFACE IDENTIFIER
2 AVSCFCTU DS    AL2          0  FUNCTION UNIT NUMBER
2 *                               BIT 15  HEADER FLAG BIT,
2 *                               MUST BE RESET UNTIL FURTHER NOTICE

```

2 *			BIT 14-12 UNUSED, MUST BE RESET
2 *			BIT 11-0 REAL FUNCTION UNIT NUMBER
2 AVSCFCT	DS	AL1	2 FUNCTION NUMBER
2 AVSCFCTV	DS	AL1	3 FUNCTION INTERFACE VERSION NUMBER
2 *			
2 AVSCRET	DS	0A	4 GENERAL RETURN CODE
2 AVSCSRET	DS	0AL2	4 SUB RETURN CODE
2 AVSCSR2	DS	AL1	4 SUB RETURN CODE 2
2 AVSCSR1	DS	AL1	5 SUB RETURN CODE 1
2 AVSCMRET	DS	0AL2	6 MAIN RETURN CODE
2 AVSCMR2	DS	AL1	6 MAIN RETURN CODE 2
2 AVSCMR1	DS	AL1	7 MAIN RETURN CODE 1
2 AVSCFHL	EQU	8	8 GENERAL OPERAND LIST HEADER LENGTH
2 *			
1 AVSCPCON	DS	X	PARAM.-CONTROL
1 AVSCPREG	EQU	X'40'	PARAM. IN REGISTER
1 AVSCXREG	EQU	X'20'	POST IN REGISTER
1 AVSCXSPE	EQU	X'04'	POST SPECIFIED
1 AVSCPTPR	EQU	X'02'	P2 CALLER
1 AVSCPKEY	EQU	X'01'	PARAM. IS KEYWORD
1 AVSCUNUD	DS	CL1	UNUSED
1 AVSCPOST	DS	H	POST-VALUE
1	ORG	AVSCPOST	
1 AVSCPOSR	DS	AL1	REG. CONT. POST-VALUE
1	DS	AL1	NOT USED IN THIS CONTEXT
1 AVSCEIID	DS	A	A(IDENTIFIER)
1	ORG	AVSCEIID	
1 AVSCREG	DS	X	REGISTER #
1 AVSCUNU2	DS	CL3	UNUSED
1 AVSC#	EQU	*-AVSCFHDR	LENGTH
1 *			
1 * RETURNCODES			
1 *			
1 AVSCOK	EQU	X'00000000'	SUCCESSFUL CALL
1 AVSCINAD	EQU	X'00010004'	INVALID PARM.-LIST
1 AVSCINEI	EQU	X'04010004'	INVALID IDENTIFIER
1 AVSCSYSE	EQU	X'08200004'	SYSTEM ERROR

ERAJV

Jobvariable löschen

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **ERAJV** löscht eine oder mehrere Jobvariablen-Einträge aus dem Dateikatalog oder nur den Jobvariablenwert (dabei wird die Jobvariablen-Länge auf Null zurückgesetzt). Der Name der Jobvariablen kann teilqualifiziert und mit Musterzeichen angegeben werden.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
ERAJV	<pre> jvid ,IGNORE={ NONE protect1 (protect1,...4) } ,CHECK={ STD NO MULTIPLE PVS SINGLE } ,PASS={ NONE kennwort1 (kennwort1,...3) } </pre>

Operation	Operanden
ERAJV (Forts.)	$,DATA = \left\{ \begin{array}{c} \text{NO} \\ \text{YES} \end{array} \right\}$ $,VERSION = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\} \quad ,MF = \left\{ \begin{array}{c} S \\ C \\ (E, \dots) \\ D \\ L \end{array} \right\} \quad ,PREFIX = \left\{ \begin{array}{c} \text{IDJ} \\ \text{pre} \end{array} \right\}$

jvid	<p>identifiziert die zu löschende Jobvariable. jvid kann sein:</p> <p>jvname ein voll- oder teilqualifizierter Pfadname einer permanenten oder temporären Jobvariablen. Die Verwendung von Musterzeichen ist zulässig. Die Angabe einer fremden Benutzerkennung oder einer Benutzerkennung mit Musterzeichen ist nur der Systembetreuung erlaubt. Der angegebene Pfadname muss mindestens aus einem der drei Teile bestehen: Katalogkennung, Benutzerkennung, Jobvariablen-Name.</p> <p>*jvlink ein gültiger Jobvariablen-Kettungsname.</p> <p>temp Alle temporären Jobvariablen des Auftrags werden gelöscht. temp steht für das mit dem Systemparameter TEMPFILe definierte Sonderzeichen zur Kennzeichnung von temporären Dateien und Jobvariablen (im Bedarfsfall die Systembetreuung fragen)</p>
IGNORE	<p>gibt an, ob beim Löschen Schutzmerkmale, die für die Jobvariable im Katalog hinterlegt wurden, ignoriert werden sollen. Der nicht-privilegierte Anwender kann die Schutzmerkmale ACCESS=READ, BASIC-ACL und EXDATE größer aktuelles Tagesdatum ignorieren. Der Operand wirkt sich nur bei Angabe von VERSION=1 aus.</p>

<u>=NONE</u>	ist Voreinstellung: Führt zu einer Fehlermeldung, falls versucht wird, eine Jobvariable zu löschen, die eine der genannten Schutzmerkmale besitzt.
<u>=(protect1,...4)</u> <u>=protect1</u>	bestimmt das zu übergehende Schutzrecht. Folgende Werte sind möglich als Einzelangabe oder als Liste: ACCESS: Die Schutzmerkmale ACCESS=READ und BASIC-ACL werden ignoriert. EXDATE: Ein Freigabedatum größer Tagesdatum wird ignoriert. RDPASS: <i>nur privilegierter Benutzer</i> Der Schutz mit einem Lesekennwort wird ignoriert. WRPASS: <i>nur privilegierter Benutzer</i> Der Schutz mit einem Schreibkennwort wird ignoriert.
CHECK	legt fest, ob auf das Löschen von Jobvariablen Einfluss genommen werden kann. Möglichkeit der Einflussnahme: Reaktion auf Systemabfrage, ob wirklich gelöscht werden soll. Dieser Operand wird im Batchmodus im Konfliktfall dynamisch angepasst, da im Batchmodus nur NO sinnvoll ist. Der Operand wirkt sich nur bei Angabe von VERSION=1 aus.
<u>=STD</u>	ist Voreinstellung: Legt als Standardwert fest: MULTIPLE für den Dialogmodus (Globalabfrage) und NO für den Prozedur-/Batchmodus (Löschen ohne Vorwarnung).
<u>=NO</u>	Das Löschen der Jobvariablen kann nicht zusätzlich beeinflusst werden.
<u>=MULTIPLE</u>	führt zu einer globalen Abfrage pro Benutzerkennung (Meldung: JVS0465), aber nur wenn mehr als eine Jobvariable zum Löschen ansteht. MULTIPLE ist im Dialog der voreingestellte Wert.
<u>=PVS</u>	führt zu einer globalen Abfrage pro Pubset (Meldung: JVS0468), aber nur wenn mehr als eine Jobvariable zum Löschen ansteht.

=SINGLE jede einzelne zu löschende Jobvariable wird abgefragt (Meldung: JVS0469).

Hinweis

Jede Meldung bei MULTIPLE, PVS und SINGLE wird durch folgende Antwortmöglichkeit ergänzt:

ANTWORT (Y=JA; N=NEIN; T=TERMINATE; [,CHECK=modus])

Antwort	Auswirkung
Y	Löschanforderung wird durchgeführt
N	Löschanforderung wird nicht durchgeführt, Meldung JVS046A
T	Das Löschen wird abgebrochen.
jede andere Antwort	Auswirkung wie „N“
optionaler Zusatz:	
CHECK=	Der CHECK-Modus wird nicht geändert.
CHECK=<modus>	Der CHECK-Modus (STD, NO, MULTIPLE, PVS oder SINGLE) wird eingestellt.

PASS ermöglicht für den Anwender das Löschen kennwortgeschützter Jobvariablen. Der Operand wirkt sich nur bei Angabe von VERSION=1 aus.
Hier genannte Kennwörter werden nicht in die Kennworttabelle des Auftrags eingetragen.

=NONE ist Voreinstellung: Untersagt das Löschen kennwortgeschützter Jobvariablen.

=(kennwort1,...3) nennt das Kennwort, das beim Löschen ignoriert werden soll. Als Listenangabe sind max. drei Kennwörter möglich.
=kennwort1

DATA bestimmt den Umfang des Löschens. Der Operand wirkt sich nur bei Angabe von VERSION=1 aus.

=NO ist Voreinstellung: Der Jobvariableneintrag wird gelöscht.

=YES der Wert der Jobvariablen wird auf Null zurückgesetzt.

MF	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 .
PREFIX	Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.
VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Der Operand MF=C/D zur Erzeugung einer CSECT bzw. DSECT wird dabei nicht unterstützt (siehe Hinweis DSECT).
=1	Die Makroauflösung entspricht dem Stand $JV \geq V10.0$.

Hinweise zur CSECT/DSECT

- Der Aufruf des Makros ERAJV mit den Operanden MF=D und VERSION=1 erzeugt eine DSECT für die Operandenliste des ERAJV-Makros (VERSION=1).
- Für den ERAJV-Makro mit VERSION=0 kann eine CSECT bzw. DSECT durch Aufruf des Makros IDJER erzeugt werden. Eine CSECT wird mit IDJER [,prefix] erzeugt, eine DSECT mit IDJER D[,prefix].

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

ERAJV    ERAJV MF=D,VERSION=1
1 *****
1 *          VERSION 400
1 *****
1 *          E R A J V    P A R A M E T E R    L I S T          *
1 *****
1          #INTF REFTYPE=REQUEST,                               C
1          INTNAME=ERAJV,INTCOMP=001
1 ERAJV    DSECT
1 *****
1 *          UNIT=41, FUNCTION=3,  VERSION=<PARAMETER VERSION>  *
1 *****
1          FHDR  MF=(C, IDJE)
2          DS   OA
2 IDJEFHE DS   OXL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 IDJEIFID DS   OA          0  INTERFACE IDENTIFIER
2 IDJEFCTU DS   AL2        0  FUNCTION UNIT NUMBER
2 *          BIT 15  HEADER FLAG BIT,

```

```

2 *                               MUST BE RESET UNTIL FURTHER NOTICE
2 *                               BIT 14-12 UNUSED, MUST BE RESET
2 *                               BIT 11-0 REAL FUNCTION UNIT NUMBER
2 IDJEFCT DS    AL1                2  FUNCTION NUMBER
2 IDJEFTV DS    AL1                3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJERET DS    0A                 4  GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJESRET DS    0AL2              4  SUB RETURN CODE
2 IDJESR2 DS    AL1                4  SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJER2OK EQU  X'00'              All correct, no additional info
2 IDJER2NA EQU  X'01'              Successful, no action was necessary
2 IDJER2WA EQU  X'02'              Warning, particular situation
2 IDJESR1 DS    AL1                5  SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'              FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'     PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'              INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'     NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'     WAIT AND RETRY
2 *
2 IDJERFSP EQU  X'00'              FUNCTION SUCCESSFULLY PROCESSED
2 IDJERPER EQU  X'01'              PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJERFNS EQU  X'01'              CALLED FUNCTION NOT SUPPORTED
2 IDJERFNA EQU  X'02'              CALLED FUNCTION NOT AVAILABLE
2 IDJERVNA EQU  X'03'              INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJERAER EQU  X'04'              ALIGNMENT ERROR
2 IDJERIER EQU  X'20'              INTERNAL ERROR
2 IDJERCAR EQU  X'40'              CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJERECR EQU  X'41'              SUBSYSTEM (SS) MUST BE CREATED
2 *                               EXPLICITELY BY CREATE-SS
2 IDJEREEN EQU  X'42'              SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJERWAR EQU  X'80'              WAIT FOR A SHORT TIME AND RETRY
2 IDJERWLR EQU  X'81'              "    LONG    "
2 IDJERWUR EQU  X'82'              WAIT TIME IS UNCALCULABLY LONG
2 *                               BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'

```



```

2 IDJERTNA EQU X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJERDH EQU X'82'          SS IN DELETE / HOLD
2 *
2 IDJEMRET DS OAL2           6 MAIN RETURN CODE
2 IDJEMR2 DS AL1             6 MAIN RETURN CODE 2
2 IDJEMR1 DS AL1             7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 IDJERLNK EQU X'FFFF'       LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJEFHL EQU 8               8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL ERAJV PARAMETER LIST *
1 *****
1 IDJEHDRI EQU X'00290301',4
1 IDJEJV DS CL80              JVNAME
1 DS CL40                     RESERVED
1 IDJEFLAG DS X              FLAGS
1 IDJEDATA EQU X'80'          7-7 DATA=YES
1 * SET JV-VALUE TO NULLSTRING
1 IDJEECT EQU X'40'          6-6 SET BY CMD PROCESSING
1 * (NOT USED BY MACROCALLER)
1 IDJELIST EQU X'20'          5-5 LIST=YES
1 * (NOT USED BY MACROCALLER)
1 IDJENPWT EQU X'10'          4-4 NO PASSWORD TEST
1 * (PRIV CALLER ONLY)
1 IDJESEVR EQU X'08'          3-3 SEVER PROCESSING
1 * (ONLY USED BY SEVER-CMD)
1 IDJETERM EQU X'04'          2-2 TERM. PROCESSING
1 * (ONLY USED BY LOGOFF-CMD)
1 IDJEENCR EQU X'02'          1-1 NO ENCRYPTION OF PASSWD
1 * REQUIRED
1 IDJENEWI EQU X'01'          0-0 NEW INTERFACE(V9.5 UP)
1 IDJECHK DS X                FLAGS FOR CHECK INDICATOR
1 IDJECNO EQU X'80'          7-7 CHECK=NO
1 * ALL JVS ARE DELETED - NO
1 * FEEDBACK TO CALLER
1 IDJECMUL EQU X'40'          6-6 CHECK=MULTIPLE
1 * IT IS ASKED IF ALL JVS OF
1 * THE CURRENT USERID SHOULD
1 * BE DELETED
1 IDJECPVS EQU X'20'          5-5 CHECK=PVS
1 * IT IS ASKED IF ALL JVS OF
1 * THE CURRENT PUBSET SHOULD
1 * BE DELETED
1 IDJECSIN EQU X'10'          4-4 CHECK=SINGLE
1 * IT IS ASKED FOR EACH JV IF

```

```

1 * THE THE JV SHOULD BE DELETED
1 IDJECDEF EQU X'08' 3-3 CHECK=STD
1 * DEFAULTS ARE USED
1 * DIALOG: MULTIPLE
1 * OTHERS: NO
1 IDJEIGNO DS X FLAGS FOR IGNORE PARAMETER
1 IDJEINON EQU X'80' 7-7 IGNORE=NONE
1 * ALL PROTECTIONS ARE CHECKED
1 * IF PROTECTION DEFINED ERASE
1 * IS REJECTED
1 IDJEIRDP EQU X'40' 6-6 IGNORE=RDPASS
1 * RDPASS PROTECTION IS
1 * IGNORED. JV IS DELETED
1 IDJEIWRP EQU X'20' 5-5 IGNORE=WRPASS
1 * WRPASS PROTECTION IS
1 * IGNORED. JV IS DELETED
1 IDJEIACC EQU X'10' 4-4 IGNORE=ACCESS
1 * ACCESS=READ IS IGNORED.
1 * JV IS DELETED
1 IDJEIEXD EQU X'08' 3-3 IGNORE=EXDATE
1 * IGNORE RETPD DEFINITION.
1 IDJEFLG1 DS X FLAG 1
1 IDJEP2 EQU X'80' 7-7 CALLER=P2
1 IDJNSTEP DS 6XL1 3 ERR'S
1 IDJERESE DS 14XL1 UNUSED
1 IDJEPWD1 DS CL4 PASSWORD1 SET DEFAULT 0
1 IDJEPWD2 DS CL4 PASSWORD2 SET DEFAULT 0
1 IDJEPWD3 DS CL4 PASSWORD3 SET DEFAULT 0
1 * THE FOLLOWING FIELD IS SUPPLIED ONLY FOR CMD PROCESSING
1 IDJEADDR DS CL4 SORT TABLE ADDRESS
1 IDJEJVS DS A RESERVED
1 IDJEPLLN EQU *-ERAJV LENGTH OF DSECT
1 *****
1 SPACE

```

GETJV

Wert ausgeben

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **GETJV** übergibt den Wert einer Benutzer- oder Sonder-Jobvariablen in einen Bereich des Benutzerprogramms.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
GETJV	$\left\{ \begin{array}{l} \text{jvid} \\ (\text{jvid } [, \text{start } [, \text{länge}]]) \end{array} \right\} , \text{bereich, grÖÙe}$ <p>[, PASS=kennwort]</p> $, \text{PARMOD} = \left\{ \begin{array}{l} 24 \\ 31 \end{array} \right\}$ $, \text{VERSION} = \left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\} , \text{MF} = \left\{ \begin{array}{l} S \\ C \\ (E, \dots) \\ D \\ L \end{array} \right\} , \text{PREFIX} = \left\{ \begin{array}{l} \text{IDJ} \\ \text{pre} \end{array} \right\}$

jvid	identifiziert die Jobvariable. jvid kann sein:
jvname	Vollqualifizierter Pfadname einer permanenten oder temporären Jobvariablen.
*jvlink	Gültiger Jobvariablen-Kettungsname.

start	Startposition für Ausgabe.		
länge	Ausgabelänge.		
bereich	<p>Adresse eines Bereichs im Benutzerprogramm, in den der Wert der Jobvariablen übertragen werden soll. Versorgung des Bereichs:</p> <p>Format <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">1111xxxx</td><td style="padding: 2px;">JV-Wert</td></tr></table></p> <p style="margin-left: 100px;"><-4Byte-></p> <p style="margin-left: 100px;"><- 1111=Gesamtlänge -----></p>	1111xxxx	JV-Wert
1111xxxx	JV-Wert		
größe	<p>gibt die Größe von „bereich“ an. Sie sollte mindestens „Länge des zu lesenden Wertes + 4“ betragen und darf nicht größer als 32767 sein. Die tatsächliche genutzte Größe von „bereich“, d.h. die Länge des JV-Werts +4, wird in die ersten 2 Bytes des Bereichs eingetragen. Ist die Gesamtlänge des Jobvariablen-Wertes größer als der Maximalwert „größe - 4“, so wird der Jobvariablen-Wert abgeschnitten, sodass dieser Maximalwert eingehalten wird.</p>		
PASS=kennwort	Lesekennwort.		
PARMOD	<p>steuert die Makroauflösung. Es wird entweder die 24-Bit- oder die 31-Bit-Schnittstelle generiert. PARMOD wird nur mit VERSION=0 ausgewertet. Wenn PARMOD nicht spezifiziert wird, erfolgt die Makroauflösung entsprechend der Angabe für den Makro GPARMOD oder der Voreinstellung für den Assembler (= 24-Bit-Adressierung).</p>		
<u>=24</u>	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 PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.		

VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Der Operand MF=D/C zur Erzeugung einer DSECT bzw. CSECT wird dabei nicht unterstützt (siehe Hinweis DSECT).
=1	Die Makroauflösung entspricht $JV \geq V10.0$.

Hinweis

Im Gegensatz zur Kommandoebene darf in der Operandenliste die Länge 0 spezifiziert werden, die dann die gesamte Jobvariablen-Länge repräsentiert.

Hinweise zur DSECT

- Der Aufruf des Makros GETJV mit den Operanden MF=D und VERSION=1 erzeugt eine DSECT für die Operandenliste des GETJV-Makros (VERSION=1).
- Eine CSECT/DSECT für den Makro mit VERSION=0 wird mit Aufruf des Makros IDJGE [D][,prefix] [,PARMOD=24/31] erzeugt.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

          GETJV   GETJV MF=D,VERSION=1
GETJV   GETJV MF=D,VERSION=1
1 *****
1 *          VERSION 203
1 *****
1 *          G E T J V   P A R A M E T E R   L I S T          *
1 *****
1          #INTF REFTYPE=REQUEST,                               C
1          INTNAME=GETJV,INTCOMP=002
1 GETJV   DSECT
1 *****
1 *          UNIT=41, FUNCTION=0,   VERSION=1 (V9.0)          *
1 *          VERSION=2 (V10.0)    *
1 *****
1          FHDR   MF=(C,IDJG)
2          DS    OA
2 IDJGFHE DS    OXL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 IDJGIFID DS    OA          0  INTERFACE IDENTIFIER
2 IDJGFCTU DS    AL2        0  FUNCTION UNIT NUMBER
2 *          BIT 15  HEADER FLAG BIT,

```

```

2 *                               MUST BE RESET UNTIL FURTHER NOTICE
2 *                               BIT 14-12 UNUSED, MUST BE RESET
2 *                               BIT 11-0 REAL FUNCTION UNIT NUMBER
2 IDJGFCT DS AL1 2 FUNCTION NUMBER
2 IDJGFCTV DS AL1 3 FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJGRET DS OA 4 GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJGSRET DS OAL2 4 SUB RETURN CODE
2 IDJGSR2 DS AL1 4 SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJGR2OK EQU X'00' All correct, no additional info
2 IDJGR2NA EQU X'01' Successful, no action was necessary
2 IDJGR2WA EQU X'02' Warning, particular situation
2 IDJGSR1 DS AL1 5 SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A X'00' FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B X'01' - X'1F' PARAMETER SYNTAX ERROR
2 * CLASS C X'20' INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D X'40' - X'7F' NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E X'80' - X'82' WAIT AND RETRY
2 *
2 IDJGRFSP EQU X'00' FUNCTION SUCCESSFULLY PROCESSED
2 IDJGRPER EQU X'01' PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJGRFNS EQU X'01' CALLED FUNCTION NOT SUPPORTED
2 IDJGRFNA EQU X'02' CALLED FUNCTION NOT AVAILABLE
2 IDJGRVNA EQU X'03' INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJGRAER EQU X'04' ALIGNMENT ERROR
2 IDJGRIER EQU X'20' INTERNAL ERROR
2 IDJGRCAR EQU X'40' CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJGRECR EQU X'41' SUBSYSTEM (SS) MUST BE CREATED
2 * EXPLICITELY BY CREATE-SS
2 IDJGRECN EQU X'42' SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJGRWAR EQU X'80' WAIT FOR A SHORT TIME AND RETRY
2 IDJGRWLR EQU X'81' " LONG "
2 IDJGRWUR EQU X'82' WAIT TIME IS UNCALCULABLY LONG
2 * BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'

```

```

2 IDJGRTNA EQU X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJGRDH EQU X'82'          SS IN DELETE / HOLD
2 *
2 IDJGMRET DS OAL2           6 MAIN RETURN CODE
2 IDJGMR2 DS AL1             6 MAIN RETURN CODE 2
2 IDJGMR1 DS AL1             7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 IDJGRLNK EQU X'FFFF'       LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJGFHL EQU 8               8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL GETJV PARAMETER LIST *
1 *****
1 IDJGHDR1 EQU X'00290002',4
1 IDJGAR31 DS A               AREA ADDRESS(31 BIT FORMAT)
1 IDJGSIZE DS H               AREA SIZE
1 IDJGJV DS CL54              JVNAME
1 IDJGPOS DS H                SUBSTRING START POSITION
1 IDJGLEN DS H                SUBSTRING LENGTH
1 IDJGPASS DS CL4             PASSWORD
1 DS XL12
1 IDJGFLAG DS X               FLAGS
1 IDJGENCR EQU X'80'          7-7 0=YES, 1=NO
1 * (ENCRYPTION)
1 IDJGFGNV EQU X'40'          6-6 0=NO, 1=YES
1 * (NUMERIC-VALUE)
1 IDJGJVIX EQU X'20'          5-5 0=NO, 1=YES
1 * (JVID INDEXED (SUBSTRING) )
1 IDJGP2 EQU X'10'            4-4 0=P1 CALLER,1=P2 CALLER
1 IDJGECT EQU X'08'            3-3 1=SET BY CMD PROCESSING
1 IDJGFGVB EQU X'04'            2-2 0=NO, 1=YES
1 * (BOOLEAN-VALUE)
1 IDJGNSTR EQU X'02'            1-1 0=NO, 1=YES
1 * (NULLSTRING DEFINED)
1 DS XL3
1 IDJGJVS DS A                RESERVED
1 IDJGPLLN EQU *-GETJV        LENGTH OF DSECT
1 *****
1 SPACE

```

JVSEL

JV-Auswahl des STAJV-Makros auf bestimmte Merkmale beschränken

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (C-/D-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = JSE

Makrobeschreibung

Der Makro JVSEL beschränkt die Jobvariablenmenge, die bei einem STAJV-Aufruf in den Anwenderbereich übertragen werden, auf Jobvariablen mit bestimmten Merkmalen.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
JVSEL	$,ACCESS=\left\{ \begin{array}{l} \underline{*ANY} \\ *READ \\ *WRITE \end{array} \right\}$ $,SHARE=\left\{ \begin{array}{l} \underline{*ANY} \\ *YES \\ *NO \end{array} \right\}$ $,PASS=\left\{ \begin{array}{l} \underline{*ANY} \\ *NONE \\ *RDPASS \\ *WRPASS \\ (list-of-pass) \end{array} \right\}$

Operation	Operanden
	<p data-bbox="400 243 934 680"> ,CRDATE= { *ANY *NONE datum datum(zeit[,]) datum(zeit1,zeit2) (datum[,]) (datum(zeit)[,]) (,datum) (,datum(zeit)) (datum1,datum2) (datum1(zeit),datum2) (datum1,(zeit),datum2(zeit)) } </p> <p data-bbox="400 747 934 1184"> ,EXDATE= { *ANY *NONE datum datum(zeit[,]) datum(zeit1,zeit2) (datum[,]) (datum(zeit)[,]) (,datum) (,datum(zeit)) (datum1,datum2) (datum1(zeit),datum2) (datum1,(zeit),datum2(zeit)) } </p> <p data-bbox="400 1234 624 1344"> ,BASACL= { *ANY *NONE *YES } </p> <p data-bbox="400 1386 745 1486"> ,OWNERAR= { *ANY *NO-ACCESS zugriffsliste } </p>

Operation	Operanden
	,GROUPAR= { <u>*ANY</u> *NO-ACCESS zugriffsliste }
	,OTHERAR= { <u>*ANY</u> *NO-ACCESS zugriffsliste }
	,GUARDS= { <u>*ANY</u> *NONE *YES (READ= { <u>*ANY</u> *NONE fname }, WRITE= { <u>*ANY</u> *NONE fname }) }
	,MANCLAS= { <u>*ANY</u> *NONE <c-string 1..8> }
	,MONJV= { <u>*ANY</u> *YES *NO }
	,CJC= { <u>*ANY</u> *YES *NO }
	,PROTACT= { <u>*ANY</u> *LEVEL-0 *LEVEL-1 *LEVEL-2 (list-of-protact) }

Operation	Operanden
	$,SIZE=\left\{ \begin{array}{l} *ANY \\ zahl \\ (zahl[,]) \\ (,zahl) \\ (zahl1,zahl2) \end{array} \right\}$ $,TIMBASE=\left\{ \begin{array}{l} *LTI \\ *UTC \end{array} \right\}$ $,VERSION=1, MF=\left\{ \begin{array}{l} D \\ C \\ L \end{array} \right\}, PREFIX=\left\{ \begin{array}{l} JSE \\ pre \end{array} \right\}$

ACCESS	Selektiert Jobvariablen abhängig von der erlaubten Zugriffsart.
= <u>*ANY</u>	ist Voreinstellung: Die erlaubte Zugriffsart ist kein Auswahlkriterium.
= *READ	Selektiert Jobvariablen, für die Schreibzugriff mit ACCESS=READ unterbunden ist, d.h. für die nur Lesezugriff zulässig ist.
= *WRITE	Selektiert Jobvariablen, für die Schreibzugriff erlaubt ist.
SHARE	Selektiert Jobvariablen in Abhängigkeit davon, ob sie mehrbenutzbar sind.
= <u>*ANY</u>	ist Voreinstellung: Die Mehrbenutzbarkeit ist kein Auswahlkriterium.
= *YES	Selektiert Jobvariablen, auf die bei aktivierter Standard-Zugriffskontrolle auch andere Benutzerkennungen Zugriff haben.
= *NO	Selektiert Jobvariablen, auf die bei aktivierter Standard-Zugriffskontrolle nur der Eigentümer bzw. Miteigentümer zugreifen darf.
PASS	Selektiert Jobvariablen in Abhängigkeit von dem vereinbarten Kennwortschutz. Werden mehrere Kennwortarten in Listenform angegeben, nimmt das System eine logische Oder-Verknüpfung vor und selektiert alle Jobvariablen, die einer der genannten Bedingungen genügen.
= <u>*ANY</u>	ist Voreinstellung: Der Kennwortschutz ist kein Auswahlkriterium.
= *NONE	Selektiert Jobvariablen, für die kein Kennwortschutz besteht.

- = *RDPASS Selektiert Jobvariablen, die durch ein Lesekennwort geschützt sind.
- = *WRPASS Selektiert Jobvariablen, die durch ein Schreibkennwort geschützt sind.
- = (list-of-pass) In einer Liste kann der Anwender mehrere Arten des Kennwortschutzes angeben. Selektiert alle Jobvariablen, die mit einem der angegebenen Kennworttypen geschützt sind.

CRDATE

Informiert über Jobvariablen in Abhängigkeit vom Erstellungsdatum; Bereichsangaben gelten jeweils inklusive der angegebenen Grenzen. In die Zukunft weisende Angaben sind für CRDATE nicht sinnvoll. Datumsangaben sind in folgender Form möglich:

1. als *absolute Datumsangabe*
ein konkretes Datum in der Form yymmdd bzw. [yy]yy-mm-dd
(yy = Jahr, mm = Monat, dd = Tag)
2. als *relative Datumsangabe*
(6-stellig einschl. Vorzeichen) Distanz zum aktuellen Tagesdatum in der Form -n für die Vergangenheit und [+n] für die Zukunft;
(gestern $\hat{=}$ -1 oder heute $\hat{=}$ ± 0)

- = *ANY ist Voreinstellung: Das Erstellungsdatum ist kein Auswahlkriterium.
- = datum Informiert über Jobvariablen, die zu dem angegebenen Datum erstellt wurden.
- = (datum[,]) Informiert über Jobvariablen, die nach dem angegebenen Datum erstellt wurden (Erstellungsdatum \geq datum).
- = (,datum) Informiert über Jobvariablen, die vor dem angegebenen Datum erstellt wurden (Erstellungsdatum \leq datum).
- = (datum1,datum2) Informiert über Jobvariablen, die innerhalb des angegebenen Zeitraums erstellt wurden (datum1 \leq Erstellungsdatum \leq datum2).
- = datum(zeit[,]) Informiert über alle Jobvariablen, die an dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.
- = datum(zeit1,zeit2) Informiert über alle Jobvariablen, die an dem angegebenen Tag und innerhalb des angegebenen Zeitraums erstellt wurden.
- = (datum(zeit)[,]) Informiert über alle Jobvariablen, die ab dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.

- = (,datum(zeit)) Informiert über alle Jobvariablen, die vor dem angegebenen Tag und ab der angegebenen Uhrzeit erstellt wurden.
- = (datum1(zeit),datum2(zeit)) Informiert über alle Jobvariablen, die in dem angegebenen Zeitraum erstellt wurden. Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.
- EXDATE Selektiert Jobvariablen in Abhängigkeit vom „Freigabedatum“. Das ist das Datum, ab dem für die Jobvariable Schreibzugriffe erlaubt sind. Bereichsangaben gelten jeweils inklusive der angegebenen Grenzen. In die Zukunft weisende Datumsangaben sind sinnvoll, wenn Schutzfristen „abgefragt“ werden. Datumsangaben sind wie sowohl *absolut* als auch *relativ* möglich (siehe „CRDATE“ auf [Seite 140](#)).
- = *ANY ist Voreinstellung: Das Freigabedatum ist kein Auswahlkriterium.
- = datum Informiert über Jobvariablen, in deren Katalogeintrag das angegebene Freigabedatum vereinbart ist.
- = (datum[,]) Informiert über Jobvariablen, deren Freigabedatum \geq datum ist.
- = (,datum) Informiert über Jobvariablen, deren Freigabedatum \leq datum ist.
- = (datum1,datum2) Informiert nur über Jobvariablen, deren Freigabedatum in den angegebenen Zeitraum fällt (datum1 \leq Freigabedatum \leq datum2).
- = datum(zeit[,]) Informiert über alle Jobvariablen, für die das angegebene Freigabedatum vereinbart ist und die Uhrzeit der Freigabe größer oder gleich der angegebenen Zeit ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!
- = datum(zeit1,zeit2) Informiert über alle Jobvariablen, für die das angegebene Freigabedatum vereinbart ist und die Uhrzeit der Freigabe innerhalb des angegebenen Zeitintervall liegt. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!
- = (datum(zeit)[,]) Informiert über alle Jobvariablen, deren Freigabedatum und Freigabezeit größer oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!

= (,datum(zeit))	Informiert über alle Jobvariablen, deren Freigabedatum und Freigabezeit kleiner oder gleich dem angegebenen Zeitpunkt ist. Die Freigabezeit (Uhrzeit bezogen auf das Freigabedatum) wird derzeit immer mit 00:00:00 Uhr im Katalog eingetragen!
= (datum1(zeit),datum2(zeit))	Informiert über alle Jobvariablen, deren Freigabedatum innerhalb des angegebenen Zeitraums liegt ($\text{datum1} \leq \text{Freigabedatum} \leq \text{datum2}$). Die Ober- und Untergrenze des angegebenen Zeitraums werden jeweils durch Angabe einer Uhrzeit genauer bestimmt.
BASACL	Selektiert Jobvariablen in Abhängigkeit davon, ob ein BASIC-ACL-Eintrag existiert.
= <u>*ANY</u>	ist Voreinstellung: Ein BASIC-ACL-Eintrag ist kein Auswahlkriterium.
= *NONE	Selektiert Jobvariablen, für die kein BASIC-ACL-Eintrag definiert ist.
= *YES	Selektiert Jobvariablen, für die ein BASIC-ACL-Eintrag definiert ist.
OWNERAR	Selektiert Jobvariablen in Abhängigkeit von den Zugriffsrechten, die in ihren BASIC-ACL-Einträgen für die Eigentümer festgelegt sind.
= <u>*ANY</u>	ist Voreinstellung: Die BASIC-ACL-Einträge für die Eigentümer sind kein Auswahlkriterium.
= *NO-ACCESS	Selektiert Jobvariablen, auf die der Eigentümer nicht zugreifen darf.
= zugriffsliste	Selektiert Jobvariablen, in deren BASIC-ACL-Eintrag für den Eigentümer mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

zugriffsliste hat folgendes Format:

$$\left[\left\{ \begin{array}{l} \text{READ} = *YES \\ \text{R} = *Y \\ \text{READ} = *NO \\ \text{R} = *N \end{array} \right\} \right] \left[, \left\{ \begin{array}{l} \text{WRITE} = *YES \\ \text{W} = *Y \\ \text{WRITE} = *NO \\ \text{W} = *N \end{array} \right\} \right])$$

Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=*YES bzw. R=*Y	Selektiert Jobvariablen, auf die der Eigentümer lesend zugreifen darf.
READ=*NO bzw. R=*N	Selektiert Jobvariablen, auf die der Eigentümer nicht lesend zugreifen darf.
WRITE=*YES bzw. W=*Y	Selektiert Jobvariablen, auf die der Eigentümer schreibend zugreifen darf.
WRITE=*NO bzw. W=*N	Selektiert Jobvariablen, auf die der Eigentümer nicht schreibend zugreifen darf.

GROUPAR	Selektiert Jobvariablen in Abhängigkeit von den Zugriffsrechten, die in ihren BASIC-ACL-Einträgen für die Mitglieder der Benutzergruppe des Eigentümers festgelegt sind.
= <u>*ANY</u>	Die BASIC-ACL-Einträge für die Mitglieder der Benutzergruppe des Eigentümers sind kein Auswahlkriterium.
= *NO-ACCESS	Selektiert Jobvariablen, auf die die Benutzergruppe des Eigentümers nicht zugreifen darf.
= zugriffsliste	Selektiert Jobvariablen, in deren BASIC-ACL-Eintrag für die Benutzergruppe des Eigentümers mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

zugriffsliste hat folgendes Format:

$$\left[\left\{ \begin{array}{l} \text{READ} = *YES \\ \text{R} = *Y \\ \text{READ} = *NO \\ \text{R} = *N \end{array} \right\} \right] \left[, \left\{ \begin{array}{l} \text{WRITE} = *YES \\ \text{W} = *Y \\ \text{WRITE} = *NO \\ \text{W} = *N \end{array} \right\} \right] \right]$$

Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=*YES bzw. R=*Y	Selektiert Jobvariablen, auf die die Benutzergruppe des Eigentümers zugreifen darf.
READ=*NO bzw. R=*N	Selektiert Jobvariablen, auf die die Benutzergruppe des Eigentümers nicht lesend zugreifen darf.
WRITE=*YES bzw. W=*Y	Selektiert Jobvariablen, auf die die Benutzergruppe des Eigentümers schreibend zugreifen darf.
WRITE=*NO bzw. W=*N	Selektiert Jobvariablen, auf die die Benutzergruppe des Eigentümers nicht schreibend zugreifen darf.

OTHERAR	Selektiert Jobvariablen in Abhängigkeit von den Zugriffsrechten, die in ihren BASIC-ACL-Einträgen für alle Anwender außerhalb der Benutzergruppe des Eigentümers festgelegt sind.
= <u>*ANY</u>	ist Voreinstellung: Die BASIC-ACL-Einträge für alle Anwender außerhalb der Benutzergruppe des Eigentümers sind kein Auswahlkriterium.
= *NO-ACCESS	Selektiert Jobvariablen, auf die Anwender außerhalb der Benutzergruppe des Eigentümers nicht zugreifen dürfen.
= zugriffsliste	Selektiert Jobvariablen, in deren BASIC-ACL-Eintrag für Anwender außerhalb der Benutzergruppe des Dateieigentümers mindestens eines der in der Liste angegebenen Zugriffsrechte vereinbart ist.

zugriffsliste hat folgendes Format:

$$\left[\begin{array}{l} \text{READ} = *YES \\ \text{R} = *Y \\ \text{READ} = *NO \\ \text{R} = *N \end{array} \right] \left[, \begin{array}{l} \text{WRITE} = *YES \\ \text{W} = *Y \\ \text{WRITE} = *NO \\ \text{W} = *N \end{array} \right] \right)$$

Die runden Klammern sind Bestandteil des Operandenwertes und müssen mit angegeben werden.

Die einzelnen Elemente der Zugriffsliste haben folgende Bedeutung:

READ=*YES bzw. R=*Y	Selektiert Jobvariablen, auf die Anwender außerhalb der Benutzergruppe des Dateieigentümers zugreifen darf.
READ=*NO bzw. R=*N	Selektiert Jobvariablen, auf die Anwender außerhalb der Benutzergruppe des Dateieigentümers nicht lesend zugreifen darf.
WRITE=*YES bzw. W=*Y	Selektiert Jobvariablen, auf die Anwender außerhalb der Benutzergruppe des Dateieigentümers schreibend zugreifen darf.
WRITE=*NO bzw. W=*N	Selektiert Jobvariablen, auf die Anwender außerhalb der Benutzergruppe des Dateieigentümers nicht schreibend zugreifen darf.

GUARDS	Selektiert Jobvariablen abhängig von einem vereinbarten Zugriffsschutz mit GUARDS (siehe Handbuch „SECOS“ [9]).
= <u>*ANY</u>	ist Voreinstellung: Der vereinbarte Zugriffsschutz mit GUARDS ist kein Auswahlkriterium.
= *NONE	Selektiert Jobvariablen, die keinen Zugriffsschutz über GUARDS definiert haben.
= *YES	Selektiert Jobvariablen, die einen Zugriffsschutz über GUARDS definiert haben.
= (READ=...,WRITE=...)	Innerhalb einer Liste kann der Anwender angeben, wie der Zugriffsschutz mit GUARDS für die auszuwählenden Jobvariablen vereinbart sein soll. Für jede Zugriffsart (Lesen und Schreiben) kann der vereinbarte Schutz genau angegeben werden. Wird für eine Zugriffsart keine Angabe gemacht, so erfolgt die Auswahl unabhängig von dem dafür vereinbarten Schutz.
Je Zugriffsart kann einer der folgenden Werte angegeben werden:	
*ANY	Der vereinbarte GUARDS-Schutz ist kein Auswahlkriterium.
*NONE	Für die angegebene Zugriffsart ist kein Guard vereinbart, d.h. der entsprechende Zugriff wird untersagt.
fname	Für die angegebene Zugriffsart sind im Guard fname alle Bedingungen für die Zugriffserlaubnis enthalten.
MANCLAS	Selektiert Jobvariablen entsprechend der HSMS-Management-Klasse zur Dateisicherung auf SM-Pubsets.
= <u>*ANY</u>	ist Voreinstellung: Die HSMS-Management-Klasse ist kein Auswahlkriterium.
= *NONE	Selektiert Jobvariablen, für die keine HSMS-Management-Klasse definiert ist.
= <c-string 1..8>	Selektiert Jobvariablen mit der angegebenen HSMS-Management-Klasse.
MONJV	Selektiert Jobvariablen in Abhängigkeit ihrer Verwendung als auftragsüberwachende Jobvariable.
= <u>*ANY</u>	ist Voreinstellung: Die Verwendung als auftragsüberwachende Jobvariable ist kein Auswahlkriterium.
= *NO	Selektiert Jobvariablen, die keinen Auftrag überwachen.

= *YES	Selektiert Jobvariablen, die einen Auftrag überwachen (siehe auch Kommando SHOW-JV-ATTRIBUTES, Ausgabefeld JV-TYPE IS MONJV im Handbuch „Kommandos“ [1]).
CJC	Selektiert Jobvariablen in Abhängigkeit ihrer Verwendung in CJC-Funktionen.
= *ANY	ist Voreinstellung: Die Verwendung als auftragsüberwachende Jobvariable ist kein Auswahlkriterium.
= *NO	Selektiert Jobvariablen, die nicht in CJC-Funktionen verwendet werden.
= *YES	Selektiert Jobvariablen, die in CJC-Funktionen verwendet werden.
PROTACT	Selektiert Jobvariablen abhängig von der Schutzstufe der höchsten aktivierten Zugriffskontrolle.

Für Zugriffe auf die Jobvariable gilt der höchste aktivierte Zugriffsschutz. Die nachfolgende Tabelle zeigt Art der Zugriffskontrolle, Schutzmerkmal, das im CATJV-Makroaufruf anzugeben ist und die Rangfolge (Schutzstufe):

Zugriffsschutz	Schutzmerkmal (Operand im CATJV)	Schutzstufe
Standard-Zugriffskontrolle	ACCESS u. SHARE	0
Einfache Zugriffskontroll-Liste	BASACL, OWNERAR, GROUPAR, OTHERAR	1
Zugriffskontrolle über GUARDS	GUARDS	2

Alle weiteren Schutzmerkmale der Jobvariablen (z.B. Kennwörter) werden unabhängig von der realisierten Schutzstufe ausgewertet.

= *ANY	ist Voreinstellung: Selektiert Jobvariablen, unabhängig von der Schutzstufe der höchsten aktivierten Zugriffskontrolle.
= *LEVEL-0	Selektiert Jobvariablen, bei denen die Zugriffe über die Standard-Zugriffskontrolle erfolgen.
= *LEVEL-1	Selektiert Jobvariablen, bei denen die Zugriffe über eine einfache Zugriffskontroll-Liste (BASIC-ACL-Schutz) erfolgen.
= *LEVEL-2	Selektiert Jobvariablen, bei denen die Zugriffe über GUARDS erfolgen.
= (list-of-protact)	Der Anwender kann in einer Liste maximal 3 Schutzstufen angeben. Selektiert Jobvariablen, bei denen die Zugriffe über eine Zugriffskontrolle erfolgen, die einer der angegebenen Schutzstufen entspricht.

SIZE	Selektiert Jobvariablen in Abhängigkeit von der Länge des Jobvariablenwertes. Bereichsangaben gelten jeweils einschließlich der Bereichsgrenzen.
= <u>*ANY</u>	ist Voreinstellung: Die Länge des Jobvariablenwertes ist kein Auswahlkriterium.
= zahl	Selektiert Jobvariablen, deren Wert die angegebene Anzahl von Bytes lang ist. Mögliche Werte: $0 \leq \text{zahl} \leq 256$
= (zahl[,])	Informiert nur über Jobvariablen, deren Wert mindestens angegebene Anzahl von Bytes lang ist ($\text{SIZE} \geq$ angegebener Wert).
= (,zahl)	Informiert nur über Jobvariablen, deren Wert höchstens angegebene Anzahl von Bytes lang ist ($\text{SIZE} \leq$ angegebener Wert).
= (zahl1,zahl2)	Informiert über Jobvariablen, bei denen die Länge des Jobvariablenwertes im angegebenen Bereich liegt ($\text{zahl1} \leq \text{Länge} \leq \text{zahl2}$, wobei $\text{zahl1} < \text{zahl2}$ sein muss).
TIMBASE	Bestimmt die Zeitbasis, die bei Angaben zu Erstellungs- bzw. Freigabedatum verwendet wird.
= <u>*LTI</u>	ist Voreinstellung: Datums- bzw. Zeitangaben werden als Angaben in lokaler Zeit interpretiert.
= *UTC	Datums- bzw. Zeitangaben werden als Angaben in UTC-Zeit (Weltzeit) interpretiert.
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.
VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
= <u>1</u>	ist Voreinstellung: Die Makroauflösung entspricht dem Stand JV V14.0.

Rückinformation und Fehleranzeigen

Die derzeit einzige Möglichkeit, eine JVSEL-Parameterliste zu verwenden, ist die Verwendung als SELADDR in einem STAJV. Dieser liefert einen Returncode gemäß [Seite 229](#) und speziell beim STAJV-Returncode 0491 (Fehler beim Selektieren) hinterlegt er zusätzlich im Feld JSERC der JVSEL-Parameterliste die Angabe, welches Selektionskriterium fehlerhaft ist.

DSECT

```

JVSEL    JVSEL MF=D
1  JVSEL    MFCHK MF=D,SUPPORT=(D,C),PREFIX=J,MACID=SE,DMACID=JSE
2  JVSEL    DSECT ,
2          *,##### PREFIX=J, MACID=JSE #####
1  *
1  *
1          DS      OA
1  JSETEXT  DC      C'SELECT'
1  JSEVERS  DC      X'01'
1  JSERC    DC      X'FF'
1  JSERCOK  EQU     X'00'          NO ERROR
1  JSERCCRD EQU     X'01'          INVALID CRDATE
1  JSERCEXD EQU     X'02'          INVALID EXDATE
1  JSERCCRT EQU     X'03'          INVALID CRTIME
1  JSERCEXT EQU     X'04'          INVALID EXTIME
1  JSERCUND EQU     X'FF'          UNDEFINED
1  *
1  JSESELIO DC      B'00000000'    SELECT INDO
1  JSESICRD EQU     X'80'          CREATION-DATE
1  JSESIEXD EQU     X'40'          EXPIRATION-DATE
1  JSESIIPWP EQU    X'20'          PASSWORD-PROTECTED
1  JSESIIPRO EQU    X'10'          PROTECTION-LEVEL
1  JSESIACC EQU     X'08'          ACCESS
1  JSESISHR EQU     X'04'          SHARE
1  JSESIBAC EQU     X'02'          BASIC-ACL
1  JSESIGUA EQU     X'01'          GUARDS
1  *
1  JSESELII DC      B'00000000'    SELECT IND1
1  JSEISISIZ EQU    X'80'          SIZE
1  JSESIMON EQU     X'40'          MONJV
1  JSESIMCL EQU     X'20'          MANAGEMENT-CLASS
1  JSESIJCJ EQU     X'10'          CJC
1  JSESI5UU EQU     X'0F'          -- UNUSED, MUST BE 0 --
1  *
1  JSEUNUS1 DC      X'0000'
1  *
1  JSEBACL  DC      B'00000000'    BASIC-ACL

```

```

1 JSEBACLY EQU X'80' YES
1 JSEBACLN EQU X'40' NONE
1 JSEBACLU EQU X'2F' -- UNUSED, MUST BE 0 --
1 *
1 JSEBOW DC B'00000000' BASIC-ACL-OWNER
1 JSEBOWRS EQU X'80' READ-RIGHT-SPECIFIED
1 JSEBOWWS EQU X'40' WRITE-RIGHT-SPECIFIED
1 JSEBOWUN EQU X'20' -- UNUSED, MUST BE 0 --
1 JSEBOWRY EQU X'10' READ = YES
1 JSEBOWWY EQU X'08' WRITE = YES
1 JSEBOWNU EQU X'04' -- UNUSED, MUST BE 0 --
1 JSEBOWNO EQU X'02' NO-ACCESS
1 JSEBOWUU EQU X'01' -- UNUSED, MUST BE 0 --
1 *
1 JSEBGR DC B'00000000' BASIC-ACL-GROUP
1 JSEBGRRS EQU X'80' READ-RIGHT-SPECIFIED
1 JSEBGRWS EQU X'40' WRITE-RIGHT-SPECIFIED
1 JSEBGRUN EQU X'20' -- UNUSED, MUST BE 0 --
1 JSEBGRRY EQU X'10' READ = YES
1 JSEBGRWY EQU X'08' WRITE = YES
1 JSEBGRNU EQU X'04' -- UNUSED, MUST BE 0 --
1 JSEBGRNO EQU X'02' NO-ACCESS
1 JSEBGRUU EQU X'01' -- UNUSED, MUST BE 0 --
1 *
1 JSEBOT DC B'00000000' BASIC-ACL-OTHERS
1 JSEBOTRS EQU X'80' READ-RIGHT-SPECIFIED
1 JSEBOTWS EQU X'40' WRITE-RIGHT-SPECIFIED
1 JSEBOTUN EQU X'20' -- UNUSED, MUST BE 0 --
1 JSEBOTRY EQU X'10' READ = YES
1 JSEBOTWY EQU X'08' WRITE = YES
1 JSEBOTNU EQU X'04' -- UNUSED, MUST BE 0 --
1 JSEBOTNO EQU X'02' NO-ACCESS
1 JSEBOTUU EQU X'01' -- UNUSED, MUST BE 0 --
1 *
1 JSECRD DC CL10' ' CREATION DATE - FROM DATE
1 JSECRT DC CL8' ' TIME
1 JSECRD2 DC CL10' ' - TO DATE
1 JSECRT2 DC CL8' ' TIME
1 *
1 JSEEXD DC CL10' ' EXPIRATION DATE - FROM DATE
1 JSEEXT DC CL8' ' TIME
1 JSEEXD2 DC CL10' ' - TO DATE
1 JSEEXT2 DC CL8' ' TIME
1 *
1 JSEGUA DC B'00000000' GUARDS SPECIFIED FLAG
1 JSEGUARS EQU X'80' 7-7 S READ-SPECIFIED
1 JSEGUAWS EQU X'40' 6-6 S WRITE-SPECIFIED
1 JSEGUAUN EQU X'20' 5-5 R -- UNUSED, MUST BE 0 --

```

```

1 JSEGUUNU EQU X'10' 4-4 R -- UNUSED, MUST BE 0 --
1 JSEGUANS EQU X'08' 3-3 S GUARDS=NONE SPECIFIED
1 JSEGUAYS EQU X'04' 2-2 S GUARDS=YES SPECIFIED
1 JSEGUAAU EQU X'03' 1-0 R -- UNUSED, MUST BE 0 --
1 JSEGUAR DC CL18' ' GUARDS-READ
1 JSEGUAW DC CL18' ' GUARDS-WRITE
1 *
1 JSEPPW DC B'00000000' PW PROTECT FLAG -
1 JSEPPWRD EQU X'80' 7-7 S READ
1 JSEPPWR EQU X'40' 6-6 S WRITE
1 JSEPPUN EQU X'20' 5-5 R -- UNUSED, MUST BE 0 --
1 JSEPPNO EQU X'10' 4-4 S NONE
1 JSEPPNU EQU X'0F' 3-0 R -- UNUSED, MUST BE 0 --
1 *
1 JSEPROL DC B'00000000' PROTECTION-LEVEL-FLAG
1 JSEPROL0 EQU X'80' 7-7 S LEVEL 0
1 JSEPROL1 EQU X'40' 6-6 S LEVEL 1
1 JSEPROL2 EQU X'20' 5-5 S LEVEL 2
1 JSEPROU EQU X'1F' 4-0 R -- UNUSED, MUST BE 0 --
1 *
1 JSESELF1 DC B'00000000' SELECT FLAG 1 -
1 JSEACCRD EQU X'80' 7-7 S ACCESS = READ
1 JSEACCR EQU X'40' 6-6 S ACCESS = WRITE
1 JSESHARY EQU X'20' 5-5 S SHARE = YES
1 JSESHARN EQU X'10' 4-4 S SHARE = NO
1 JSEMONY EQU X'08' 3-3 S MONJV = YES
1 JSEMONN EQU X'04' 2-2 S MONJV = NO
1 JSECJCY EQU X'02' 1-1 S CJC = YES
1 JSECJCN EQU X'01' 0-0 S CJC = NO
1 *
1 JSESELF2 DC B'00000000' SELECT FLAG 2 -
1 JSETLTI EQU X'80' 7-7 S TIME-BASE = LOCAL
1 JSESF2UU EQU X'7F' 6-0 R -- UNUSED, MUST BE 0 --
1 *
1 JSEUNUS2 DC X'000000'
1 *
1 JSESIZE DC A(0) SIZE - FROM
1 JSESIZE2 DC A(0) - TO
1 *
1 JSEMGML DC CL8' ' MANAGEMENT-CLASS
1 JSEFUT4 DC 60X'00' SPACE FOR FUTURE USE, MUST BE 0
1 JSESPLLN EQU *-JVSEL

```

LNKJV

Jobvariablen JV-LINK-Einträge ausgeben

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **LNKJV** überträgt Informationen über die Verknüpfung von Jobvariablen mit Jobvariablen-Kettungsnamen (JV-LINK) aus der JV-LINK-Tabelle in einen Benutzerbereich.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
LNKJV	bereich [,länge] $\left[, \left\{ \begin{array}{l} \text{JVNAME=jvname} \\ \text{LINK=*jvlink} \end{array} \right\} \right]$ $, \text{ODSECT} = \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$ $, \text{VERSION} = \left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\} , \text{MF} = \left\{ \begin{array}{l} \text{S} \\ \text{C} \\ \text{(E,...)} \\ \text{D} \\ \text{L} \end{array} \right\} , \text{PREFIX} = \left\{ \begin{array}{l} \text{IDJ} \\ \text{pre} \end{array} \right\}$

bereich symbolische Adresse eines Bereichs im Programm, in den die LNKJV-Information übertragen werden soll.

Versorgung des Bereichs durch LNKJV:

l1	javlink ₁	jvname ₁	l2	javlink ₂ ...
1Byte	8Byte	max.54Byte		

<----->

l1=Gesamtlänge des Eintrags, max. 63 Bytes

Das Ende der aufgelisteten LNKJV-Informationen wird mit einem Byte gekennzeichnet. Dieses letzte Byte besitzt den Wert X'00' wenn die LNKJV-Informationen vollständig in den Benutzerbereich übertragen wurden. Konnte der Benutzerbereich nicht alle Einträge aufnehmen, besitzt das letzte Byte den Wert X'01'.

länge explizite Länge des Benutzerbereichs.
Fehlt die Angabe, wird die implizite Länge von bereich verwendet.
Die Länge des Benutzerbereichs sollte mindestens 63 Byte groß sein.

JVNAME=jvname voll- oder teilqualifizierter Pfadname einer permanenten oder temporären Jobvariablen, deren Kettungsname zusätzlich im definierten Bereich hinterlegt werden soll. Die Verwendung von Musterteichen ist zulässig.

LINK=*javlink gültiger Kettungsname einer Jobvariablen (wie in DCLJV definiert), deren vollständiger Pfadname zusätzlich im definierten Bereich hinterlegt werden soll.

ODSECT Gibt an, ob zusätzlich eine DSECT für die Ausgabe einer einzelnen LNKJV-Information zu generieren ist. Die Angabe ist nur mit MF=D sinnvoll.

=NO ist Voreinstellung: Nur eine DSECT für die Operandenliste des LNKJV-Makros wird generiert.

=YES generiert zusätzlich eine DSECT für die Ausgabe einer einzelnen LNKJV-Information (Aufbau siehe „bereich“ bzw. DSECT).

MF Zur Beschreibung der Operanden MF und PREFIX siehe [Seite 88](#).
PREFIX Die gültigen Werte sind zu Beginn der Makrobeschreibung und aus dem Aufrufformat ersichtlich.

VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$.
=1	Die Makroauflösung entspricht $JV \geq V10.0$.

Hinweise zur DSECT

- Der Aufruf des Makros LNKJV mit den Operanden MF=D erzeugt eine DSECT. Bei der Angabe VERSION=1 wird die DSECT mit Standardheader erzeugt.
- Standardmäßig wird eine DSECT für die Operandenliste des LNKJV-Makros erzeugt. Zusätzlich wird bei Angabe von ODSECT=YES eine DSECT für die Ausgabe einer LNKJV-Information erzeugt.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

LNKJV    LNKJV MF=D,VERSION=1,ODSECT=YES
1 *****
1 *      VERSION 320
1 *****
1 *      L N K J V   P A R A M E T E R   L I S T                *
1 *****
1          #INTF REFTYPE=REQUEST,                                C
1          INTNAME=LNKJV,INTCOMP=001
1 LNKJV    DSECT
1 *****
1 *      UNIT=41, FUNCTION=7,  VERSION=<PARAMETER VERSION>      *
1 *****
1          FHDR  MF=(C,IDJL)
2          DS   0A
2 IDJLFHE  DS   0XL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 IDJLIFID DS   0A          0  INTERFACE IDENTIFIER
2 IDJLFCTU DS   AL2        0  FUNCTION UNIT NUMBER
2 *
2 *          BIT 15  HEADER FLAG BIT,
2 *          MUST BE RESET UNTIL FURTHER NOTICE
2 *          BIT 14-12 UNUSED, MUST BE RESET
2 *          BIT 11-0  REAL FUNCTION UNIT NUMBER
2 IDJLFCT  DS   AL1        2  FUNCTION NUMBER
2 IDJLFCTV DS   AL1        3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJLRET  DS   0A          4  GENERAL RETURN CODE

```

```

2 *
2 * GENERAL_RETURN_CODE CLEARED (X'0000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJLSRET DS    OAL2                4  SUB RETURN CODE
2 IDJLSR2  DS    AL1                4  SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJLR2OK EQU   X'00'                All correct, no additional info
2 IDJLR2NA EQU   X'01'                Successful, no action was necessary
2 IDJLR2WA EQU   X'02'                Warning, particular situation
2 IDJLSR1  DS    AL1                5  SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A      X'00'                FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B      X'01' - X'1F'        PARAMETER SYNTAX ERROR
2 * CLASS C      X'20'                INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D      X'40' - X'7F'        NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E      X'80' - X'82'        WAIT AND RETRY
2 *
2 IDJLRFSP EQU   X'00'                FUNCTION SUCCESSFULLY PROCESSED
2 IDJLRPER EQU   X'01'                PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJLRFNS EQU   X'01'                CALLED FUNCTION NOT SUPPORTED
2 IDJLRFNA EQU   X'02'                CALLED FUNCTION NOT AVAILABLE
2 IDJLRVNA EQU   X'03'                INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJLRAER EQU   X'04'                ALIGNMENT ERROR
2 IDJLRIER EQU   X'20'                INTERNAL ERROR
2 IDJLRCAR EQU   X'40'                CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJLRECR EQU   X'41'                SUBSYSTEM (SS) MUST BE CREATED
2 *                                     EXPLICITELY BY CREATE-SS
2 IDJLRECN EQU   X'42'                SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJLRWAR EQU   X'80'                WAIT FOR A SHORT TIME AND RETRY
2 IDJLRWLR EQU   X'81'                "          LONG          "
2 IDJLRWUR EQU   X'82'                WAIT TIME IS UNCALCULABLY LONG
2 *                                     BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJLRTNA EQU   X'81'                SS TEMPORARILY NOT AVAILABLE
2 IDJLRDH  EQU   X'82'                SS IN DELETE / HOLD
2 *
2 IDJLMRET DS    OAL2                6  MAIN RETURN CODE
2 IDJLMR2  DS    AL1                6  MAIN RETURN CODE 2
2 IDJLMR1  DS    AL1                7  MAIN RETURN CODE 1
2 *

```

```

2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XYYYY')
2 *
2 IDJLRLNK EQU   X'FFFF'           LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJLFHL  EQU   8                 8   GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 *           END OF STANDARD HEADER - START SPECIAL LNKJV PARAMETERLIST   *
1 *****
1 IDJLHDRI      EQU   X'00290701',4
1 IDJLLINK      DS    CL8           LINKNAME
1 IDJLNAME      DS    CL80         JOBVARIABLE NAME
1 IDJLUNU2      DS    CL20         UNUSED
1 IDJLADDR      DS    A            AREA ADDRESS
1 IDJLSIZE      DS    H            AREA SIZE
1 IDJLFLAG      DS    X            INDICATOR
1 IDJLNOLI      EQU   X'80'        LINK NOT SPECIFIED
1 IDJLNONA      EQU   X'10'        JVNAME NOT SPECIFIED
1 IDJLECT       EQU   X'08'        SET BY CMD PROCESSING
1 IDJLP2        EQU   X'04'        P2 CALLER
1              DS    CL1           ALIGNMENT
1 IDJLJVS       DS    A            RESERVED
1 IDJLPLLN      EQU   *-LNKJV
1              SPACE
1 *****
1 *           L N K J V   O U T P U T                                     *
1 *****
1 IDJEL        DSECT
1 IDJELLN      DS    C            LAYOUT LENGTH
1 IDJELINK     DS    CL8         LINK NAME
1 IDJELJVN     DS    CL54        JV NAME (FULLY QUALIFIED)
1 IDJELLEN     EQU   *-IDJEL     LENGTH OF MACRO
1 *****
1              SPACE

```

ONEVT

Bedingung für ein Jobvariablen-Ereignis setzen

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ (bei VERSION=0): Standardform/C-/D-/E-/L-Form
 S-Typ (bei VERSION=1): Standardform/C-/D-/E-/L-/M-Form
 siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Makrobeschreibung

Im Makroaufruf **ONEVT** wird eine Bedingung angegeben und eine Ereigniskennung vorgegeben. Während der Programmausführung wird vom System immer dann ein **POSSIG** abgesetzt, wenn das Ereignis „Bedingung erfüllt“ oder „Katalog exportiert“ eintritt. Die Maximalzahl abzusetzender **POSSIG**-Aufrufe kann angegeben werden. Zur Ereignissteuerung siehe Handbuch „Makroaufrufe an den Ablaufteil“ [4].

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
ONEVT	$\left\{ \begin{array}{l} \text{'bed.ausdruck'} \\ \text{adr1} \\ \text{(r1)} \end{array} \right\} , \text{EIID} = \left\{ \begin{array}{l} \text{adr} \\ \text{(r)} \end{array} \right\}$ $\left[, \text{POST} = \left\{ \begin{array}{l} \text{wert} \\ \text{(r)} \end{array} \right\} \right] , \text{COUNT} = \left\{ \begin{array}{l} \frac{1}{\text{zahl}} \\ \text{(r)} \end{array} \right\}$

Operation	Operanden
ONEVT (Forts.)	$\left. \begin{array}{l} \left. \begin{array}{l} \underline{S} \\ (D, pre) \\ D \\ (E, \dots) \\ C \\ (C, pre) \\ L \end{array} \right\} \\ , VERSION = \underline{0}, MF = \end{array} \right\}$ $\left. \begin{array}{l} \left. \begin{array}{l} \underline{S} \\ C \\ D \\ E \\ L \\ M \end{array} \right\} [, PARAM = \left\{ \begin{array}{l} adr \\ (r) \end{array} \right\}], PREFIX = \left\{ \begin{array}{l} J \\ pre \end{array} \right\} \\ , MACID = \left\{ \begin{array}{l} VSC \\ macid \end{array} \right\} \end{array} \right\}$

'bed.ausdruck'

Die Bedingung wird als Direktoperand angegeben und muss in Apostrophe eingeschlossen werden. Wegen der Verwendung des Apostrophs in der Assemblersprache als syntaktisches Zeichen müssen folgende Regeln beachtet werden:

Jeder Apostroph innerhalb von „bed.ausdruck“ muss verdoppelt werden. Die Maximallänge des „bed.ausdruck“ ohne die einschließenden Apostrophe beträgt 127 Bytes.

Sonder-Jobvariablen sind nicht zulässig.

adr1

symbolische Adresse eines Bereichs im Benutzerprogramm, der den „bed.ausdruck“ enthält.

Das Satzformat muss vom Typ „variabel“ sein. Der Bereich beginnt mit einem 4 Byte langen Feld, dessen erste zwei Byte die Länge des „bed.ausdruck“ in Byte + 4 enthalten.

Beispiel

```

SYMADR1      DC   Y(END-SYMADR1)
              DS   CL2
              DC   'bedingter ausdruck'
END          EQU  *
```

- (r1) r1 = Register, das die Adresse des „bed.ausdruck“ enthält, der wie unter adr1 beschrieben aufgebaut sein muss. Der „bed.ausdruck“ darf keine Sonder-Jobvariablen enthalten.
- EIID benennt die Ereigniskurzkenung, die durch den TU-Eventing Makroaufruf **ENAEI** dem Benutzerprogramm zur Verfügung gestellt wird. Der Makro **ENAEI** muss vor dem **ONEVT**-Makro aufgerufen werden.
- =adr symbolische Adresse der Ereigniskurzkenung,
- =(r) r = Register, das die Adresse der Ereigniskurzkenung enthält.
- POST gibt einen 2 Byte langen Wert an, der auf Programmebene zur Identifizierung des **ONEVT** verwendet werden kann. Er wird an das Benutzerprogramm in den zwei am weitesten rechts stehenden Bytes des Post Codes übergeben. Der Post Code ist 4 Byte lang und wird dem Programm bei Ereignissen wie „Bedingung erfüllt“ etc. mitgeteilt.
- =wert 2 Byte lange Assembler-Konstante beliebigen Formats.
- =(r) r = Register, das die Adresse eines 2 Byte langen Feldes enthält, in dem der Benutzer den Wert hinterlegt hat.
- Format des Post Codes:
- | ereignisbezogene Anzeige | Bedingungsergebnis | 'ONEVT' Kennungs-„wert“ |
|--------------------------|--------------------|-------------------------|
| 1 Byte | 1 Byte | 2 Byte |
- Bedeutung der ersten beiden Bytes:
- ereignisbezogene Anzeige:
 - X'14' gibt an, dass das Ereignis durch einen ONEVT-Makro verursacht wurde.
 - Bedingungsergebnis:
 - gibt den 'Grund' für den vom System angegebenen POSSIG an.
 - X'00': Bedingung erfüllt
 - X'08': Katalog exportiert
- Ist der Post-Operand nicht vorhanden, so wird auch kein „wert“ zur ONEVT-Kennung (d.h. also X'00000000') übermittelt.

COUNT	gibt an, wie viel POSSIG -Aufrufe maximal vom System abgesetzt werden dürfen ($1 \leq \text{zahl} \leq 32767$); Standardwert = 1. Vom COUNT-Operanden hängt es ab, wie oft das Programm über ein Ereignis unterrichtet wird.
=zahl	Anzahl der erlaubten POSSIG -Aufrufe für den Fall „Bedingung erfüllt“. Wenn die Anzahl erreicht ist, erlischt die Wirksamkeit des Makroaufrufs ONEVT . Unabhängig von der Angabe im COUNT-Operanden erlischt die Wirksamkeit des ONEVT -Makros unmittelbar nach einem POSSIG-Aufruf, der durch das Ereignis „Katalog exportiert“ verursacht wurde oder durch einen DONEVT-Makroaufruf.
=(r)	r = Register, das die Adresse eines Halbwortes enthält, in dem der Wert des COUNT-Operanden abgespeichert ist.
MF PREFIX MACID PARAM	Zur Beschreibung der Operanden MF, PREFIX, MACID und PARAM siehe Seite 88 . Die gültigen Werte und Voreinstellungen von MF für diesen Makro sind zu Beginn der Makrobeschreibung, von PREFIX und MACID bei der Beschreibung des Operanden VERSION angegeben.
VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Bei der C- und D-Form des Makroaufrufs kann ein Präfix (pre=1..3 Buchstaben) angegeben werden. Voreinstellung: pre=ONE
=1	Die Makroauflösung entspricht dem Stand $JV \geq V10.0$. Bei der C-, D- oder M-Form des Makroaufrufs kann ein Präfix (p=1 Buchstabe) und bei der C- oder M-Form zusätzlich eine Macid MACID angegeben werden.

Funktionsweise

Der Benutzer kann das Ergebnis „Bedingung erfüllt“ - über das er vom POSSIG unterrichtet wird - nach seinen Vorstellungen der BS2000-TU-Ereignissteuerung unterziehen, z.B. abwarten, bis „Bedingung erfüllt“ eintritt (SOLSIG COND=) oder dass asynchron ein Contingency-Prozess ausgeführt wird, wenn „Bedingung erfüllt“ gilt (SOLSIG COID=....).

Rückinformation und Fehleranzeigen

bei VERSION=0:

R15

b	b							a	a

 Über die Ausführung des Makros ONEVT (VERSION=0) wird im Register R15 ein Returncode übergeben:
(aa = Rücksprungschalter, bb = Sekundärindikator)

X'bb'	X'aa'	Erläuterung
X'00'	X'00'	Ausführung normal beendet.
X'00'	X'04'	Funktion nicht ausgeführt: ungültige Adresse für die Ereigniskennung bzw. die Bedingung oder ungültige COUNT-Angabe.
X'04'	X'04'	Funktion nicht ausgeführt: Ereigniskennung nicht gefunden.
X'08'	X'04'	Funktion nicht ausgeführt: ungültiger bedingter Ausdruck.
X'10'	X'04'	Funktion nicht ausgeführt: auf eine angegebene Jobvariable kann nicht zugegriffen werden.
	X'FF'	Funktion nicht ausgeführt: CJC im System nicht verfügbar.

bei VERSION=1:

Nach Initialisierung des Standardheaders (bei Aufruf mit MF=S/L) wird die Rückkehrinformation an der symbolischen Adresse <PREFIX><MACID>RET zur Verfügung gestellt (4 Byte).

Standardheader

c	c							a	a

 Über die Ausführung des Makros ONEVT (VERSION=1) wird im Standardheader ein Returncode übergeben:
(aaaa = MAINCODE, bb = SUBCODE1, cc = SUBCODE2)

X'cc'	X'aaaa'	Erläuterung
X'00'	X'0000'	Ausführung normal beendet.
X'00'	X'0004'	Funktion nicht ausgeführt: ungültige Adresse für die Bedingung bzw. die Ereigniskennung oder unzulässiger Wert für COUNT.
X'04'	X'0004'	Funktion nicht ausgeführt: Ereigniskennung nicht gefunden.
X'08'	X'0004'	Funktion nicht ausgeführt: Bedingung fehlerhaft.
X'10'	X'0004'	Funktion nicht ausgeführt: auf eine genannte Jobvariable kann nicht zugegriffen werden.
X'14'	X'0004'	Funktion nicht ausgeführt: es ist nicht genügend Speicherplatz vorhanden.
X'18'	X'0004'	Funktion nicht ausgeführt: die Ereignissteuerung ist nicht verfügbar.
	X'FFFF'	Funktion nicht ausgeführt: Fehler bei der Initialisierung des Standardheaders, siehe Seite 234 .

Zusätzlich zu den bei VERSION=0 bzw. 1 genannten Returncodes können bei grundlegenden Fehlern (z.B. ungültige Adresse der Operandenliste) auch allgemeine JV-Returncodes auftreten. Deren Bedeutung kann dem Kommentar zum entsprechenden Returncode im Makro IDEJVS ([Seite 229](#)) entnommen werden.

DSECT

```

ONEVT    ONEVT MF=D,VERSION=1
1 ONEVT  MFCHK MF=D,                                     C
1        SUPPORT=(C,D,E,L,M,S),                         C
1        PREFIX=J,                                       C
1        MACID=VSC,                                       C
1        DMACID=VSC,                                     C
1        DNAME=ONEVTPL,                                   C
1        PARAM=,                                         C
1        SVC=190
2 ONEVT  DSECT ,
2        *,##### PREFIX=J, MACID=VSC #####
1 JVSCFHDR  FHDR MF=(C,JVSC),EQUATES=NO
2 JVSCFHDR DS    0A
2 JVSCFHE  DS    0XL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 JVSCIFID DS    0A          0  INTERFACE IDENTIFIER
2 JVSCFCTU DS    AL2          0  FUNCTION UNIT NUMBER
2 *
2 *          BIT 15  HEADER FLAG BIT,
2 *          MUST BE RESET UNTIL FURTHER NOTICE
2 *          BIT 14-12 UNUSED, MUST BE RESET
2 *          BIT 11-0  REAL FUNCTION UNIT NUMBER
2 JVSCFCT  DS    AL1          2  FUNCTION NUMBER
2 JVSCFCTV DS    AL1          3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 JVSCRET  DS    0A          4  GENERAL RETURN CODE
2 JVSCSRET DS    0AL2        4  SUB RETURN CODE
2 JVSCSR2  DS    AL1          4  SUB RETURN CODE 2
2 JVSCSR1  DS    AL1          5  SUB RETURN CODE 1
2 JVSCMRET DS    0AL2        6  MAIN RETURN CODE
2 JVSCMR2  DS    AL1          6  MAIN RETURN CODE 2
2 JVSCMR1  DS    AL1          7  MAIN RETURN CODE 1
2 JVSCFHL  EQU    8          8  GENERAL OPERAND LIST HEADER LENGTH
2 *
1 JVSCUNU1 DS    CL3          UNUSED
1 JVSCPCON DS    X          PARAM.-CONTROL
1 JVSCCORE EQU    X'80'      A(CONDITION) IN REGISTER
1 JVSCEREG EQU    X'40'      A(IDENTIFIER) IN REGISTER
1 JVSCPREG EQU    X'20'      POST IN REGISTER
1 JVSCCREG EQU    X'10'      COUNT IN REGISTER
1 JVSCPTPR EQU    X'02'      P2 CALLER

```

1	JVSCCOND	DS	A	A(CONDITION)
1		ORG	JVSCCOND	
1	JVSCCONR	DS	AL1	REG. CONT. A(COND.)
1		DS	AL3	NOT USED IN THIS CONTEXT
1	JVSCEIID	DS	A	A(IDENTIFIER)
1		ORG	JVSCEIID	
1	JVSCEIDR	DS	AL1	REG. CONT. A(IDENTIFIER)
1		DS	AL3	NOT USED IN THIS CONTEXT
1	JVSCPOST	DS	H	POST-VALUE
1		ORG	JVSCPOST	
1	JVSCPOSR	DS	AL1	REG. CONT. POST-VALUE
1		DS	AL1	NOT USED IN THIS CONTEXT
1	JVSCCNT	DS	H	COUNT-VALUE
1		ORG	JVSCCNT	
1	JVSCCNTR	DS	AL1	REG. CONT. COUNT-VALUE
1		DS	AL1	NOT USED IN THIS CONTEXT
1	JVSCJVS	DS	F	RESERVED
1	JVSC#	EQU	*-JVSCFHDR	LENGTH
1	*			
1	* RETURNCODES			
1	*			
1	JVSCOK	EQU	X'00000000'	SUCCESSFUL CALL
1	JVSCINAD	EQU	X'00010004'	INVALID PARM.-LIST
1	JVSCINEI	EQU	X'04010004'	INVALID IDENTIFIER
1	JVSCSYTE	EQU	X'08000004'	SYNTAX ERROR IN CONDITION
1	JVSCJVNA	EQU	X'10000004'	JV NOT ACCESSABLE
1	JVSCSYSE	EQU	X'14000004'	SYSTEM ERROR (\$GETMEM)
1	JVSCBOER	EQU	X'18000004'	SYSTEM ERROR (BOURSES)

RELJV

JV-LINK-Eintrag löschen

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **RELJV** löscht einen oder alle JV-LINK-Einträge aus der JV-LINK-Tabelle. Die Auswahl des zu löschenden Eintrags erfolgt über den Kettungsnamen.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
RELJV	$[\text{LINK}=\text{*jvlink}]$ $,MF=\left\{ \begin{array}{l} \underline{S} \\ \underline{C} \\ (E, \dots) \\ \underline{D} \\ \underline{L} \end{array} \right\}, \text{PREFIX}=\left\{ \begin{array}{l} \underline{IDJ} \\ \text{pre} \end{array} \right\}$

LINK=*jvlink gültiger Kettungsname einer Jobvariablen, (wie in DCLJV definiert), der aus der JV-LINK-Tabelle gelöscht werden soll. Der Kettungsname darf einschließlich des „*“ 8 Zeichen lang sein. Besteht der Kettungsname aus Leerzeichen, werden alle Einträge aus der JV-LINK-Tabelle gelöscht.

MF
PREFIX Zur Beschreibung der Operanden MF und PREFIX siehe [Seite 88](#). Die gültigen Werte sind zu Beginn der Makrobeschreibung dargestellt und aus dem Aufrufformat ersichtlich.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

RELJV    RELJV MF=D
1 *****
1 *          VERSION 310
1 *****
1 *          RELJV  PARAMETER  LIST                               *
1 *****
1          #INTF REFTYPE=REQUEST,                                C
1          INTNAME=RELJV,INTCOMP=001
1 RELJV    DSECT
1 *****
1 *          UNIT=41, FUNCTION=33, VERSION=1                      *
1 *****
1          FHDR    MF=(C, IDJR)
2          DS      OA
2 IDJRFHE  DS      OXL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 IDJRIFID DS      OA          0  INTERFACE IDENTIFIER
2 IDJRFCTU DS      AL2         0  FUNCTION UNIT NUMBER
2 *
2 *          BIT 15  HEADER FLAG BIT,
2 *          MUST BE RESET UNTIL FURTHER NOTICE
2 *          BIT 14-12 UNUSED, MUST BE RESET
2 *          BIT 11-0  REAL FUNCTION UNIT NUMBER
2 IDJRFCT  DS      AL1         2  FUNCTION NUMBER
2 IDJRFCTV DS      AL1         3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJRRET  DS      OA          4  GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJRSRET DS      OAL2         4  SUB RETURN CODE
2 IDJRSR2  DS      AL1         4  SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJRR2OK EQU    X'00'          All correct, no additional info
2 IDJRR2NA EQU    X'01'          Successful, no action was necessary
2 IDJRR2WA EQU    X'02'          Warning, particular situation
2 IDJRSR1  DS      AL1         5  SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *

```

```

2 * CLASS A      X'00'          FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B      X'01' - X'1F'  PARAMETER SYNTAX ERROR
2 * CLASS C      X'20'          INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D      X'40' - X'7F'  NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E      X'80' - X'82'  WAIT AND RETRY
2 *
2 IDJRRFSP EQU   X'00'          FUNCTION SUCCESSFULLY PROCESSED
2 IDJRRPER EQU   X'01'          PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJRRFNS EQU   X'01'          CALLED FUNCTION NOT SUPPORTED
2 IDJRRFNA EQU   X'02'          CALLED FUNCTION NOT AVAILABLE
2 IDJRRVNA EQU   X'03'          INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJRRRAER EQU  X'04'          ALIGNMENT ERROR
2 IDJRRRIER EQU  X'20'          INTERNAL ERROR
2 IDJRRCAR EQU   X'40'          CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJRRECR EQU   X'41'          SUBSYSTEM (SS) MUST BE CREATED
2 *                               EXPLICITELY BY CREATE-SS
2 IDJRRECN EQU   X'42'          SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJRRWAR EQU   X'80'          WAIT FOR A SHORT TIME AND RETRY
2 IDJRRWLR EQU   X'81'          "      LONG      "
2 IDJRRWUR EQU   X'82'          WAIT TIME IS UNCALCULABLY LONG
2 *                               BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJRRTNA EQU   X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJRRDH EQU    X'82'          SS IN DELETE / HOLD
2 *
2 IDJRMRET DS    0AL2           6  MAIN RETURN CODE
2 IDJRMR2 DS     AL1            6  MAIN RETURN CODE 2
2 IDJRMR1 DS     AL1            7  MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 IDJRRLNK EQU   X'FFFF'        LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJRFHL EQU    8              8  GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL RELJV PARAMETER LIST *
1 *****
1 IDJRHDR1 EQU   X'00292101',4
1 IDJRJVL DS     CL8              JV LINKNAME
1 IDJRFLG DS     XL4              FLAG RESERVED
1 IDJRJVS DS     A                RESERVED
1 IDJRPLLN EQU   *-RELJV          LENGTH OF DSECT
1 *****
1 SPACE

```

SETJV

Jobvariable setzen

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **SETJV** ordnet einer Jobvariablen einen Wert zu, der in einem Bereich des Benutzerprogramms gespeichert ist.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
SETJV	$\left\{ \begin{array}{l} \text{jvid1} \\ (\text{jvid1}[\text{,}[\text{start}] \text{,}[\text{länge}]]) \end{array} \right\}$ $\left\{ \begin{array}{l} \text{,bereich} \\ \text{,EXPR=jvid2} \end{array} \right\}$ $[\text{,PASS=kennwort}]$ $\text{,VERSION}=\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right\}, \text{,PARMOD}=\left\{ \begin{array}{l} 24 \\ 31 \end{array} \right\}, \text{,MF}=\left\{ \begin{array}{l} \text{S} \\ \text{C} \\ (\text{E}, \dots) \\ \text{D} \\ \text{L} \end{array} \right\}, \text{,PREFIX}=\left\{ \begin{array}{l} \text{IDJ} \\ \text{pre} \end{array} \right\}$

jvid1	identifiziert die Jobvariable. jvid kann sein:
	<ul style="list-style-type: none"> jvname Vollqualifizierter Pfadname einer permanenten oder temporären Jobvariablen. *jvlink Gültiger Jobvariablen-Kettungsname.
start	Startposition (erstes zu änderndes Byte) im JV-Wert.

länge	Anzahl Zeichen, die gesetzt werden sollen.
bereich	Adresse eines Bereiches im Benutzerprogramm, der den Jobvariablenwert enthält. Der Bereich muss mit einem vier Byte großen Längenfeld beginnen, dessen erstes Halbwort die Länge des Wertes + vier Byte Längenfeld enthält.
EXPR=jvid2	gibt an, dass die Jobvariable „jvid“ auf den Wert der Jobvariablen „jvid2“ gesetzt werden soll. Für „jvid2“ sind die gleichen Jobvariablen-Namen wie für „jvid“ sowie zusätzlich die von Sonder-Jobvariablen zulässig.
PASS=kennwort	Lese- oder Schreibkennwort.
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung dargestellt und aus dem Aufrufformat ersichtlich.
VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \leq V8.7$. Der Operand MF=D/C zur Erzeugung einer DSECT bzw. CSECT wird dabei nicht unterstützt (siehe Hinweis DSECT).
=1	Die Makroauflösung entspricht $JV \geq V10.0$.
PARMOD	steuert die Makroauflösung. Es wird entweder die 24-Bit- oder die 31-Bit-Schnittstelle generiert. PARMOD wird nur mit VERSION=0 ausgewertet. Wenn PARMOD nicht spezifiziert wird, erfolgt die Makroauflösung entsprechend der Angabe für den Makro GPARMOD oder der Voreinstellung für den 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.

Hinweis zur Operandenliste

Im Gegensatz zur Kommandoebene darf in der Operandenliste die Länge 0 spezifiziert werden, die dann die gesamte Jobvariablen-Länge repräsentiert.

Hinweise zur DSECT

- Der Aufruf des Makros SETJV mit den Operanden MF=D und VERSION=1 erzeugt eine DSECT für die Operandenliste des SETJV-Makros (VERSION=1).
- Eine CSECT/DSECT für den Makro mit VERSION=0 wird mit Aufruf des Makros IDJSE [D][,prefix] [,PARMOD=24/31] erzeugt.

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

    SETJV    SETJV MF=D,VERSION=1
1 SETJV    $SETJV  ,.EXPR=,PASS=NONE,ENCR=YES,MF=D,          C
1          PARMOD=,VERSION=1,CALLER=USER,                  C
1          PREFIX=IDJ,JVTYPE=STRING,NULLSTR=NO
2 *****
2 *        VERSION 203
2 *****
2 *        S E T J V   P A R A M E T E R   L I S T          *
2 *****
2          #INTF REFTYPE=REQUEST,                            C
2          INTNAME=SETJV,INTCOMP=002
2 SETJV    DSECT
2 *****
2 *        UNIT=41, FUNCTION=1,  VERSION=2  (V10.0)        *
2 *****
2          FHDR    MF=(C,IDJS)
3          DS     0A
3 IDJSFHE  DS     0XL8          0  GENERAL PARAMETER AREA HEADER
3 *
3 IDJSIFID DS     0A          0  INTERFACE IDENTIFIER
3 IDJSFCTU DS     AL2         0  FUNCTION UNIT NUMBER
3 *
3 *          BIT 15  HEADER FLAG BIT,
3 *          MUST BE RESET UNTIL FURTHER NOTICE
3 *          BIT 14-12 UNUSED, MUST BE RESET
3 *          BIT 11-0  REAL FUNCTION UNIT NUMBER
3 IDJSFCT  DS     AL1         2  FUNCTION NUMBER
3 IDJSFCTV DS     AL1         3  FUNCTION INTERFACE VERSION NUMBER
3 *
3 IDJSRET  DS     0A          4  GENERAL RETURN CODE

```



```

3 *
3 * GENERAL_RETURN_CODE CLEARED (X'0000000') MEANS
3 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
3 *
3 IDJSSRET DS    OAL2                4  SUB RETURN CODE
3 IDJSSR2  DS    AL1                4  SUB RETURN CODE 2
3 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
3 * Standard subcode2 values as defined by convention:
3 IDJSR2OK EQU  X'00'                All correct, no additional info
3 IDJSR2NA EQU  X'01'                Successful, no action was necessary
3 IDJSR2WA EQU  X'02'                Warning, particular situation
3 IDJSSR1  DS    AL1                5  SUB RETURN CODE 1
3 *
3 * GENERAL INDICATION OF ERROR CLASSES
3 *
3 * CLASS A    X'00'                FUNCTION WAS SUCCESSFULLY PROCESSED
3 * CLASS B    X'01' - X'1F'        PARAMETER SYNTAX ERROR
3 * CLASS C    X'20'                INTERNAL ERROR IN CALLED FUNCTION
3 * CLASS D    X'40' - X'7F'        NO CLASS SPECIFIC REACTION POSSIBLE
3 * CLASS E    X'80' - X'82'        WAIT AND RETRY
3 *
3 IDJSRFSP EQU  X'00'                FUNCTION SUCCESSFULLY PROCESSED
3 IDJSRPER EQU  X'01'                PARAMETER SYNTAX ERROR
3 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
3 IDJSRFNS EQU  X'01'                CALLED FUNCTION NOT SUPPORTED
3 IDJSRFNA EQU  X'02'                CALLED FUNCTION NOT AVAILABLE
3 IDJSRVNA EQU  X'03'                INTERFACE VERSION NOT SUPPORTED
3 *
3 IDJSRAER EQU  X'04'                ALIGNMENT ERROR
3 IDJSRIER EQU  X'20'                INTERNAL ERROR
3 IDJSRCAR EQU  X'40'                CORRECT AND RETRY
3 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
3 IDJSRECR EQU  X'41'                SUBSYSTEM (SS) MUST BE CREATED
3 *                                     EXPLICITELY BY CREATE-SS
3 IDJSRECN EQU  X'42'                SS MUST BE EXPLICITELY CONNECTED
3 *
3 IDJSRWAR EQU  X'80'                WAIT FOR A SHORT TIME AND RETRY
3 IDJSRWLR EQU  X'81'                "          LONG          "
3 IDJSRWUR EQU  X'82'                WAIT TIME IS UNCALCULABLY LONG
3 *                                     BUT RETRY IS POSSIBLE
3 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
3 IDJSRTNA EQU  X'81'                SS TEMPORARILY NOT AVAILABLE
3 IDJSRDH  EQU  X'82'                SS IN DELETE / HOLD
3 *
3 IDJSMRET DS    OAL2                6  MAIN RETURN CODE
3 IDJSMR2  DS    AL1                6  MAIN RETURN CODE 2
3 IDJSMR1  DS    AL1                7  MAIN RETURN CODE 1
3 *

```

```

3 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XYYYY')
3 *
3 IDJSRLNK EQU   X'FFFF'           LINKAGE ERROR / REQ. NOT PROCESSED
3 IDJSFHL EQU   8                   8   GENERAL OPERAND LIST HEADER LENGTH
3 *
2 *****
2 * END OF STANDARD HEADER. START OF SPECIAL SETJV PARAMETER LIST      *
2 *****
2 IDJSHDRI      EQU   X'00290102',4
2 IDJSJV        DS    CL54                JVNAME (JV TO BE SET)
2 IDJSPOS       DS    H                   SUBSTRING START POSITION
2 IDJSLEN       DS    H                   SUBSTRING LENGTH
2              DS    H                   RESERVED
2 IDJSPASS      DS    CL4                 PASSWORD
2              DS    XL12                RESERVED
2 IDJSAR31     DS    A                   SET VALUE ADDR
2 IDJSLENV      DS    CL1                 LENGTH OF SET VALUE
2 *              ONLY USED BY CMD PROCESSING
2 IDJSJV2       DS    CL54                JVNAME2 (JV TO BE READ)
2              DS    XL1                 ALIGNMENT
2 IDJSPOS2     DS    H                   SUBSTRING START POSITION
2 IDJSLEN2     DS    H                   SUBSTRING LENGTH
2 IDJSOP       DS    CL1                 RESERVED
2 IDJSF        DS    X                   FLAGS
2 IDJSFSE1     EQU   X'80'               7-7 PRIV SETJV REQUEST
2 *              0=P1 CALLER,1=P2 CALLER
2 IDJSFRD      EQU   X'40'               6-6 TYPE=READ
2 IDJSFWT      EQU   X'20'               5-5 TYPE=WRITE
2 IDJSFUN      EQU   X'10'               4-4 TYPE=SHLOCK
2 IDJSFSH      EQU   X'08'               3-3 PROT=SHARE
2 IDJSFEX      EQU   X'04'               2-2 PASSWD IS GIVEN
2 IDJSRW       EQU   X'02'               1-1 RDPSTYPE=READ
2 IDJNOTF      EQU   X'01'               0-0 NOTIF=NO
2 IDJSF1       DS    X                   FLAGS
2 IDJSENCR     EQU   X'80'               7-7 0=YES, 1=NO
2 *              (ENCRYPTION)
2 IDJSF1NV     EQU   X'40'               6-6 0=NO, 1=YES
2 *              (NUMERIC VALUE)
2 IDJSJVIX     EQU   X'20'               5-5 0=NO, 1=YES
2 *              (JVID (1 OR 2) INDEXED)
2 *              4-4 RESERVED FOR MONJV-HANDLER
2 IDJSECT      EQU   X'08'               3-3 1=SET BY CMD PROCESSING
2 IDJSF1BV     EQU   X'04'               2-2 0=NO, 1=YES
2 *              (BOOLEAN VALUE)
2 IDJSNSTR     EQU   X'02'               1-1 0=NO, 1=YES
2 *              (NULLSTRING DEFINED)
2              DS    XL1                 ALIGNMENT
2 IDJSCTAD     DS    A                   CATALOG ENTRY ADDR

```

```
2 IDJSJVER      DS      CL54              JV IN ERROR
2              DS      XL2              RESERVED
2 IDJSJVS       DS      A                RESERVED
2 IDJSPLLN      EQU     *-SETJV          LENGTH OF DSECT
2 *****
2              SPACE
```

STAJV Merkmale von Jobvariablen ausgeben

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-/E-/L-Form)
siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand PREFIX“ auf Seite 89](#)).

Voreinstellung: PREFIX = IDJ

Makrobeschreibung

Der Makro **STAJV** überträgt die Merkmale einer Jobvariablen in einen Anwenderbereich, bzw. erstellt in diesem Bereich eine Namensliste von Jobvariablen.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
STAJV	<p>[jvid]</p> <p>,bereich [,größe]</p> <p>,OUTPUT=$\left\{ \begin{array}{c} \text{OLD} \\ \text{NEW} \end{array} \right\}$ [,LIST=$\left\{ \begin{array}{c} \text{OLD} \\ \text{NEW} \end{array} \right\}$],TIMBASE=$\left\{ \begin{array}{c} *UTC \\ *LTI \end{array} \right\}$</p> <p>,ODSECT=$\left\{ \begin{array}{c} \text{NO} \\ \text{YES} \end{array} \right\}$ [,SELADDR=auswahl],SORT=$\left\{ \begin{array}{c} *JVNAM \\ \text{NO} \end{array} \right\}$</p> <p>,VERSION=$\left\{ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \right\}$,PARMOD=$\left\{ \begin{array}{c} 24 \\ 31 \end{array} \right\}$,MF=$\left\{ \begin{array}{c} S \\ C \\ (E, \dots) \\ D \\ L \end{array} \right\}$</p> <p>,PREFIX=$\left\{ \begin{array}{c} \text{IDJ} \\ \text{pre} \end{array} \right\}$</p>

jvid	<p>bezeichnet die Jobvariable, deren Merkmale in den Anwenderbereich übertragen werden. Sonder-Jobvariablen sind nicht möglich. jvid kann sein:</p> <p>jvname ein voll- oder teilqualifizierter Pfadname einer permanenten oder temporären Jobvariablen. Die Verwendung von Musterzeichen (Wildcards) ist für Versionen > 0 zulässig. Nur mit der vollqualifizierten Angabe werden die Merkmale einer Jobvariablen in den Anwenderbereich übertragen, ansonsten wird eine Liste der Jobvariablenennamen im Anwenderbereich abgelegt. Gibt ein Anwender eine fremde Benutzerkennung an, so werden nur die Namen der Jobvariablen in den Anwenderbereich übertragen, die laut Katalogeintrag einen fremden Zugriff gestatten. Dies ist SHARE=YES bzw. R oder W für GROUP bzw. OTHERS wenn ein BASIC-ACL-Schutz besteht.</p> <p>*jvlink ein gültiger Jobvariablen-Kettungsname.</p> <p>temp Eine Namensliste aller temporären Jobvariablen des Auftrags wird ausgegeben. temp steht für das mit dem Systemparameter TEMPFILE definierte Sonderzeichen zur Kennzeichnung von temporären Dateien und Jobvariablen (im Bedarfsfall die Systembetreuung fragen).</p>
bereich	<p>Dieser Eintrag bezeichnet den Anwenderbereich, in den die Informationen übertragen werden sollen (siehe Funktionsweise).</p>
größe	<p>legt die Länge des Bereichs in Byte fest. Bis zu VERSION=3 beträgt die maximale Länge 32767 Byte. Die Länge des Bereichs ist abhängig vom Ausgabeformat:</p> <p>OUTPUT=OLD :</p> <ul style="list-style-type: none">– $\text{größe} \geq 60$ Byte (Standardwert 60 Byte) <p>OUTPUT=NEW:</p> <ul style="list-style-type: none">– $\text{größe} \geq 80$ Byte (Standardwert 80 Byte); bei Angabe von VERSION=2 (JV V10.0)– $\text{größe} \geq 120$ Byte (Standardwert 120 Byte); bei Angabe von VERSION=3 (JV V11.0 und V11.2)– $\text{größe} \geq 160$ Byte (Standardwert 160 Byte); bei Angabe von VERSION=4 (JV \geq V12.0)

OUTPUT	<p>gibt an, in welchem Format die Ausgabe für eine Jobvariable angefordert wird, deren Name <i>vollqualifiziert</i> angegeben ist. <i>Diese Angabe ist nur ab Version 2 und mit MF=L oder MF=S erlaubt.</i></p>
= <u>OLD</u>	<p>ist Voreinstellung: Die angeforderte Information enthält nur den festen Teil des Katalogeintrags, aber <i>ohne</i> den Namen der Jobvariablen (60 Byte). Eine DSECT für dieses Ausgabeformat wird mit dem Makro IDJE erzeugt (siehe Hinweis DSECT).</p>
=NEW	<p>Die Information wird im neuen Format angefordert.</p> <ul style="list-style-type: none">– Bei Angabe von VERSION=2 wird die Information im Layout von JV V10.0 ausgegeben (≥ 80 Byte).– Bei Angabe von VERSION=3 wird die Information im Layout von JV V11.0 bzw. JV V11.2 ausgegeben (≥ 120 Byte).– Bei Angabe von VERSION=4 wird die Information im Layout von JV ≥ V12.0 ausgegeben (≥ 160 Byte).
LIST	<p>entscheidet über das Ausgabeformat im Anwenderbereich. Dieser Operand wirkt sich nur bei Version ≥ 1 und <i>teilqualifizierter</i> Angabe des Pfadnamens aus (siehe Funktionsweise).</p>
= <u>OLD</u>	<p>ist die Voreinstellung und bedeutet, dass das alte Ausgabeformat (ohne :catid:\$userid.) unterstützt wird.</p>
=NEW	<p>gibt den vollständigen Pfadnamen aus (mit :catid:\$userid.).</p>
TIMBASE	<p>entscheidet über die Ausgabebasis von Erstellungs- und Lösungsdatum. Dieser Operand hat nur Auswirkungen, wenn VERSION=4 und OUTPUT=NEW angegeben sind.</p>
=* <u>UTC</u>	<p>ist Voreinstellung: Die Zeitbasis ist die UTC-Zeit (Weltzeit)</p>
=* <u>LTI</u>	<p>gibt an, dass die lokale Zeit im aufrufenden Rechnersystem als Zeitbasis verwendet wird.</p>
ODSECT	<p>gibt an, für welches Ausgabeformat eine DSECT zu erzeugen ist. <i>Angabe nur mit MF=D erlaubt.</i></p>
= <u>NO</u>	<p>ist Voreinstellung: Die DSECT für das alte Ausgabeformat kann mit dem Makro IDJE erzeugt werden (siehe OUTPUT=OLD).</p>
=YES	<p>Die DSECT wird für das neue Ausgabeformat (80/120/160 Byte; siehe OUTPUT=NEW) erzeugt.</p>

SELADDR	beschränkt die Menge der Jobvariablen, die durch den voll- bzw. teilqualifizierten Pfadnamen jvid angegeben ist, durch weitere Auswahlkriterien. Ohne Angabe des Operanden wird das Feld IDJFJVS auf Null gesetzt und es findet keine weitere Selektion statt. Die Angabe ist nur möglich bei Version ≥ 2 .
=auswahl	symbolische Adresse der Parameterliste, die durch Aufruf des JVSEL-Makros (siehe Seite 136ff) mit den gewünschten Auswahlkriterien erzeugt wurde. Siehe auch „ Funktionsweise “ auf Seite 176 .
SORT	bestimmt die Sortierung der Katalogeinträge/Pfadnamen in der Ausgabe.
=*JVNAM	ist Voreinstellung: Die Katalogeinträge/Pfadnamen werden alphabetisch sortiert ausgegeben.
=NO	Die Katalogeinträge/Pfadnamen werden in der Reihenfolge ausgegeben, wie sie im Katalog stehen.
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung dargestellt und aus dem Aufrufformat ersichtlich. Der Operand MF=D/C zur Erzeugung einer DSECT bzw. CSECT wird erst ab VERSION=2 unterstützt (siehe Hinweis DSECT).
VERSION	legt fest, welcher Version die Makroauflösung entsprechen soll.
=0	ist Voreinstellung: Die Makroauflösung entspricht dem Stand JV < V8.7. In dieser Version werden folgende Angaben noch nicht unterstützt: <ul style="list-style-type: none">– MF=C/D (siehe Hinweis DSECT)– LIST=NEW– OUTPUT=NEW– SELADDR=
=1	Die Makroauflösung entspricht dem Stand JV V8.7. In dieser Version werden folgende Angaben noch nicht unterstützt: <ul style="list-style-type: none">– MF=C/D (siehe Hinweis DSECT)– OUTPUT=NEW– SELADDR=
=2	Die Makroauflösung entspricht dem Stand JV V10.0.
=3	Die Makroauflösung entspricht dem Stand JV V11.0 und V11.2.
=4	Die Makroauflösung entspricht dem Stand JV \geq V12.0.

PARMOD	steuert die Makroauflösung. Es wird entweder die 24-Bit- oder die 31-Bit-Schnittstelle generiert. PARMOD wird nur mit VERSION=0 ausgewertet. Wenn PARMOD nicht spezifiziert wird, erfolgt die Makroauflösung entsprechend der Angabe für den Makro GPARMOD oder der Voreinstellung für den 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.

Funktionsweise

Bei Angabe eines *vollqualifizierten* Jobvariablen-Namens werden die Kontrollinformationen aus dem Jobvariablen-Katalogeintrag in den angegebenen Bereich übertragen. Der Operand OUTPUT bestimmt dabei das Format der Ausgabe.

Die DSECT für den Katalogeintrag kann mit MF=D erzeugt werden. Das Kennwortfeld wird auf binär null gesetzt. Ist der angegebene Bereich zu klein um die Informationen aufzunehmen, so erhält das Programm den Returncode X'0490' in den rechten beiden Bytes von Register 15.

Wird der Jobvariablen-Name *nicht vollqualifiziert* angegeben (teilqualifiziert oder mit Musterzeichen) oder wird das Sonderzeichen für temporäre Jobvariablen angegeben, wird eine Liste der Jobvariablen-Namen in den angegebenen Bereich übertragen. Das Format der übertragenen Information ist abhängig von der Angabe im Operanden LIST:

	LIST=OLD	LIST=NEW	
namenslänge ₁	1 Byte	1 Byte	
jvname ₁ , <u>ohne</u> catid, userid	1 – 41 Zeichen	14–54 Zeichen	jvname, <u>mit</u> catid, userid
.....			
namenslänge _n	1 Byte	1 Byte	
jvname _n <u>ohne</u> catid, userid	1 – 41 Zeichen	14–54 Zeichen	<u>mit</u> catid, userid
Ende	1 Byte(X'00'/X'01')	1 Byte(X'00'X'01')	

„namenslänge“ bezeichnet die Länge des zugehörigen Jobvariablen-Namens (plus ein Zeichen Längensfeld). Das ordnungsgemäße Ende der Liste wird durch X'00' gekennzeichnet. Ist der Bereich nicht ausreichend, um alle Jobvariablen-Namen aufzunehmen, so wird das letzte Byte auf X'01' gesetzt.

Bei Angabe eines teilqualifizierte Jobvariablen-Namens wird in der Parameterliste des STAJV-Aufrufs zusätzlich ein entsprechender Wert zurückgegeben (IDJPQFN=X'20').

Zusätzliche Selektion der Jobvariablen

Die Menge der zu übertragenden Jobvariablen-Namen kann auf Jobvariablen mit bestimmten Eigenschaften beschränkt werden. Bei der Angabe eines vollqualifizierten Jobvariablen-Namens kann das Übertragen der Kataloginformationen von bestimmten Eigenschaften abhängig gemacht werden.

Die gewünschten Auswahlkriterien werden in einem JVSEL-Makroaufruf spezifiziert. Im STAJV-Aufruf muss dann im Operanden SELADDR die symbolische Adresse der erzeugten JVSEL-Parameterliste angegeben werden. Wird SELADDR nicht angegeben, wird das Feld IDJFJVS auf Null gesetzt (X'00000000'=keine Auswahl nach Selektionskriterien).

Hinweis

Werden keine Operanden angegeben, so wird eine Liste aller Jobvariablen ausgegeben, die im Standard-Katalog der Benutzererkennung stehen, unter der auch der Auftrag abläuft.

Hinweise zur DSECT

- Der Aufruf des Makros STAJV mit den Operanden MF=D und VERSION=2/3/4 erzeugt eine DSECT für die Operandenliste des STAJV-Makros (VERSION=2/3/4). Mit der Angabe ODSECT=YES wird zusätzlich eine DSECT für die Ausgabe eines Katalogeintrages erzeugt.
- Für VERSION=0 bzw. VERSION=1 gilt:
Der Makroaufruf IDJE [D][,prefix] erzeugt eine CSECT/DSECT für den Katalogeintrag.
Der Makroaufruf IDJST [D][,prefix][,PARMOD=24/31][,VERSION=0/1] erzeugt eine CSECT/DSECT für die Operandenliste des STAJV-Makros.

Rückinformation und Fehleranzeige

siehe [Seite 229](#).

DSECT

```

STAJV    STAJV MF=D,VERSION=4,ODSECT=YES
1 *****
1 *          VERSION 400
1 *****
1 *          S T A J V    P A R A M E T E R    L I S T          *
1 *****
1          #INTF REFTYPE=REQUEST,                                C
1          INTNAME=STAJV,INTCOMP=005
1 STAJV    DSECT
1 *****
1 *          UNIT=41, FUNCTION=2,  VERSION=<PARAMETER VERSION>  *
1 *****
1          FHDR  MF=(C,IDJF)
2          DS   OA
2 IDJFFHE  DS   OXL8          0  GENERAL PARAMETER AREA HEADER
2 *
2 IDJFIFID DS   OA          0  INTERFACE IDENTIFIER
2 IDJFFCTU DS   AL2          0  FUNCTION UNIT NUMBER
2 *
2 *          BIT 15  HEADER FLAG BIT,
2 *          MUST BE RESET UNTIL FURTHER NOTICE
2 *          BIT 14-12 UNUSED, MUST BE RESET
2 *          BIT 11-0  REAL FUNCTION UNIT NUMBER
2 IDJFFCT  DS   AL1          2  FUNCTION NUMBER
2 IDJFFCTV DS   AL1          3  FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJFRET  DS   OA          4  GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJFSRET DS   OAL2          4  SUB RETURN CODE
2 IDJFSR2  DS   AL1          4  SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJFR2OK EQU  X'00'          All correct, no additional info
2 IDJFR2NA EQU  X'01'          Successful, no action was necessary
2 IDJFR2WA EQU  X'02'          Warning, particular situation
2 IDJFSR1  DS   AL1          5  SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A    X'00'          FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B    X'01' - X'1F'  PARAMETER SYNTAX ERROR
2 * CLASS C    X'20'          INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D    X'40' - X'7F'  NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E    X'80' - X'82'  WAIT AND RETRY

```

```

2 *
2 IDJFRFSP EQU X'00'          FUNCTION SUCCESSFULLY PROCESSED
2 IDJFRPER EQU X'01'          PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJFRFNS EQU X'01'          CALLED FUNCTION NOT SUPPORTED
2 IDJFRFNA EQU X'02'          CALLED FUNCTION NOT AVAILABLE
2 IDJFRVNA EQU X'03'          INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJFRAER EQU X'04'          ALIGNMENT ERROR
2 IDJFRIER EQU X'20'          INTERNAL ERROR
2 IDJFRCAR EQU X'40'          CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJFRECR EQU X'41'          SUBSYSTEM (SS) MUST BE CREATED
2 *                               EXPLICITELY BY CREATE-SS
2 IDJFRECN EQU X'42'          SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJFRWAR EQU X'80'          WAIT FOR A SHORT TIME AND RETRY
2 IDJFRWLR EQU X'81'          "      LONG      "
2 IDJFRWUR EQU X'82'          WAIT TIME IS UNCALCULABLY LONG
2 *                               BUT RETRY IS POSSIBLE
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJFRTNA EQU X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJFRDH EQU X'82'          SS IN DELETE / HOLD
2 *
2 IDJFMRET DS OAL2           6 MAIN RETURN CODE
2 IDJFMR2 DS AL1            6 MAIN RETURN CODE 2
2 IDJFMR1 DS AL1            7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XYYYY')
2 *
2 IDJFRLNK EQU X'FFFF'          LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJFFHL EQU 8                8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL STAJV PARAMETER LIST *
1 *****
1 IDJFHDRI EQU X'00290205',4
1 IDJAR31 DS F                AREA ADDRESS(31 BIT FORMAT)
1 IDJSIZE DS F                AREA SIZE
1 IDJFLAGS DS X              FLAGS
1 IDJPASSW EQU X'00'          7-7 S NOT USED (DEL. V9.5)
1 IDJNAREA EQU X'40'          6-6 S ADDR NOT GIVEN
1 *                               SET BY JVSTAEX
1 IDJPQFN EQU X'20'          5-5 S PART. QUAL. JVNAME
1 *                               SET BY JVSTAEX
1 *                               FOR CMD PROCESSING
1 IDJFECT EQU X'10'          4-4 S SET BY CMD PROCESSING
1 IDJLNEW EQU X'08'          3-3 S OUTPUT JVNAME LONG

```

```

1 IDJFCE      EQU  X'04'          2-2 S NEW CE OUTPUT V10
1 IDJFP2      EQU  X'02'          1-1 S P2 CALLER
1 IDJFCE11    EQU  X'01'          0-0 S NEW CE OUTPUT V11
1 IDJFLAG1    DS    X              FLAGS
1 IDJFCE12    EQU  X'80'          7-7 S NEW CE OUTPUT V12
1 IDJFTLTI    EQU  X'40'          6-6 S TIME-BASE = LOCAL
1 IDJFSORT    EQU  X'20'          5-5 S SORT      = NO
1             DS    X              RESERVED
1             DS    CL16           RESERVED
1 IDJFJV      DS    CL80           JV-NAME
1             DS    CL41           RESERVED
1 * * * * *
1 * THE FOLLOWING FIELD IS USED ONLY FOR CMD PROCESSING *
1 * * * * *
1 IDJSRTAD    DS    A              ADDRESS OF JVNAME
1 IDJFJVS     DS    A              ADDRESS OF SELECTION PL
1 IDJFPLLN    EQU  *-STAJV        LENGTH OF DSECT
1             SPACE
1 *****
1 *           S T A J V   O U T P U T           *
1 *****
1 IDJES      DSECT
1 IDJEDMS    DS    CL1            0 DMS INDICATOR
1 IDJESJV    EQU  X'04'          S SHARE = YES
1 IDJEROA    EQU  X'08'          S READ ONLY ACCESS
1 IDJERPP    EQU  X'10'          S READ PASSWORD SPECIFIED
1 IDJEWPP    EQU  X'20'          S WRITE PASSWORD SPECIFIED
1 IDJEDM2    DS    CL1            1 DMS INDICATOR2
1 IDJENSH    EQU  X'80'          S NON-SHARE FLAG
1 IDJECON    EQU  X'40'          S CONVERTED CE
1 IDJENCR    EQU  X'01'          S PASSWORD IS ENCRYPTED
1 IDJERD     DS    CL4            2 READ PASSWORD
1 IDJEWRT    DS    CL4            6 WRITE PASSWORD
1 IDJECRD    DS    OCL14         10 CREATION DATE
1 IDJECRY    DS    CL4            YEAR
1 IDJECRM    DS    CL2            MONTH
1 IDJECRT    DS    CL2            DAY
1 IDJECRH    DS    CL2            HOURS
1 IDJECRI    DS    CL2            MINUTES
1 IDJECRS    DS    CL2            SECONDS
1 IDJEEXD    DS    OCL14         24 EXPIRATION DATE
1 IDJEEXY    DS    CL4            YEAR
1 IDJEEXM    DS    CL2            MONTH
1 IDJEEXT    DS    CL2            DAY
1 IDJEEXH    DS    CL2            HOURS
1 IDJEEXI    DS    CL2            MINUTES
1 IDJEEXS    DS    CL2            SECONDS
1 IDJEAACL   DS    CL1            38 ACCESS CONTROL LIST

```

```

1 IDJEMAC      EQU  X'80'      S BASIC ACL PRESENT
1 IDJEGACT     EQU  X'40'      S GUARD PRESENT
1 IDJEURD     EQU  X'20'      S USER: READ-PRIVILEG
1 IDJEUWR     EQU  X'10'      S USER: WRITE-PRIVILEG
1 IDJEGRD     EQU  X'08'      S GROUP: READ-PRIVILEG
1 IDJEGWR     EQU  X'04'      S GROUP: WRITE-PRIVILEG
1 IDJEORD     EQU  X'02'      S OTHERS: READ-PRIVILEG
1 IDJEOWR     EQU  X'01'      S OTHERS: WRITE-PRIVILEG
1 IDJEDIV     DS    CL1        39 ADDITIONAL CHARASTERISTICS
1 IDJETYP     EQU  X'80'      S NUMERIC-TYPE
1 IDJEMJV     EQU  X'40'      S MONJV
1 IDJETEMP    EQU  X'20'      S TEMPORARY INDICATOR
1 IDJENSTD    EQU  X'10'      S NULL STRING DEFINED
1 IDJEBLID    EQU  X'08'      S BOOLEAN-TYPE
1 IDJECAT     DS    CL4        40 GUARD: CATID
1 IDJEREAD    DS    CL18       44 GUARD: READ
1 IDJEWRT     DS    CL18       62 GUARD: WRITE
1 IDJEMANC    DS    CL8        80 MANAGEMENT CLASS
1 IDJEVLL     DS    CL2        88 JV VALUE LENGTH
1 IDJECLN     DS    CL1        90 CATID LENGTH
1 IDJEULN     DS    CL1        91 USERID LENGTH
1 IDJEJLN     DS    CL1        92 JV NAME LENGTH
1 IDJEPLN     DS    CL1        93 PATHNAME LENGTH
1 IDJENAM     DS    CL54       94 JV NAME
1             DS    CL12       148 RESERVED
1 IDJELEN     EQU  *-IDJEDMS    LENGTH OF FIXED PORTION
1 *****
1             SPACE

```

TERM

Programm und Prozedurabschnitt beenden

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/E-/L-Form)

Makrobeschreibung

Der TERM-Makro führt folgende Funktionen durch:

- Programm beenden (Standardwert)
- Programm und Prozedurabschnitt beenden (Operand UNIT=STEP)
- Speicherauszug ausgeben (Operand DUMP)
- Rückkehrcode an programmüberwachende Jobvariable übergeben (Operand URETCD)

Vor Programmbeendigung werden alle laufenden Ein-/Ausgabe-Operationen beendet.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
TERM	$\text{UNIT} = \left\{ \begin{array}{c} \text{PRGR} \\ \text{STEP} \end{array} \right\}, \text{DUMP} = \left\{ \begin{array}{c} \text{N} \\ \text{Y} \end{array} \right\}, \text{MODE} = \left\{ \begin{array}{c} \text{N[NORMAL]} \\ \text{A[BNORMAL]} \end{array} \right\}$ $[, \text{URETCD} = \left\{ \begin{array}{c} \text{adr} \\ (r) \\ \text{code} \end{array} \right\}]$ $, \text{MF} = \left\{ \begin{array}{c} \text{S} \\ (\text{E}, \dots) \\ \text{L} \end{array} \right\}$

UNIT bestimmt, ob bei der Beendigung des Programms nach Betriebsarten unterschieden werden soll.

=PRGR

ist Voreinstellung: Das Programm wird beendet.

- `=STEP` beendet das Programm, wobei unterschieden wird, in welcher Betriebsart das Programm gelaufen ist.
- Dialogbetrieb:
Wurde das Programm in einer Nicht-S-Prozedur aufgerufen, so verzweigt das System zusätzlich zum nächsten SET-JOB-STEP-, EXIT-JOB- oder CANCEL-PROCEDURE-Kommando. Wurde das Programm in einer S-Prozedur aufgerufen, so löst das System zusätzlich die SDF-P-Fehlerbehandlung aus.
 - Batchbetrieb (ENTER-Prozedur):
Das System verzweigt zusätzlich zum nächsten SET-JOB-STEP- oder EXIT-JOB-Kommando.

Hinweis

Folgende Angaben werden empfohlen:
 UNIT=PRGR mit MODE=NORMAL
 UNIT=STEP mit MODE=ABNORMAL

- DUMP entscheidet, ob ein Speicherauszug ausgegeben werden soll.
- `=N` ist Voreinstellung: Ein Speicherauszug wird nicht ausgegeben.
- `=Y` gibt einen Speicherauszug aus, sofern kein Kommando MODIFY-TEST-OPTIONS mit DUMP=NO gegeben wurde.
- MODE bestimmt die Art der Beendigung des Programms.
- `=N[NORMAL]` ist Voreinstellung: Das Programm soll normal beendet werden. Ist eine programmüberwachende Jobvariable definiert, so wird ihre Zustandsanzeige auf C'\$T▯'gesetzt.
- `=A[BNORMAL]` Das Programm soll abnormal beendet werden. Die Meldung
 ABNORMAL PROGRAM TERMINATION (&00)“ wird ausgegeben.
 (&00)= NRT0001, wenn UNIT=PRGR angegeben wurde
 (&00)= NRT0101, wenn UNIT=STEP angegeben wurde
- Für Anwender von Jobvariablen:
 Ist eine programmüberwachende Jobvariable definiert, so wird ihre Zustandsanzeige auf \$A gesetzt.

Hinweis

siehe Operand UNIT.

URETCD	bestimmt den Rückkehrcode, der an die programmüberwachende Jobvariable übergeben wird (linksbündig, Byte 4 bis 7). Ist keine programmüberwachende Jobvariable definiert, wird der Operand ignoriert. Fehlt der Operand, so wird der Wert C'▯▯▯▯' in die programmüberwachende Jobvariable übergeben.
=code	gibt einen 1 bis 4 Byte langen alphanumerischen Wert, dezimal (C'cccc') oder sedezimal (X'xxxxxxxx') an.
=adr	adr = relative virtuelle Adresse eines 4 Zeichen langen alphanumerischen Wertes an.
=(r)	r = Register, in dem ein 4 Zeichen langer alphanumerischer Wert steht. Mit dieser Angabe wird Register R0 zerstört.
MF	Zur allgemeinen Beschreibung des Operanden MF, der dazugehörigen Operandenwerte und der evtl. nachfolgenden Operanden siehe Abschnitt „Allgemeine Makro-Operanden“ auf Seite 88 . Die gültigen MF-Werte sind zu Beginn der Makrobeschreibung bei Makrotyp angegeben und aus dem Aufrufformat ersichtlich.

Funktionsweise

Wenn der Makroaufruf ausgeführt wird, tritt Folgendes auf:

- Alle dem Programm zugewiesenen und noch geöffneten Dateien werden geschlossen.
- Der Speicherplatz für das Programm wird freigegeben.
- Die BBS-Operationsliste und die Einträge der BBS-Programmtabelle werden freigegeben.
- Eine ggf. definierte STXIT-Routine für die Ereignisklasse TERM wird aktiviert.
- AIDSYS wird aufgerufen mit dem Ereignis „TERM“.
- In der Geräteliste werden die Byte 8 - 30 für jedes freigegebene Gerät gelöscht. In der Operationsliste wird das erste Byte auf X'FF' gesetzt. In dem Eintrag der Programmtabelle wird die Programmanfangsadresse auf 0 gesetzt (4 Byte).
- Anschließend wechselt das System in den Kommando-Modus.

Hinweise

- Ist im URETCD-Operanden eine ungültige Adresse angegeben, so wird der Operand ignoriert.

- Register R1 enthält die Adresse des Datenbereichs.
Ist die Adresse des Datenbereichs ungültig oder wurden fehlerhafte Operanden eingegeben, so wird `TERM UNIT=STEP,MODE=ABNORMAL,DUMP=Y` ausgeführt und folgende Fehlermeldung ausgegeben:

```
%.... ABNORMAL PROGRAM TERMINATION NRT0601
```

- Der Aufruf des Makros `TERM` mit dem Operanden `DUMP=Y` veranlasst die Meldung „PROCESSING INTERRUPTED AT...“

Die Ausgabe des Speicherabzugs hängt vom Wert des Operanden `DUMP` im Kommando `MODIFY-TEST-OPTIONS` ab. Bei `DUMP=*STD` (Default-Wert) veranlasst `TERM` eine der folgenden Meldungen:

- im Dialogbetrieb fragt das System:
„DUMP DESIRED ? REPLY (Y=YES, N=NO)“
Die Ausgabe des Dumps hängt von der Antwort ab.
- im Batchbetrieb und in Prozeduren wird der Dump unterdrückt und folgende Meldung ausgegeben:
„SYSTEM REGULATIONS PROHIBIT DUMP“

TIMJV

Auftragsüberwachende JV modifizieren

Allgemeines

Anwendungsgebiet: Jobvariablen

Makrotyp: S-Typ: (Standardform/C-/D-E-/L-Form)

Bei der C-Form und D-Form des Makroaufrufs kann ein Präfix PREFIX (pre = 1..3 Buchstaben) angegeben werden (siehe [Abschnitt „Der Operand MF“ auf Seite 88](#)).

Voreinstellung: PREFIX=IDJ

Makrobeschreibung

Mit dem Makro TIMJV können auf Programmebene folgende Elemente im Systemteil einer auftragsüberwachenden Jobvariablen modifiziert werden (siehe auch Kommando MODIFY-MONJV):

- Zeitstempel (Operand TIMESTAMP)
- Name des zu überwachenden Auftrages (Operand DESCRIPTOR)
- Auftragspezifische Informationen (Operand USER-INFORMATION)

Format und Position der Elemente im Systemteil sind im [Abschnitt „Werte für überwachende Jobvariablen“ auf Seite 54](#) beschrieben.

Makroaufrufformat und Operandenbeschreibung

Operation	Operanden
TIMJV	<p>[MONJV=jvid]</p> <p>,TIMESTAMP= { <u>*UNCHANGED</u> } *SET }</p> <p>,DESCRIPTOR= { <u>*UNCHANGED</u> } auftragsname }</p> <p>,INFO= { <u>*UNCHANGED</u> } auftragsinfos }</p> <p>[,PASS=kennwort]</p> <p>,MF= { S C (E,...) D L } ,PREFIX= { IDJ } ,VERSION=<u>1</u> pre }</p>

MONJV=jvid	Name der auftragsüberwachenden Jobvariablen. Ohne Angabe einer Jobvariablen wirkt der Makro auf die auftragsüberwachende Jobvariable des Auftrags, in dem er aufgerufen wurde.
TIMESTAMP	Gibt an, ob ein Zeitstempel für die Auftragsüberwachung gesetzt werden soll.
=* <u>UNCHANGED</u>	Voreinstellung: Es wird kein Zeitstempel geschrieben.
=*SET	Es wird ein Zeitstempel in UTC-Zeit geschrieben. Format: yyyy-mm-ddhhmss.
DESCRIPTOR	Gibt an, ob der Name des zu überwachenden Auftrags gesetzt werden soll.
=* <u>UNCHANGED</u>	Voreinstellung: Es wird kein Wert geschrieben.
=auftragsname	Name des zu überwachenden Auftrags (max. 8 Zeichen).

INFO	Gibt an, ob auftragsspezifische Informationen gesetzt werden sollen.
=* <u>UNCHANGED</u>	Voreinstellung: Es wird kein Wert geschrieben.
=auftragsinfo	Auftragsspezifische Informationen (max. 58 Byte Zeichen).
PASS=kennwort	Lese- oder Schreibkennwort der Jobvariablen.
MF PREFIX	Zur Beschreibung der Operanden MF und PREFIX siehe Seite 88 . Die gültigen Werte sind zu Beginn der Makrobeschreibung dargestellt und aus dem Aufrufformat ersichtlich.
VERSION=1	Voreinstellung: Die Makroauflösung entspricht dem Stand $JV \geq V13.0B$.

Hinweis

Die Operanden **TIMESTAMP**, **DESCRIPTOR** und **INFO** dürfen nur bei der Makrogenerierung mit **MF=S** bzw. **MF=L** angegeben werden

Rückinformation und Fehleranzeigen

siehe [Seite 229](#).

DSECT

```

TIMJV    TIMJV MF=D
1 *****
1 *      VERSION 312
1 *****
1 *      T I M J V   P A R A M E T E R   L I S T                *
1 *****
1          #INTF REFTYPE=REQUEST,                                C
1          INTNAME=TIMJV,INTCOMP=001
1 TIMJV    DSECT
1 *****
1 *      UNIT=41, FUNCTION=34,  VERSION=1                        *
1 *****
1          FHDR    MF=(C, IDJT)
2          DS      OA
2 IDJTFHE DS    OXL8          0    GENERAL PARAMETER AREA HEADER
2 *
2 IDJTIFID DS    OA          0    INTERFACE IDENTIFIER
2 IDJTFCU  DS    AL2          0    FUNCTION UNIT NUMBER

```

```

2 *                               BIT 15   HEADER FLAG BIT,
2 *                               MUST BE RESET UNTIL FURTHER NOTICE
2 *                               BIT 14-12 UNUSED, MUST BE RESET
2 *                               BIT 11-0   REAL FUNCTION UNIT NUMBER
2 IDJTFCT DS      AL1              2   FUNCTION NUMBER
2 IDJTFCTV DS    AL1              3   FUNCTION INTERFACE VERSION NUMBER
2 *
2 IDJTRET DS      0A              4   GENERAL RETURN CODE
2 *
2 * GENERAL_RETURN_CODE CLEARED (X'00000000') MEANS
2 * REQUEST SUCCESSFUL PROCESSED AND NO ADDITIONAL INFORMATION
2 *
2 IDJTSRET DS     0AL2            4   SUB RETURN CODE
2 IDJTSR2 DS     AL1             4   SUB RETURN CODE 2
2 * ALWAYS CLEARED (X'00') IF MAIN_RETURN_CODE IS X'FFFF'
2 * Standard subcode2 values as defined by convention:
2 IDJTR2OK EQU   X'00'            All correct, no additional info
2 IDJTR2NA EQU   X'01'            Successful, no action was necessary
2 IDJTR2WA EQU   X'02'            Warning, particular situation
2 IDJTSR1 DS     AL1             5   SUB RETURN CODE 1
2 *
2 * GENERAL INDICATION OF ERROR CLASSES
2 *
2 * CLASS A   X'00'              FUNCTION WAS SUCCESSFULLY PROCESSED
2 * CLASS B   X'01' - X'1F'      PARAMETER SYNTAX ERROR
2 * CLASS C   X'20'              INTERNAL ERROR IN CALLED FUNCTION
2 * CLASS D   X'40' - X'7F'      NO CLASS SPECIFIC REACTION POSSIBLE
2 * CLASS E   X'80' - X'82'      WAIT AND RETRY
2 *
2 IDJTRFSP EQU   X'00'            FUNCTION SUCCESSFULLY PROCESSED
2 IDJTRPER EQU   X'01'            PARAMETER SYNTAX ERROR
2 * 3 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'01' - X'1F'
2 IDJTRFNS EQU   X'01'            CALLED FUNCTION NOT SUPPORTED
2 IDJTRFNA EQU   X'02'            CALLED FUNCTION NOT AVAILABLE
2 IDJTRVNA EQU   X'03'            INTERFACE VERSION NOT SUPPORTED
2 *
2 IDJTRAER EQU   X'04'            ALIGNMENT ERROR
2 IDJTRIER EQU   X'20'            INTERNAL ERROR
2 IDJTRCAR EQU   X'40'            CORRECT AND RETRY
2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'40' - X'7F'
2 IDJTRECR EQU   X'41'            SUBSYSTEM (SS) MUST BE CREATED
2 *                               EXPLICITELY BY CREATE-SS
2 IDJTREC� EQU   X'42'            SS MUST BE EXPLICITELY CONNECTED
2 *
2 IDJTRWAR EQU   X'80'            WAIT FOR A SHORT TIME AND RETRY
2 IDJTRWLR EQU   X'81'            "      LONG      "
2 IDJTRWUR EQU   X'82'            WAIT TIME IS UNCALCULABLY LONG
2 *                               BUT RETRY IS POSSIBLE

```

```

2 * 2 GLOBALLY DEFINED ISL ERROR CODES IN CLASS X'80' - X'82'
2 IDJTRTNA EQU X'81'          SS TEMPORARILY NOT AVAILABLE
2 IDJTRDH EQU X'82'          SS IN DELETE / HOLD
2 *
2 IDJTMRET DS OAL2           6 MAIN RETURN CODE
2 IDJTMR2 DS AL1            6 MAIN RETURN CODE 2
2 IDJTMR1 DS AL1            7 MAIN RETURN CODE 1
2 *
2 * SPECIAL LAYOUT OF LINKAGE_MAIN_RETURN_CODE (YYYY IN X'00XXYYYY')
2 *
2 IDJTRLNK EQU X'FFFF'       LINKAGE ERROR / REQ. NOT PROCESSED
2 IDJTJFHL EQU 8             8 GENERAL OPERAND LIST HEADER LENGTH
2 *
1 *****
1 * END OF STANDARD HEADER. START OF SPECIAL TIMJV PARAMETER LIST *
1 *****
1 IDJTHDRI EQU X'00292201',4
1 IDJTJV DS CL54             MONJV
1 IDJTFLG DS XL1             FLAG
1 IDJTTIME EQU X'80'         7-7 1=SET TIMESTAMP 0=UNCH
1 IDJTDEUN EQU X'40'         6-6 1=DESCRIPTOR UNCHANGED
1 *                          0=DESCRIPTOR GIVEN
1 IDJTINUN EQU X'20'         5-5 1=INFO UNCHANGED
1 *                          0=INFO GIVEN
1 IDJTRES1 DS XL1            RESERVED
1 IDJTPASS DS CL4            PASSWORD
1 DS XL12
1 IDJTDESC DS CL8            DESCRIPTOR
1 IDJTINFO DS CL58           INFO
1 IDJTRES2 DS XL2            RESERVED
1 IDJTJVS DS A               RESERVED
1 IDJTPLLN EQU *-TIMJV       LENGTH OF DSECT
1 *****
1 SPACE

```

5 Beispiele

Beispiel 1 (Verwaltung von Jobvariablen)

Gezeigt wird die Anwendung der Kommandos zur Verwaltung von Jobvariablen:

```
/show-jv-attr----- (1)
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR1
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS1
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS2
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS3
%SUM      00004 JV'S; JV-VALUE = 00000000 BYTES
/create-jv jv=jv.perm.error2----- (2)
/show-jv-attributes jv.perm.error*----- (3)
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR1
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR2
%SUM      00002 JV'S; JV-VALUE = 00000000 BYTES
/modify-jv jv=jv.perm.error2,set-val=c'No Error'----- (4)
/show-jv-attr jv=jv.perm.error2,inf=*all-attr----- (5)
%0000008 :4V05:$COGNITAS.JV.PERM.ERROR2
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:14:52 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%SUM      00001 JV'S; JV-VALUE = 00000008 BYTES
/show-jv jv=jv.perm.error2----- (6)
%No Error
```

- (1) Es werden alle permanenten Jobvariablen ausgegeben.
- (2) Die Jobvariable JV.PERM.ERROR2 wird mit Standardwerten neu erzeugt.
- (3) Es werden alle Jobvariablen ausgegeben, die mit der Zeichenfolge „JV.PERM.ERROR“ beginnen.
- (4) Die Jobvariable JV.PERM.ERROR2 erhält den Inhalt 'No Error'.
- (5) Alle Merkmale der Jobvariablen JV.PERM.ERROR2 werden ausgegeben.
- (6) Der Inhalt der Jobvariablen JV.PERM.ERROR2 wird ausgegeben.

```

/mod-jv-attr jv=jv.perm.error2,prot=(write-pass=c'c5aq') ----- (7)
/modify-jv jv=jv.perm.error2,set-val=c'write error' ----- (8)
% JVS04B1 PASSWORD NOT SPECIFIED. COMMAND REJECTED
/add-pass password=c'c5aq' ----- (9)
/modify-jv jv=jv.perm.error2,set-val=c'write error'
/show-jv jv=jv.perm.error2 ----- (10)
%write error
/show-jv-attr jv=jv.perm.error2,inf=*all-attr ----- (11)
%0000011 :4V05:$COGNITAS.JV.PERM.ERROR2
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:17:41 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = YES
SUM 00001 JV'S; JV-VALUE = 00000011 BYTES
/create-jv jv=#jv.temp.t1 ----- (12)
/show-jv-attr
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR1
%0000011 :4V05:$COGNITAS.JV.PERM.ERROR2
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS1
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS2
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS3
%SUM 00005 JV'S; JV-VALUE = 00000011 BYTES
/show-jv-attr jv=# ----- (13)
%0000000 :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T1
%SUM 00001 JV'S; JV-VALUE = 00000000 BYTES
/show-jv-attr select=*by-attr(password=*write-pass) ----- (14)
%0000011 :4V05:$COGNITAS.JV.PERM.ERROR2
%SUM 00001 JV'S; JV-VALUE = 00000011 BYTES

```

- (7) Die Jobvariable JV.PERM.ERROR2 wird mit dem Kennwort 'C5AQ' gegen unberechtigtes Schreiben geschützt.
- (8) Der schreibgeschützten Jobvariablen JV.PERM.ERROR2 kann kein neuer Inhalt zugewiesen werden.
- (9) Um einen neuen Inhalt zu setzen, wird das Kennwort in die Kennwort-Tabelle des Auftrags eingetragen werden.
- (10) Ausgabe des neuen Inhalts 'write error'.
- (11) Ausgabe aller Merkmale der Jobvariablen.
- (12) Die temporäre Jobvariable JV.TEMP.1 wird erzeugt.
- (13) Mit JV=# werden alle temporären Jobvariablen ausgegeben.
- (14) Es werden nur permanente Jobvariablen ausgegeben, die mit Schreibkennwort geschützt sind.


```

/CREATE-JV JV=JV.PERM.ERROR3,PROT=(BASIC-ACL=*STD)----- (15)
/SHOW-JV-ATTR JV=JV.PERM.ERROR3,INF=*ALL-ATTR
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR3
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% OWNER = R W GROUP = - - OTHERS = - -
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:22:23 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%SUM 00001 JV'S; JV-VALUE = 00000000 BYTES
/MOD-JV-ATTR JV=JV.PERM.ERROR3,NEW-NAME=#JV.TEMP.T3----- (16)
% JVS0449 ONLY DEFAULT ATTRIBUTES PERMITTED FOR TEMPORARY JOB VARIABLE.
COMMAND REJECTED
/MOD-JV-ATTR JV=JV.PERM.ERROR3,NEW-NAME=#JV.TEMP.T3,PROT=(BASIC-ACL=*NONE)
/SHOW-JV-ATTR JV=#,INF=*ALL-ATTR
%0000000 :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T1
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:20:56 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%0000000 :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T3
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:22:23 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%SUM 00002 JV'S; JV-VALUE = 00000000 BYTES
/SHOW-JV-ATTR JV=#J*
%0000000 :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T1
%0000000 :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T3
%SUM 00002 JV'S; JV-VALUE = 00000000 BYTES

```

- (15) Die permanente Jobvariable JV.PERM.ERROR3 wird erzeugt und mit einer BASIC-ACL geschützt.
- (16) Ein Umbenennen der permanenten Jobvariablen JV.PERM.ERROR3 in die temporäre Jobvariable JV.TEMP.T3 ist erst möglich, wenn die Schutzmerkmale explizit auf den Standardwert zurückgesetzt werden.

```

/delete-jv jv=#jv.temp. _____ (17)
% JVS0465 DELETE ALL JOB VARIABLES ':4V05:$COGNITAS.S.187.OFDB.JV.TEMP.' OF
USER ID? REPLY (Y=YES; N=NO; T=TERMINATE COMMAND; ?=EXPLAIN ADDITIONAL
OPTIONS)?y
/delete-jv jv=jv.perm.status* _____ (18)
% JVS0465 DELETE ALL JOB VARIABLES ':4V05:$COGNITAS.JV.PERM.STATUS*' OF USER
ID? REPLY (Y=YES; N=NO; T=TERMINATE COMMAND; ?=EXPLAIN ADDITIONAL
OPTIONS)?y,check=single
% JVS0469 DELETE JOB VARIABLE ':4V05:$COGNITAS.JV.PERM.STATUS1'? REPLY
(Y=YES; N=NO; T=TERMINATE; ,CHECK=NEW MODE)?y
% JVS0469 DELETE JOB VARIABLE ':4V05:$COGNITAS.JV.PERM.STATUS2'? REPLY
(Y=YES; N=NO; T=TERMINATE; ,CHECK=NEW MODE)?y
% JVS0469 DELETE JOB VARIABLE ':4V05:$COGNITAS.JV.PERM.STATUS3'? REPLY
(Y=YES; N=NO; T=TERMINATE; ,CHECK=NEW MODE)?y
/show-jv-attr jv.perm.,inf=*all-attr
%000000 :4V05:$COGNITAS.JV.PERM.ERROR1
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:11:13 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%000011 :4V05:$COGNITAS.JV.PERM.ERROR2
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:17:41 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = YES
%SUM 00002 JV'S; JV-VALUE = 00000011 BYTES
/mod-jv-attr jv=jv.perm.error1,prot=(basic-acl=*previous) _____ (19)
/show-jv-attr jv.perm.error1,inf=*all-attr
%000000 :4V05:$COGNITAS.JV.PERM.ERROR1
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% OWNER = R W GROUP = - - OTHERS = - -
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:11:13 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%SUM 00001 JV'S; JV-VALUE = 00000000 BYTES

```

- (17) Alle temporären Jobvariablen, deren Name mit „JV.TEMP.“ beginnt, sollen gelöscht werden.
- (18) Alle permanenten Jobvariablen, die mit JV.PERM.STATUS beginnen, sollen gelöscht werden. Während des Löschens wird der Kontrollmodus für die Jobvariablen JV.PERM.STATUS* geändert: Kontrollabfrage für jede zu löschende Jobvariable.
- (19) Die Jobvariable JV.PERM.ERROR1 wird mit einer BASIC-ACL geschützt, deren Werte entsprechend der Standard-Zugriffskontrolle gesetzt werden.

```

/ create-jv jv=jv.perm.status5,prot=(basic-acl=*std) ----- (20)
/ show-jv-attr jv=**status5,inf=*all-attr
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS5
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% OWNER = R W GROUP = - - OTHERS = - -
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:32:25 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%SUM 00001 JV'S; JV-VALUE = 00000000 BYTES
/ create-jv jv=jv.perm.status6 ----- (21)
/ mod-jv-attr jv=jv.perm.status6,prot=(retention-period=10)
/ show-jv-attr jv=**status6,inf=*all-attr
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS6
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-14
% CRE-TIME = 14:34:54 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%SUM 00001 JV'S; JV-VALUE = 00000000 BYTES
/ show-jv-attr select=*by-attr(basic-acl=*yes) ----- (22)
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR1
%0000000 :4V05:$COGNITAS.JV.PERM.STATUS5
SUM 00002 JV'S; JV-VALUE = 00000000 BYTES
/ set-jv-link link=status6,jv=jv.perm.status6
/ set-jv-link link=stat,jv=jv.perm.status6 ----- (23)
/ show-jv-link
% LINK-NAME JV-NAME
% *STAT :4V05:$COGNITAS.JV.PERM.STATUS6
% *STATUS6 :4V05:$COGNITAS.JV.PERM.STATUS6
/ set-jv-link link=stat,jv=jv.perm.status5 ----- (24)
/ show-jv-link
% LINK-NAME JV-NAME
% *STAT :4V05:$COGNITAS.JV.PERM.STATUS5
% *STATUS6 :4V05:$COGNITAS.JV.PERM.STATUS6

```

- (20) Die Jobvariable JV.PERM.STATUS5 wird neu erzeugt und mit einer BASIC-ACL geschützt, in der nur der Eigentümer alle Zugriffsrechte besitzt.
- (21) Die Jobvariable JV.PERM.STATUS6 wird neu erzeugt und anschließend für zehn Tage gegen Änderung geschützt.
- (22) Es werden nur Jobvariablen ausgegeben, die mit einer BASIC-ACL geschützt sind.
- (23) Für die Jobvariable JV.PERM.STATUS6 werden in der JV-LINK-Tabelle des Auftrags zwei Einträge unter dem Kettungsnamen STATUS6 und STAT angelegt.
- (24) Für die Jobvariable JV.PERM.STATUS5 wird ein Kettungsname STAT angelegt und der bereits existierende damit überschrieben.

```

/set-jv-link link=temp1,jv=#jv.temp.t1 _____ (25)
/show-jv-link
% LINK-NAME   JV-NAME
% *STAT       :4V05:$COGNITAS.JV.PERM.STATUS5
% *STATUS6    :4V05:$COGNITAS.JV.PERM.STATUS6
% *TEMP1      :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T1
/show-jv-attr jv=*link(link=temp1),inf=*all-attr _____ (26)
%0000000 :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T1
% USER-ACC   = OWNER-ONLY   ACCESS      = WRITE
% CRE-DATE   = 2010-08-04   EXPIR-DATE = 2010-08-04
% CRE-TIME   = 14:38:41    EXPIR-TIME = 00:00:00
% READ-PASS  = NONE
% WRITE-PASS = NONE
%SUM        00001 JV'S; JV-VALUE = 00000000 BYTES
/del-jv jv=*link(link=status6) _____ (27)
% JVS04A3 ERROR WHEN DELETING JOB VARIABLE ':4V05:$COGNITAS.JV.PERM.STATUS6'
% JVS04B6 EXPIRATION DATE FOR JOB VARIABLE NOT YET REACHED. COMMAND REJECTED
/del-jv jv=*link(link=status6),ignore-prot=*expir _____ (28)
/del-jv jv=jv.perm.status5,dialog-control=*jv-change _____ (29)
% JVS0469 DELETE JOB VARIABLE ':4V05:$COGNITAS.JV.PERM.STATUS5'? REPLY
(Y=YES; N=NO; T=TERMINATE; ,CHECK=NEW MODE)?y
/show-jv-attr
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR1
%0000011 :4V05:$COGNITAS.JV.PERM.ERROR2
%SUM      00002 JV'S; JV-VALUE = 00000011 BYTES
/show-jv-link _____ (30)
% LINK-NAME   JV-NAME
% *STAT       :4V05:$COGNITAS.JV.PERM.STATUS5
% *STATUS6    :4V05:$COGNITAS.JV.PERM.STATUS6
% *TEMP1      :4V05:$COGNITAS.S.187.0FDB.JV.TEMP.T1

```

- (25) Für die temporäre Jobvariable JV.TEMP.T1 wird ein Eintrag unter dem Kettungsna-
men TEMP1 angelegt. Da die Jobvariable noch nicht existiert, wird sie vom System
neu erzeugt.
- (26) Die Jobvariablen können in Kommandos über den Kettungsnamen angesprochen
werden.
- (27) Die Jobvariable JV.PERM.STATUS6 soll gelöscht werden; sie wird über ihren Link-
namen angesprochen. Die Jobvariable kann nicht gelöscht werden, da ihr Freiga-
bedatum noch nicht erreicht ist
- (28) Die Jobvariable JV.PERM.STATUS6 kann gelöscht werden, wenn das Freigabeda-
tum nicht beachtet wird.
- (29) Durch Angabe des Operanden dialog-control=*jv-change wird für die zu lö-
schende Jobvariable eine Kontrollabfrage durchgeführt. Die Jobvariable
JV.PERM.STATUS5 wird gelöscht.

- (30) Obwohl die Jobvariablen JV.PERM.STATUS5 und JV.PERM.STATUS6 gelöscht wurden, sind alle JV-LINK-Einträge noch vorhanden!

```

/mod-jv-attr jv=jv.perm.error1,prot=(basic-acl=(owner=(read=y,write=n),
      group=*no-access,others=*no-access)) _____ (31)
/show-jv-attr jv.perm.error1,inf=*all-attr
%0000000 :4V05:$COGNITAS.JV.PERM.ERROR1
% USER-ACC = OWNER-ONLY ACCESS = WRITE
% OWNER = R - GROUP = - - OTHERS = - -
% CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
% CRE-TIME = 14:11:13 EXPIR-TIME = 00:00:00
% READ-PASS = NONE
% WRITE-PASS = NONE
%SUM 00001 JV'S; JV-VALUE = 00000000 BYTES
/del-jv jv=jv.perm.error1,dialog-control=*jv-change
% JVS0469 DELETE JOB VARIABLE ':4V05:$COGNITAS.JV.PERM.ERROR1'? REPLY
(Y=YES; N=NO; T=TERMINATE; ,CHECK=NEW MODE)?y
% JVS04A3 ERROR WHEN DELETING JOB VARIABLE ':4V05:$COGNITAS.JV.PERM.ERROR1'
% JVS04BF REQUESTED ACCESS TO JV NOT PERMITTED DUE TO EXISTING JV
PROTECTION. COMMAND REJECTED
/del-jv jv=jv.perm.error1,ignore-protection=*access _____ (32)

```

- (31) Die Jobvariable wird mit einer BASIC-ACL gegen versehentliches Überschreiben geschützt: der Eigentümer darf nur lesend zugreifen und alle anderen Benutzer haben keinen Zugriff.
- (32) Beim Löschen der Jobvariable JV.PERM.ERROR1 müssen die Schutzattribute (hier BASIC-ACL) beachtet werden.

Beispiel 2 (Programmüberwachung)

Das Beispiel soll die Benutzung überwachender Jobvariablen für die Kommunikation zwischen Programmen zeigen. Es sind zwei Programme, jedes innerhalb eines Prozesses, vorhanden.

Programm-1 soll auf jeden Fall laufen, wogegen Programm-2 je nach Ausgang von Programm-1 laufen soll oder nicht. Deshalb muss Programm-2 die Beendigung von Programm-1 abwarten und prüfen, ob es normal beendet wurde.

Die überwachende Jobvariable wird von JOB2 nach erfolgter Abfrage gelöscht. Beide Aufträge werden unter derselben Benutzerkennung gestartet. Der Auftrag kann nach Programm-1 noch weitere Verarbeitungsschritte vorsehen, unabhängig von der Durchführung von Auftrag JOB2.

```

/      SET-LOGON-PARAMETERS JOB-NAME=JOB1
/      START-PROGRAM FROM-FILE=PROGRAM-1,MONJV=JV.PROG1
/      SET-JOB-STEP
/      SKIP-COMMANDS TO-LABEL=FEHL,IF=*JV(CONDITION=((JV.PROG1,1,2)=C'$A'))
/      START-EXE FROM-FILE=PROGRAM-XY 1
      .
      .
      .
      .
/.FEHL SHOW-JV JV=JV.PROG1
/      EXIT-JOB

/      SET-LOGON-PARAMETERS JOB-NAME=JOB2
/      SHOW-JV JV=JV.PROG1
/      WAIT-EVENT UNTIL=*JV(CONDITION=((JV.PROG1,1,2)=C'$T' OR -
/                                     (JV.PROG1,1,2)=C'$A')), -
/                                     TIME-LIMIT=3600,TIMEOUT-LABEL=FEHL)
/      SKIP-COMMANDS TO-LABEL=FEHL,IF=JV(CONDITION=((JV.PROG1,1,2)=C'$A'))
/      START-EXE FROM-FILE=PROGRAM-2
/.FEHL SHOW-JV JV=JV.PROG1
/      DELETE-JV JV=JV.PROG1
/      EXIT-JOB
    
```

¹ Das Kommando START-EXECUTABLE-PROGRAM ersetzt ab BLSSERV 2.3 das Kommando START-PROGRAM

Beispiel 3 (Programmüberwachung)

Das Beispiel zeigt die Benutzung des TERM-Makros, mit dem ein vom Benutzer definierter Rückkehrcode in die das Programm überwachende Jobvariable eingetragen werden kann. Ein Programm PROG1 setzt im Fehlerfall die überwachende Jobvariable auf einen Wert größer 1. Der Benutzer kann nun diesen Wert abfragen, um zu entscheiden, ob es sinnvoll ist, ein weiteres Programm (PROG2) ausführen zu lassen oder nicht. Man beachte, dass vom System die Zustandsanzeige (3 Stellen) gesetzt wird, sodass der vom Benutzer geänderte Wert bei Position 4 anfängt. Seine Länge ist 4 Zeichen. Soll PROG2 von einem anderen Auftrag gestartet werden, so muss dort vorher auf das Programmende von PROG1 gewartet werden:

```
/WAIT-EVENT UNTIL=*JV(CONDITION=((ERROR,1,2)=C'$T') OR ((ERROR,1,2)=C'$A'),
  TIME-LIMIT=...,TIMEOUT-LABEL=...)
```

Programm PROG1

```
PROG1  START
      :
FEHL3  LR R9,R15
      :
      TERM URETCD=(R9) ----- (1)
      END
      :

/      SET-LOGON-PARAMETERS JOB-NAME=BEISP3
/      SET-JV-LINK JV-NAME=ERROR
/      START-ASSEMBH
//     COMPILE SOURCE=PROG1,...
//     END
/      START-EXE FROM-FILE=*OMF,MONJV=ERROR 1
/      SET-JOB-STEP
/      SKIP-COMMANDS TO-LABEL=EXIT,IF=*JV(CONDITION=((ERROR,4,4)>C'0001')
/      START-EXE FROM-FILE=PROG2
/.EXIT EXIT-JOB
```

- (1) Der Operand URETCD im Term-Makro sorgt dafür, dass in der programmüberwachenden Jobvariablen hinter der Zustandsanzeige in Byte 4-7 des Wert-Feldes der Inhalt des genannten Registers aufbereitet in 4 Dezimalstellen abgelegt wird (siehe auch Handbuch „Makroaufrufe an den Ablaufteil“ [4]).

¹ Das Kommando START-EXECUTABLE-PROGRAM ersetzt ab BLSSERV 2.3 das Kommando START-PROGRAM

Beispiel 4 (Auftragsüberwachung)

Die Aufträge AJOB, BJOB und CJOB werden von einem Steuerauftrag aus gestartet und überwacht.

```

/          SET-LOGON-PARAMETERS JOB-NAME=STEUER
/          ENTER-JOB FROM-FILE=AJOB,MONJV=JV.AJOB
/          ENTER-JOB FROM-FILE=BJOB,MONJV=JV.BJOB
/          ENTER-JOB FROM-FILE=CJOB,MONJV=JV.CJOB _____ (1)
/          SHOW-JOB-STATUS JOB-ID=*MONJV(JV.AJOB) _____ (2)
/          WAIT-EVENT UNTIL=*JV(CONDITION=((JV.AJOB,1,2)=C'$T' OR -
/                                     (JV.AJOB,1,2)=C'$A'),-
/                                     TIME-LIMIT=3600,TIMEOUT-LABEL=FEHL1) ____ (3)
/          SKIP-COMMANDS TO-LABEL=NORMAL,IF=*JV(CONDITION=((JV.AJOB,1, -
/                                     2)=C'$T')) - (4)
/.FEHL1   CANCEL-JOB JOB-ID=*MONJV(JV.CJOB) _____ (5)
/.NORMAL  SET-JOB-STEP
/          SKIP-COMMANDS TO-LABEL=ENDE,IF=*JV(CONDITION=((JV.BJOB,1,2) -
/                                     NE C'$R'))
/          CHANGE-TASK-PRIORITY JOB-ID=*MONJV(JV.BJOB),RUN-PRIORITY=130 (6)
/.ENDE    EXIT-JOB

```

- (1) Die Aufträge werden gestartet und überwacht.
- (2) Informationen über AJOB werden abgefragt.
- (3) Warten bis AJOB beendet (normal oder abnormal), aber maximal 3600 Sekunden.
- (4) Abfrage, ob AJOB normal oder abnormal beendet wurde.
- (5) Im Falle der abnormalen Beendigung von AJOB wird die Verarbeitung von CJOB ebenfalls abgebrochen.
- (6) Falls CJOB noch läuft, wird die Priorität verändert.

Beispiel 5 (Auftragsüberwachung)

Folgende Abhängigkeit besteht zwischen den Aufträgen AJOB, BJOB, CJOB, EJOB und FJOB:

Starte AJOB erst, wenn BJOB normal beendet ist.

Wenn CJOB normal beendet ist, dann starte gleichzeitig EJOB und FJOB.

Die gewünschte Steuerung wird durch folgenden Koordinations-Job realisiert:

```

/SET-LOGON-PARAMETERS JOB-NAME=MASTER
/REMARK *****
/REMARK **          BJOB UND CJOB STARTEN          **
/REMARK *****
/ENTER-JOB FROM-FILE=BJOB,MONJV=JV.BJOB,JOB-NAME=BJOB
/ENTER-JOB FROM-FILE=CJOB,MONJV=JV.CJOB,JOB-NAME=CJOB
/REMARK *****
/REMARK ** WENN BJOB FERTIG, AJOB STARTEN          **
/REMARK *****
/ADD-CJC-ACTION CONDITION=((JV.BJOB,1,2)=C'$T'),-
/          NAME=BJOB,TIME-LIMIT=3600
/ENTER-JOB FROM-FILE=AJOB,JOB-NAME=AJOB
/END-CJC-ACTION
/REMARK *****
/REMARK ** WENN CJOB FERTIG, EJOB UND FJOB STARTEN*
/REMARK *****
/ADD-CJC-ACTION CONDITION=((JV.CJOB,1,2)=C'$T'),-
/          NAME=CJOB,TIME-LIMIT=3600
/ENTER-JOB FROM-FILE=EJOB,JOB-NAME=EJOB
/ENTER-JOB FROM-FILE=FJOB,JOB-NAME=FJOB
/END-CJC-ACTION
/REMARK *****
/REMARK ** WARTEN BIS ALLE EREIGNISSE EINGETETEN **
/REMARK ** SIND UND ALLE AKTIONEN GESTARTET SIND **
/REMARK *****
/WAIT-EVENT UNTIL=*JV(CONDITION=((JV.BJOB,1,2)=C'$T' OR -
/          (JV.BJOB,1,2)=C'$A') AND -
/          ((JV.CJOB,1,2)=C'$T' OR -
/          (JV.CJOB,1,2)=C'$A')),-
/          TIME-LIMIT=3600)
/EXIT-JOB

```

Beispiel 6 (Auftragsüberwachung)

Der Auftrag SPV kontrolliert den Ablauf von JOBA, JOBB und JOBC. JOBA beeinflusst durch Setzen der Jobvariablen MONA den Start von JOBB, sowie das „gewaltsame“ Ende von JOBC (CANCEL-JOB). MONB und MONC sind überwachende Jobvariablen für JOBB und JOBC.

Auftrag SPV

```
/ SET-LOGON-PARAMETERS JOB-NAME=SPV
/ ASSIGN-SYSOUT TO-FILE=OUT.E.SPV
/ REMARK *** JOBVARIABLEN DEKLARIEREN ***
/ SET-JV-LINK JV-NAME=MONA
/ SET-JV-LINK JV-NAME=MONB
/ SET-JV-LINK JV-NAME=MONC
/ REMARK *** JOBA STARTEN ***
/ ENTER-JOB FROM-FILE=JOBA, JOB-CLASS=JCBATCH
/ SHOW-USER-STATUS
/ REMARK *** JOBB STARTEN WENN MONA GESETZT ***
/ WAIT-EVENT UNTIL=*JV(CONDITION=(MONA=C'B STARTEN'), TIME-LIMIT=100)
/ ENTER-JOB FROM-FILE=JOBB, MONJV=MONB, JOB-CLASS=JCBATCH
/ SHOW-USER-STATUS
/ SHOW-JOB-STATUS JOB-ID=*MONJV(MONB)
/ REMARK ***JOBC STARTEN WENN JOBB BEENDET ***
/ ADD-CJC-ACTION CONDITION=((MONB,1,2)=C'$T'), TIME-LIMIT=100
/ ENTER-JOB FROM-FILE=JOBC, MONJV=MONC, JOB-CLASS=JCBATCH
/ END-CJC-ACTION
/ WAIT-EVENT UNTIL=*JV(CONDITION=((MONC,1,2)=C'$R'))
/ SHOW-JOB-STATUS JOB-ID=*MONJV(MONC)
/ REMARK *** JOBC CANCELN WENN MONA GESETZT IST ***
/ WAIT-EVENT UNTIL=*JV(CONDITION=(MONA=C'C CANCELN'), TIMEOUT=W1)
/ CANCEL-JOB JOB-ID=*MONJV(MONC)
/.W1 WAIT-EVENT UNTIL=*JV(TIME-LIMIT=10, TIMEOUT-LABEL=W2)
/.W2 SHOW-USER-STATUS
/ WAIT-EVENT UNTIL=*JV(TIME-LIMIT=20, TIMEOUT-LABEL=W3)
/.W3 EXIT-JOB
```

Auftrag JOBA

```

/SET-LOGON-PARAMETERS JOB-NAME=JOBA
/ASSIGN-SYSOUT TO=OUT.E.JOBA
/WRITE-TEXT TEXT=C'*****'
/WRITE-TEXT TEXT=C'***      H I E R   I S T   J O B A      ****'
/WRITE-TEXT TEXT=C'*****'
/MODIFY-JV JV=MONA,SET-VALUE=C'B STARTEN'
/WAIT-EVENT UNTIL=*JV(CONDITION=((MONB,1,2)=C'$R'))
/WRITE-TEXT TEXT=C'*****'
/WRITE-TEXT TEXT=C'***      J O B B   G E S T A R T E T      ****'
/WRITE-TEXT TEXT=C'*****'
/WAIT-EVENT UNTIL=*JV(TIME-LIMIT=60,TIMEOUT-LABEL=W1)
/.W1 MODIFY-JV JV=MONA,SET-VALUE=C'C CANCELN'
/WAIT-EVENT UNTIL=*JV(CONDITION=((MONC,1,2)=C'$A'))
/WRITE-TEXT TEXT=C'*****'
/WRITE-TEXT TEXT=C'***      J O B C   G E C A N C E L T      ****'
/WRITE-TEXT TEXT=C'*****'
/EXIT-JOB

```

Auftrag JOBB

```

/SET-LOGON-PARAMETERS JOB-NAME=JOBB
/ASSIGN-SYSOUT TO=OUT.E.JOBB
/WRITE-TEXT TEXT=C'*****'
/WRITE-TEXT TEXT=C'***      H I E R   I S T   J O B B      ****'
/WRITE-TEXT TEXT=C'*****'
/WAIT-EVENT UNTIL=*JV(TIME-LIMIT=20,TIMEOUT-LABEL=W1)
/.W1 EXIT-JOB

```

Auftrag JOBC

```

/SET-LOGON-PARAMETERS JOB-NAME=JOBC
/ASSIGN-SYSOUT TO=OUT.E.JOBC
/WRITE-TEXT TEXT=C'*****'
/WRITE-TEXT TEXT=C'***      H I E R   I S T   J O B C      ****'
/WRITE-TEXT TEXT=C'*****'
/SET-JV-LINK JV=#LOOP
/.S1 MODIFY-JV JV=#LOOP,SET-VALUE=C'RUN'
/   WAIT-EVENT UNTIL=*JV(CONDITION=(#LOOP=C'END'),-
/                                     TIME-LIMIT=20,TIMEOUT-LABEL=S1)
/EXIT-JOB

```

Ablaufprotokoll des kontrollierenden Auftrags SPV (OUT.E.SPV)

```

/   REMARK *** JOBVARIABLEN DEKLARIEREN ***
/   SET-JV-LINK JV-NAME=MONA
/   SET-JV-LINK JV-NAME=MONB
/   SET-JV-LINK JV-NAME=MONC
/   REMARK *** JOBA STARTEN ***
/   ENTER-JOB FROM-FILE=JOBA, JOB-CLASS=JCBATCH
%   JMS0066 JOB 'JOBA' ACCEPTED ON 10-08-04 AT 15:54, TSN = OFDN
/   SHOW-USER-STATUS
NAME      TSN TYPE      PRI      CPU-USED CPU-MAX ACCOUNT#
JOBA      OFDN 1 WT      9 255      0.0        200 89001
COGDIA    OFDB 3 DIALOG  0 240      0.7721     32767 89001
SPV       OFDM 2 BATCH  9 255      0.0216     200 89001
%   SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
%   SRO0376 NO RSO JOB OF TYPE 'T7' PRESENT
%   SPS0420 RSO WARNING : SOME RSO PRINT-JOBS CANNOT BE DISPLAYED
/   REMARK *** JOBB STARTEN WENN MONA GESETZT ***
/   WAIT-EVENT UNTIL=*JV(CONDITION=(MONA=C'B STARTEN'), TIME-LIMIT=100)
%   CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:54:40
%   CJC0021 WAIT COMMAND: CONDITION = TRUE AT 15:54:41
/   ENTER-JOB FROM-FILE=JOBB, MONJV=MONB, JOB-CLASS=JCBATCH
%   JMS0066 JOB 'JOBB' ACCEPTED ON 10-08-04 AT 15:54, TSN = OFDP
/   SHOW-USER-STATUS
NAME      TSN TYPE      PRI      CPU-USED CPU-MAX ACCOUNT#
JOBB      OFDP 1 WT      9 255      0.0        200 89001
          OFDB 3 DIALOG  0 240      0.7721     32767 89001
SPV       OFDM 2 BATCH  9 255      0.0311     200 89001
JOBA      OFDN 2 BATCH  9 255      0.0150     200 89001
%   SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
%   SRO0376 NO RSO JOB OF TYPE 'T7' PRESENT
%   SPS0420 RSO WARNING : SOME RSO PRINT-JOBS CANNOT BE DISPLAYED
/   SHOW-JOB-STATUS JOB-ID=*MONJV(MONB)
TSN:      OFDP      TYPE:      1 WT      NOW:      2010-08-04.155441
JOBNAME:  JOBB      PRI:      9 255      SPOOLIN:  2010-08-04.1554
USERID:   COGNITAS JCLASS:  JCBATCH  INTYPE:    0
ACCNB:    89001     CPU-MAX:  200      START:    SOON
REPEAT:   NO        RERUN:    NO        FLUSH:    NO
ORIGFILE: :4V05:$COGNITAS.JOBB
MONJV:    :4V05:$COGNITAS.MONB
/   REMARK *** JOBC STARTEN WENN JOBB BEENDET ***
/   ADD-CJC-ACTION CONDITION=((MONB,1,2)=C'$T'), TIME-LIMIT=100
/   ENTER-JOB FROM-FILE=JOBC, MONJV=MONC, JOB-CLASS=JCBATCH
/   END-CJC-ACTION
%   CJC0050 CJC ACTION WITH ID = 2, LABEL = *NONE: ACCEPTED FOR FURTHER
EVENTS AT 15:54:41, COUNT = 1
/   WAIT-EVENT UNTIL=*JV(CONDITION=((MONC,1,2)=C'$R'))
%   CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:54:41

```

```

% CJC0051 CJC ACTION WITH ID = 2, LABEL = *NONE: CONDITION TRUE AT
15:55:04, COUNT = 0
% CJC0064 CJC ACTION WITH ID = 2, LABEL = *NONE: START OF ON OR TIMEOUT
SEQUENCE
% JMS0066 JOB 'JOBC' ACCEPTED ON 10-08-04 AT 15:55, TSN = OFDR
% CJC0065 CJC ACTION WITH ID = 2, LABEL = *NONE: END OF ON OR TIMEOUT
SEQUENCE
% CJC0052 CJC ACTION WITH ID = 2, LABEL = *NONE: TERMINATION NORMAL
% CJC0021 WAIT COMMAND: CONDITION = TRUE AT 15:55:04
/ SHOW-JOB-STATUS JOB-ID=*MONJV(MONC)
TSN: OFDR TYPE: 2 BATCH NOW: 2010-08-04.155504
JOBNAME: JOBC PRI: 9 255 SPOOLIN: 2010-08-04.1555
USERID: COGNITAS JCLASS: JCBATCH LOGON: 2010-08-04.1555
ACCNB: 89001 CPU-MAX: 200 CPU-USED:000000.0050
REPEAT: NO RERUN: NO FLUSH: NO
MRSCAT: HOLD: NO START: SOON
TID: 00010067 UNP/Q#: 00/000
CMD:
ORIGFILE::4V05:$COGNITAS.JOBC
MONJV: :4V05:$COGNITAS.MONC
/ REMARK *** JOBC CANCELN WENN MONA GESETZT IST ***
/ WAIT-EVENT UNTIL=*JV(CONDITION=(MONA=C'C CANCELN'),TIMEOUT-LABEL=W1)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:55:04
% CJC0021 WAIT COMMAND: CONDITION = TRUE AT 15:55:41
/ CANCEL-JOB JOB-ID=*MONJV(MONC)
% CAN000K CANCEL PROCESSING STARTED FOR TSN 'OFDR' WITH USER ID 'COGNITAS'
/.W1 WAIT-EVENT UNTIL=*JV(TIME-LIMIT=10,TIMEOUT-LABEL=W2)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:55:41
% CJC0022 WAIT COMMAND: TIMEOUT AT 15:55:52, SKIP TO TIMEOUT LABEL OR NEXT
STEP
/.W2 SHOW-USER-STATUS
NAME TSN TYPE PRI CPU-USED CPU-MAX ACCOUNT#
OFDB 3 DIALOG 0 240 0.7756 32767 89001
SPV OFDM 2 BATCH 9 255 0.0525 200 89001
JOB 0FDB 4 PR 255 1 0 0 0FDP
JOB 0FDM 4 PR 255 1 0 0 0FDM
JOB 0FDS 4 PR 255 1 0 0 0FDR
JOB 0FDT 4 PR 255 1 0 0 0FDN
% SPS0420 RSO WARNING : SOME RSO PRINT-JOBS CANNOT BE DISPLAYED
/ WAIT-EVENT UNTIL=*JV(TIME-LIMIT=20,TIMEOUT=W3)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:55:52
% CJC0022 WAIT COMMAND: TIMEOUT AT 15:56:12, SKIP TO TIMEOUT LABEL OR NEXT
STEP
/.W3 EXIT-JOB
% EXC0419 /LOGOFF AT 1556 ON 10-08-04 FOR TSN '0FDM'
% EXC0421 CPU TIME USED: 0.0595

```

Ablaufprotokoll des Auftrags JOBA (OUT.E.JOBA)

```

/WRITE-TEXT TEXT=C'*****'
*****
/WRITE-TEXT TEXT=C'***      H I E R   I S T   J O B A          ****'
***      H I E R   I S T   J O B A          ****
/WRITE-TEXT TEXT=C'*****'
*****
/MODIFY-JV JV=MONA,SET-VALUE=C'B STARTEN'
/WAIT-EVENT UNTIL=*JV(CONDITION=((MONB,1,2)=C'$R'))
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:54:40
% CJC0021 WAIT COMMAND: CONDITION = TRUE AT 15:54:41
/WRITE-TEXT TEXT=C'*****'
*****
/WRITE-TEXT TEXT=C'***      J O B B   G E S T A R T E T          ****'
***      J O B B   G E S T A R T E T          ****
/WRITE-TEXT TEXT=C'*****'
*****
/WAIT-EVENT UNTIL=*JV(TIME-LIMIT=60,TIMEOUT-LABEL=W1)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:54:41
% CJC0022 WAIT COMMAND: TIMEOUT AT 15:55:41, SKIP TO TIMEOUT LABEL OR NEXT
STEP
/.W1 MODIFY-JV JV=MONA,SET-VALUE=C'C CANCELN'
/WAIT-EVENT UNTIL=*JV(CONDITION=((MONC,1,2)=C'$A'))
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:55:41
% CJC0021 WAIT COMMAND: CONDITION = TRUE AT 15:55:42
/WRITE-TEXT TEXT=C'*****'
*****
/WRITE-TEXT TEXT=C'***      J O B C   G E C A N C E L T          ****'
***      J O B C   G E C A N C E L T          ****
/WRITE-TEXT TEXT=C'*****'
*****
/EXIT-JOB
% EXC0419 /LOGOFF AT 1555 ON 10-08-04 FOR TSN 'OFDN'
% EXC0421 CPU TIME USED: 0.0257

```

Ablaufprotokoll des Auftrags JOBB (OUT.E.JOBB)

```

/WRITE-TEXT TEXT=C'*****'
*****
/WRITE-TEXT TEXT=C'***      H I E R   I S T   J O B B          ****'
***      H I E R   I S T   J O B B          ****
/WRITE-TEXT TEXT=C'*****'
*****
/WAIT-EVENT UNTIL=*JV(TIME-LIMIT=20,TIMEOUT-LABEL=W1)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:54:41
% CJC0022 WAIT COMMAND: TIMEOUT AT 15:55:02, SKIP TO TIMEOUT LABEL OR NEXT
STEP
/.W1 EXIT-JOB
% EXC0419 /LOGOFF AT 1555 ON 10-08-04 FOR TSN 'OFDP'
% EXC0421 CPU TIME USED: 0.0178

```

Ablaufprotokoll des Auftrags JOBC (OUT.E.JOBC)

```

/WRITE-TEXT TEXT=C'*****'
*****
/WRITE-TEXT TEXT=C'***      H I E R   I S T   J O B C          ****'
***      H I E R   I S T   J O B C          ****
/WRITE-TEXT TEXT=C'*****'
*****
/SET-JV-LINK JV=#LOOP
/.S1 MODIFY-JV JV=#LOOP,SET-VALUE=C'RUN'
/   WAIT-EVENT UNTIL=*JV(CONDITION=(#LOOP=C'END'),
TIME-LIMIT=20,TIMEOUT-LABEL=S1)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:55:04
% CJC0022 WAIT COMMAND: TIMEOUT AT 15:55:25, SKIP TO TIMEOUT LABEL OR NEXT
STEP
/.S1 MODIFY-JV JV=#LOOP,SET-VALUE=C'RUN'
/   WAIT-EVENT UNTIL=*JV(CONDITION=(#LOOP=C'END'),
TIME-LIMIT=20,TIMEOUT-LABEL=S1)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 15:55:25
% CAN00BY CANCELLED BY 'BTCH OFDM COGNITAS SPV'
% NRTT201 TASK TERMINATION DUE TO /CANCEL(-JOB) COMMAND
% EXC0419 /LOGOFF AT 1555 ON 10-08-04 FOR TSN 'OFDR'
% EXC0421 CPU TIME USED: 0.0245

```

Beispiel 7 (Jobvariablenmakros, ohne ONEVT-Makro)

Nachfolgendes Programm SRC.BJV zeigt die Anwendung einiger JV-Makros:

Quellprogramm SRC.BJV

```

BJV      START
        PRINT NOGEN
        BALR 3,0
        USING *,3
        STAJV JV.A,STA,120,OUTPUT=NEW,VERSION=4 _____ (1)
DTH1    DCLJV JV.A,LINK=*LINK,VERSION=1 _____ (2)
        MVC STA,NULL2
        STAJV JV.,STA,120,OUTPUT=NEW,VERSION=4 _____ (3)
DTH2    CATJV JV.A,JV.N,STATE=U,RDPASS=C'JV',VERSION=1
        SETJV JV.N,SET1,PASS=C'JV',VERSION=1
        GETJV JV.N,GET,30,PASS=C'JV',VERSION=1
DTH3    MVC GET,NULL1
        SETJV (JV.N,3,4),SET2,PASS=C'JV',VERSION=1 _____ (4)
        GETJV JV.N,GET,30,PASS=C'JV',VERSION=1
DTH4    MVC GET,NULL1
        SETJV JV.N,SET2,PASS=C'JV',VERSION=1 _____ (5)
        GETJV JV.N,GET,30,PASS=C'JV',VERSION=1
DTH5    MVC GET,NULL1
        ERAJV JV.N,VERSION=1 _____ (6)
DTH6    MVC STA,NULL2
        STAJV JV.,STA,120,OUTPUT=NEW,SELADDR=SEL1,VERSION=4 _____ (7)
DTH7    ERAJV JV.N,PASS=C'JV',VERSION=1 _____ (8)
*
DTH8    TERM
*****
        DS OF
STA     DS L120
*
SEL1    JVSEL MF=L,PASS=RDPASS
*
GET     DS L50
*
SET1    DC Y(SETEND1-SET1)
        DS CL2
        DC 'ABCDEFGHIJKLMNPOQRSTUVWXYZ'
SETEND1 EQU *
*
SET2    DC Y(SETEND2-SET2)
        DS CL2
        DC '34567'
SETEND2 EQU *
*

```



```

NULL1   DS      OCL50
NULL2   DC      120X'00'
        END
        END

```

Die im Quellprogramm mit (1) bis (8) gekennzeichneten Punkte werden im nachfolgenden Ablaufprotokoll erläutert.

Ablaufprotokoll

```

/start-assembh
% BLS0500 PROGRAM 'ASSEMBH', VERSION '01.2D01' OF '2007-08-07' LOADED
% BLS0552 COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2009. ALL RIGHTS
RESERVED
% ASS6010 V01.2D01 OF BS2000 ASSEMBH READY
//compile source=src.bjv,module-lib=ass.plamlib(elem=bjv),test-support=yes
% ASS6011 ASSEMBLY TIME: 430 MSEC
% ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
% ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
% ASS6006 LISTING GENERATOR TIME: 59 MSEC
//end
% ASS6012 END OF ASSEMBH
/load-exe from=(lib=ass.plamlib,elem=bjv),test-opt=*aid 1
% BLS0517 MODULE 'BJV' LOADED
/%insert dth1
/resume-program
STOPPED AT LABEL: DTH1 , SRC_REF: 40, SOURCE: BJV , PROC: BJV
/%display %15 _____ (1)
*** TID: 0001005C *** TSN: OFFF *****
CURRENT PC: 0000009C CSECT: BJV*****
%15 = 00000433

```

- (1) Der STAJV-Makro soll den Zustand der Jobvariablen JV.A in das Feld STA übergeben. Register 15 zeigt: JV.A ist im System nicht vorhanden.

¹ Das Kommando LOAD-EXECUTABLE-PROGRAM ersetzt ab BLSSERV 2.3 das Kommando LOAD-PROGRAM

```

/%insert dth2
/Resume-program
STOPPED AT LABEL: DTH2 , SRC_REF: 92, SOURCE: BJV , PROC: BJV
/%display sta
SRC_REF: 92 SOURCE: BJV PROC: BJV
*****
STA          =
|.JV.A.JV.DO.1.JV.E.1.JV.MON.JV.N.JV.PERM.ERROR.JV.PERM.ERROR.READ.JV.PERM.ER
R|
|OR2.JV.PERM.WAIT.JV.PROG.....|
/%display sta%x _____ (2)
CURRENT PC: 000001B4 CSECT: BJV
*****
V'00000868' = STA          + #'00000000'
00000868 (00000000) 05D1E54B C108D1E5 4BC4D64B F107D1E5 .JV.A.JV.DO.1.JV
00000878 (00000010) 4BC54BF1 07D1E54B D4D6D505 D1E54BD5 .E.1.JV.MON.JV.N
00000888 (00000020) 0ED1E54B D7C5D9D4 4BC5D9D9 D6D913D1 .JV.PERM.ERROR.J
00000898 (00000030) E54BD7C5 D9D44BC5 D9D9D6D9 4BD9C5C1 V.PERM.ERROR.REA
000008A8 (00000040) C40FD1E5 4BD7C5D9 D44BC5D9 D9D6D9F2 D.JV.PERM.ERROR2
000008B8 (00000050) 0DD1E54B D7C5D9D4 4BE6C1C9 E308D1E5 .JV.PERM.WAIT.JV
000008C8 (00000060) 4BD7D9D6 C7000000 00000000 00000000 .PROG.....
000008D8 (00000070) 00000000 00000000
.....

/%insert dth3
/Resume-program
STOPPED AT LABEL: DTH3 , SRC_REF: 188, SOURCE: BJV , PROC: BJV
/%display get _____ (3)
SRC_REF: 188 SOURCE: BJV PROC: BJV
*****
GET          = |.. ABCDEFGHIJKLMNOPQRSTUVWXYZ.....|

```

- (2) Der DCLJV-Makro katalogisiert die Jobvariable JV.A und ordnet ihr den Kettungs-namen *LINK zu. Der STAJV-Makro übergibt die Namen aller Jobvariablen, die mit „JV.“ beginnen, im Feld STA. Voreingestellt ist LIST=OLD, d.h. die Ausgabe erfolgt ohne catid und userid.
- (3) JV.A wird in JV.N umbenannt und erhält das Lesekennwort „JV“. Dieses Kennwort muss der Benutzer angeben, wenn er auf die Jobvariable zugreifen will. Der SETJV-Makro setzt JV.N auf den im Feld SET1 angegebenen Wert. Anschließend liest der Makro GETJV diesen Wert in das Feld GET ein.

```

/%insert dth4
/ resume-program
STOPPED AT LABEL: DTH4 , SRC_REF: 258, SOURCE: BJV , PROC: BJV
/%display get _____ (4)
SRC_REF: 258 SOURCE: BJV PROC: BJV
*****
GET          = |.. AB3456GHIJKLMNOPQRSTUVWXYZ.....|

/%insert dth5
/ resume-program
STOPPED AT LABEL: DTH5 , SRC_REF: 328, SOURCE: BJV , PROC: BJV
/%display get _____ (5)
SRC_REF: 328 SOURCE: BJV PROC: BJV
*****
GET          = |.. 34567.....|

/%insert dth6
/ resume-program
STOPPED AT LABEL: DTH6 , SRC_REF: 360, SOURCE: BJV , PROC: BJV
/%display %15 _____ (6)
CURRENT PC: 000006E6 CSECT: BJV
*****
%15          = 000004B1

```

- (4) Ab Byte 3 sollen 4 Bytes des Jobvariablen-Wertes mit dem Inhalt von Feld SET2 überschrieben werden. Das Feld GET zeigt den Jobvariablen-Wert nach Ausführung des SETJV-Makros.
- (5) Der Jobvariablen-Wert wird ab Startposition (=1, Standard) vollständig durch den in SET2 angegebenen Inhalt ersetzt.
- (6) Die Jobvariable soll gelöscht werden. Da das erforderliche Lesekennwort nicht angegeben wurde, enthält Register 15 einen entsprechenden Fehlercode.

```

/%insert dth7
/Resume-program

STOPPED AT LABEL: DTH7 , SRC_REF: 396, SOURCE: BJV , PROC: BJV
/%display sta
SRC_REF: 396 SOURCE: BJV PROC: BJV
*****
STA          =
|.JV.N.....|
.|
|.....|
/%display sta%x _____ (7)
CURRENT PC: 00000798 CSECT: BJV
*****
V'00000868' = STA          + #'00000000'
00000868 (00000000) 05D1E54B D5000000 00000000 00000000 .JV.N.....
00000878 (00000010) 00000000 00000000 00000000 00000000 .....
          REPEATED LINES: 4
000008C8 (00000060) 00000000 00000000 00000000 00000000 .....
000008D8 (00000070) 00000000 00000000 .....

/%insert dth8
/Resume-program

STOPPED AT LABEL: DTH8 , SRC_REF: 423, SOURCE: BJV , PROC: BJV
/%display %15 _____ (8)
CURRENT PC: 0000084A CSECT: BJV
*****
%15          = 00000000

```

- (7) Der STAJV-Makro übergibt die Namen aller Jobvariablen, die mit JV.“ beginnen und mit einem Lesekennwort geschützt sind (Selektion erfolgt durch die mit dem JVSEL-Makro generierte Operandenliste), im Feld STA. Voreingestellt ist LIST=OLD, d.h. die Ausgabe erfolgt ohne catid und userid.
- (8) Das Lesekennwort wird beim Löschen angegeben und die Jobvariable wird erfolgreich gelöscht.

Beispiel 8 (mit ONEVT-Makro)

Das Programm BONEVT setzt einen ONEVT-Makroaufruf für die Ereigniskennung ONEVTEST ab, der immer dann einen POSSIG-Aufruf verursacht, wenn eine der folgenden Bedingungen erfüllt ist:

- Jobvariable JV1 ist zum Zeitpunkt des ONEVT-Aufrufs auf 'MELDUNG' gesetzt
- JV1 wird auf 'MELDUNG' gesetzt
- Katalog wird exportiert

Die Anzahl der POSSIG-Aufrufe ist auf drei beschränkt (COUNT=3), d.h. dieser ONEVT-Makroaufruf registriert maximal 3 mal „Bedingung erfüllt“. Mit SOLSIG wird ein Signal angefordert und anschließend die übergebene Post Code-Information geprüft. Abhängig vom Bedingungsergebnis gibt das Programm dann nach jedem SOLSIG eine Meldung aus (MELD1, MELD2 oder FMELD).

Programm BONEVT

```

BONEVT  START
        BALR 3,0
        USING *,3
        PRINT NOGEN
ANF     DCLJV JV1,VERSION=1           "JV1 definieren"
        ENAEI EINAME=ONEVTEST,EIIDRET=KUKE,SCOPE=GLOBAL
        ONEVT 'JV1=' 'MELDUNG' ',EIID=KUKE,POST='B1',COUNT=3,VERSION=1
CHECK   CHKEI EIID=KUKE             "Warteschlange prüfen"
        LR 5,1
        CMD '%DISPLAY %15,%5'
SIGNAL  MVC EMPF,NULL              "Signal anfordern"
        SOLSIG EIID=KUKE,COND=UNCOND,RPOSTAD=EMPF,LIFETIM=10
*** POST-CODE PRUEFEN *****
PRUEF   CLI ANZ,X'14'              "Ereignisanzeige"
        BNE FMELD
        CLC ONKEN,POST             "POST-Wert"
        BNE FMELD
        CLI ERG,X'00'              "JV1 gesetzt"
        BE MELD1
        CLI ERG,X'08'              "Katalog exportiert"
        BE MELD2
        B FMELD
*** MELDUNGEN, DIE AUSGEGEBEN WERDEN *****
MELD1   WROUT MLDG1,ENDE          "JV1 gesetzt"
        B CHECK
MELD2   WROUT MLDG2,ENDE          "Katalog exportiert"
        B FRAGE
FMELD   WROUT FEHLER,ENDE         "kein POSSIG durch ONEVT empfangen"
        B FRAGE
FRAGE   WRTRD ABFRAG,,EINB,,5,ENDE "Programm beenden ?"

```

```

        CLI   ANTW,'N'
        BE    DONEVT           "Programm nicht beenden"
        CLI   ANTW,'J'
        BE    ENDE             "beenden"
        B     FRAGE            "Abfrage wiederholen"
*
DONEVT  DONEVT EIID=KUCHE,VERSION=1
        DISEI EIID=KUCHE           "Ereigniskennung löschen"
        ERAJV JV1,VERSION=1
        B     ANF
ENDE    TERM
*** DEFINITIONEN *****
EMPF    DS    0F
ANZ     DS    CL1
ERG     DS    CL1
ONKEN   DS    CL2
POST    DC    C'B1'
KUCHE   DC    C'KURZ'
NULL    DC    F'0'
MLDG1   DC    Y(END1-MLDG1)
        DS    L2
        DC    X'01'
        DC    'JV1 WURDE GESETZT'
END1    EQU   *
MLDG2   DC    Y(END2-MLDG2)
        DS    L2
        DC    X'01'
        DC    'KATALOG WURDE EXPORTIERT'
END2    EQU   *
FEHLER  DC    Y(FEND-FEHLER)
        DS    L2
        DC    X'01'
        DC    '*** FEHLER ***'
FEND    EQU   *
ABFRAG  DC    Y(ABEND-ABFRAG)
        DS    CL2
        DC    X'01'
        DC    'PROGRAMM BEENDEN?(J/N)'
ABEND   EQU   *
EINB    DS    OCL5
        DS    CL4
ANTW    DS    CL1
        END

```

Die Wirkung des ONEVT-Makros wird in diesem Beispiel an der Bedingung „JV1 wird auf 'MELDUNG' gesetzt“ demonstriert:

Ablaufprotokoll LST.BONEVT des Programms BONEVT

```

(IN)      mod-job-options logging=*par(listing=*yes)
(IN)      delete-system-file system-file=*omf
(IN)      start-assembly----- (1)
(OUT)    % PROGRAM 'ASSEMBH', VERSION '01.2D01' OF '2007-08-07' LOADED
(OUT)    % COPYRIGHT (C) FUJITSU TECHNOLOGY SOLUTIONS 2009.
( )      ALL RIGHTS RESERVED
(OUT)    % ASS6010 V01.2D01 OF BS2000 ASSEMBH READY
(IN)      compile source=bonevt,mod-lib=ass.plamlib(bonevt),test-support=yes
(OUT)    ASS6011 ASSEMBLY TIME: 604 MSEC
(OUT)    % ASS6018 0 FLAGS, 0 PRIVILEGED FLAGS, 0 MNOTES
(OUT)    % ASS6019 HIGHEST ERROR-WEIGHT: NO ERRORS
(OUT)    % ASS6006 LISTING GENERATOR TIME: 66 MSEC
(IN)      end
(OUT)    % ASS6012 END OF ASSEMBH
(IN)      load-exe from=(lib=ass.plamlib,elem=bonevt),test-opt=*aid----- (2)
(OUT)    % BLS0517 MODULE 'BONEVT' LOADED
(IN)      %insert check----- (3)

(IN)      resume-program
(OUT)
(OUT)    STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT
(IN)      mod-jv jv=jv1,set-val='MELDUNG'----- (4)
(IN)      mod-jv jv=jv1,set-val='MELDUNG'
(IN)      mod-jv jv=jv1,set-val='MELDUNG'
(IN)      mod-jv jv=jv1,set-val='MELDUNG'
(IN)      mod-jv jv=jv1,set-val='MELDUNG'

(IN)      resume-program
(OUT)    *** TID: 0001005E *** TSN: 0FDB *****
(NL)     CURRENT PC: 000000E0 CSECT: BONEVT *****
(NL)     %15 = 2C000000----- (5)
(NL)     %5 = 00000003----- (6)

```

- (1) Das Programm BONEVT wird mit dem Assembler ASSEMBH übersetzt und in der Bibliothek ASS.PLAMLIB abgelegt.
- (2) Das Programm wird geladen und soll mit der Testhilfe AID getestet werden.
- (3) Mit dem AID-Kommando %INSERT wird der Testpunkt CHECK vereinbart.
- (4) Die Jobvariable JV1 wird fünfmal auf 'MELDUNG' gesetzt.
- (5) Der CHKEI-Makro zeigt die POSSIG-Warteschlange (Register 15).
- (6) Der CHKEI-Makro zeigt, dass die Anzahl der POSSIG-Aufrufe 3 ist (Register 5).

```

(OUT)   JV1 WURDE GESETZT _____ (7)
(OUT)
(OUT)   STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT

(IN)    %display EMPF%X _____ (8)
(OUT)   CURRENT PC: 000000A8   CSECT: BONEVT *****
(NL)    V'0000028C' = EMPF      + #'00000000'
(NL)    0000028C (00000000) 1400C2F1                      ..B1

(IN)    resume-program
(OUT)   CURRENT PC: 000000E0   CSECT: BONEVT *****
(NL)    %15                      = 2C000000
(NL)    %5                        = 00000002
(OUT)   JV1 WURDE GESETZT
(OUT)   STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT
(IN)    resume-program
(OUT)   %15                      = 2C000000
(NL)    %5                        = 00000001
(OUT)   JV1 WURDE GESETZT
(OUT)   STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT
(IN)    resume-program
(OUT)   %15                      = 30000000
(NL)    %5                        = 8F0000AC
(OUT)   ***   FEHLER   *** _____ (9)

(OUT)   PROGRAMM BEENDEN?(J/N)
(IN)    N
(OUT)
(OUT)   STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT

```

- (7) Der erste POSSIG wurde mit SOLSIG angefordert. Das Bedingungsergebnis wird abgefragt und dementsprechend eine Meldung ausgegeben.
- (8) Das Empfangsfeld EMPF enthält nach dem SOLSIG-Aufruf den übergebenen Post Code:
- | | |
|---------|---|
| X'14' | POSSIG wurde durch einen ONEVT-Makro verursacht |
| X'00' | Bedingungsergebnis „Jobvariable gesetzt“ |
| X'C2F1' | ONVEVT-Kennungswert B1 |
- (9) Nachdem alle 3 POSSIG-Aufrufe mit SOLSIG angefordert wurden, ist die Warteschlange leer (Register 15 des CHKEI-Makros: SI=X'30'). Die Prüfung des Post Codes nach einem weiteren SOLSIG führt zur Meldung „FEHLER“. Die Abfrage wird mit „N“ beantwortet, d.h. das Programm wird ab ANF noch einmal durchlaufen - ein erneuter ONEVT-Makro wird abgesetzt.


```

(IN)      mod-jv jv=jv1,set-val='MELDUNG' _____ (10)
(IN)      mod-jv jv=jv1,set-val='MELDUNG'

(IN)      resume-program
(OUT)     %15          = 2C000000
(NL)     %5           = 00000002 _____ (11)
(OUT)     JV1 WURDE GESETZT
(OUT)
(OUT)     STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT
(IN)      del-jv jv=jv1 _____ (12)
(OUT)     % JVS04A3 ERROR WHEN DELETING JOB VARIABLE ':4V05:$COGNITAS.JV1'
(OUT)     % JVS0447 JV NAME BEING USED BY CJC FUNCTION. COMMAND REJECTED

(IN)      resume-program
(OUT)     %15          = 2C000000
(NL)     %5           = 00000001
(OUT)     JV1 WURDE GESETZT
(OUT)
(OUT)     STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT
(IN)      resume-program
(OUT)     %15          = 30000000
(NL)     %5           = 8F0000AC
(OUT)     *** FEHLER ***
(OUT)     PROGRAMM BEENDEN?(J/N)
(IN)      N
(OUT)
(OUT)     STOPPED AT LABEL: CHECK ,SRC_REF: 84, SOURCE: BONEVT, PROC: BONEVT
(IN)      show-jv-attr jv=jv1 _____ (13)
(OUT)     0000000 :4V05:$COGNITAS.JV1
(NL)     SUM 00001 JV'S; JV-VALUE = 00000000 BYTES
(IN)     show-jv-attr jv=jv1,inf=*all-attr
(OUT)     000000 :4V05:$COGNITAS.JV1
(NL)     USER-ACC = ALL-USERS ACCESS = WRITE
(NL)     CRE-DATE = 2010-08-04 EXPIR-DATE = 2010-08-04
(NL)     CRE-TIME = 16:51:15 EXPIR-TIME = 00:00:00
(NL)     READ-PASS = NONE
(NL)     WRITE-PASS = NONE
(NL)     SUM 00001 JV'S; JV-VALUE = 00000000 BYTES

```

- (10) Das Ereignis „JV1 setzen“ wird zweimal signalisiert.
- (11) Die Anzahl der POSSIG-Aufrufe ist 2.
- (12) Der Versuch, die Jobvariable JV1 zu löschen wird abgewiesen, da Jobvariablen, die in Makros (und Kommandos) der bedingungsabhängigen Auftragssteuerung verwendet werden, gegen Löschen geschützt sind.
- (13) Die Merkmale der Jobvariablen JV1 werden ausgegeben.

```
(IN) show-jv jv=jv1 _____ (14)
(OUT) % JVS04B2 SPECIFIED JOB VARIABLE SUBSTRING EMPTY OR ILLEGAL.
( ) COMMAND REJECTED
(IN) resume-program
(OUT) %15 = 30000000
(NL) %5 = 8F0000AC
(OUT) *** FEHLER ***
(OUT) PROGRAMM BEENDEN?(J/N)
(IN) J
(IN) del-jv jv=jv1 _____ (15)
(IN) show-j-attr jv=jv1
(OUT) % JVS0433 REQUESTED JOB VARIABLE NOT CATALOGED. COMMAND REJECTED
(IN) assign-syslst to=*prim
```

(14) Der Jobvariablen JV1 ist kein Wert zugewiesen.

(15) Die Jobvariable JV1 kann gelöscht werden.

Beispiel 9

In diesem Beispiel wird die Anwendung des Kommandos MODIFY-JV-CONDITIONALLY und der Makros CSWJV, LNKJV und TIMJV gezeigt. Eine Prozedur und ein ENTER-Auftrag tauschen Informationen aus.

Das Beispiel wird mit dem Kommando /CALL-PROCEDURE PROC.BSP9 gestartet. Das ablauf-fähige Modul TESTJV muss in der Bibliothek ASS.PLAMLIB abgelegt sein.

Prozedur PROC.BSP9

```

/      BEG-PROC LOGGING=*CMD
/      ASSIGN-SYSLST TO=LST.PROC
/      MODIFY-JOB-OPTION LOGGING=*PARAMETER(LISTING=*YES)
/      SET-JV-LINK JV=JV.E.1
/      MODIFY-JV   JV=JV.E.1,SET-VALUE=C'***'
/      SET-JV-LINK JV=JV.DO.1
/      MODIFY-JV   JV=JV.DO.1,SET-VALUE=C'***'
/      SET-JV-LINK JV=JV.MON
/      ENTER-JOB  FROM-FILE=TEST.ENTER,MONJV=JV.MON,JOB-CLASS=JCBATCH
/.SKIP1  SHOW-JV  JV=JV.MON
/      WAIT-EVENT *JV(CONDITION=((JV.MON,1,2)=C'$R'), -
/                TIME-LIMIT=90,TIMEOUT-LABEL=SKIP1)
/.PRUEF  MODIFY-JV-CONDITIONALLY JV=(JV.DO.1),-
/                IF-VALUE=C'GESTARTET',SET-VALUE=C'VERSTANDEN',-
/                LABEL=WEIT1
/      SKIP-COMMANDS TO-LABEL=PRUEF
/.WEIT1  SHOW-JV  JV=JV.DO.1
/      SHOW-JV  JV=JV.E.1
/      SHOW-JV  JV=JV.MON
/      SHOW-JOB-STAT *MONJV(JV.MON)
/      WAIT-EVENT *JV(CONDITION=((JV.DO.1)=C'WEITER GEHTS'), -
/                TIME-LIMIT=15,TIMEOUT-LABEL=WEIT1)
/.WEIT2  MODIFY-JV  JV=JV.E.1,SET-VALUE=C'PROGRAMM FORTSETZEN'
/.SKIP3  SKIP-COMMANDS TO-LABEL=ENDE,-
/                IF=*JV(CONDITION=((JV.E.1,12,4)=C'ENDE'))
/      SHOW-USER-STATUS
/      SHOW-JV  JV=JV.E.1
/      SHOW-JV  JV=JV.MON
/      WAIT-EVENT UNTIL=*JV(TIME-LIMIT=45,TIMEOUT-LABEL=SKIP3)
/.ENDE  REMARK **** TESTJOB BEENDET ****
/      SHOW-JV  JV=JV.MON
/      ASSIGN-SYSLST TO=*PRIMARY
/      MODIFY-JOB-OPTION
/      END-PROCEDURE

/      BEG-PROC LOGGING=*CMD
/      ASSIGN-SYSLST TO=LST.PROC
/      MODIFY-JOB-OPTION LOGGING=*PARAMETER(LISTING=*YES)

```

Datei TEST.ENTER

```

/.TESTJV SET-LOGON-PARAMETERS
/      ASSIGN-SYSOUT TO=OUT.E.TESTJV
/      SET-JV-LINK JV-NAME=JV.DO.1
/      SET-JV-LINK JV-NAME=JV.PROG
/      MODIFY-JV JV=JV.DO.1,SET-VALUE=C'GESTARTET'
/      START-EXE FROM-FILE=(LIB=ALF.ASS.PLAMLIB,ELEM=TESTJV),-
/          TEST-OPTIONS=*AID,MONJV=JV.PROG
/      SHOW-JV JV=JV.MON
/      SHOW-JV JV=JV.PROG
/      SKIP-COMMANDS TO-LABEL=ENDE
/      SET-JOB-STEP
/      MODIFY-JV JV=JV.E.1,SET-VALUE=C'ABNORMALES ENDE'
/.ENDE  WAIT-EVENT UNTIL=*JV(TIME-LIMIT=120,TIMEOUT-LABEL=TIME)
/.TIME  MODIFY-JV JV=(JV.E.1,12,4),SET-VALUE=C'ENDE'
/      EXIT-JOB SYSTEM-OUTPUT=*NONE

```

Quellprogramm SRC.TESTJV

```

TESTJV  START
        TITLE 'TESTPROGRAMM FUER CSWJV'
*
* REGISTERZUWEISUNG
        BALR 3,0
        USING *,3
        PRINT NOGEN
*
* JOBVARIABLE JV.E.1 WIRD DER LINKNAME *ENTER ZUGEWIESEN
* DER LINKNAME *ENTER WIRD IM FELD LNKBER ABGELEGT
* MIT DEM MAKRO CMD WIRD DER INHALT VON LNKBER AUSGEGEBEN
*
        DCLJV JV.E.1,LINK=*ENTER,VERSION=1
        LNKJV LNKBER,LINK=*ENTER,VERSION=1
        CMD  '%DISPLAY','LNKBER'
*
* DER WERT VON JV.DO.1 WIRD MIT DEM INHALT VON VERGL VERGLICHEN
* BEI GLEICHHEIT SOLL DER SETZWERT NEUWERT GESETZT WERDEN
*
CSWJV   CSWJV JV.DO.1,VERGL,NEUWERT,VERSION=1
*
* MELDUNG MELD1 *** CSWJV AUSGEFUEHRT *** WIRD AUSGEGEBEN
*   UND MELDUNG UND ZEITSTEMPEL IN MONJV GESETZT
*
        WROUT MELD1,TERM
        TIMJV MF=S,MONJV=*SMONJVJ,TIMESTAMP=*SET,DESCRIPTOR='TESTJV',-
            INFO='*** CSWJV AUSGEFUEHRT ***'
        CMD  'SHOW-JV','JV=*LINK(SMONJVJ)'

```

```

*
* DAS FELD JVDOW WIRD MIT DEM INHALT VON JV.DO.1 VERSORGT
* VERGLEICH, OB JVDOW DEN WERT 'WEITER GEHTS' ENTHAELT
*
      GETJV JV.DO.1,JVDOW,30,VERSION=1
      CLC   JVDOW+4(12),='WEITER GEHTS'
      BE    FORTSETZ
*
* DA BEI UNGLEICHHEIT DER WERT VON JV.DO.1 IN DAS FELD VERGL
* UEBERTRAGEN WIRD, WIRD VERGL WIEDER AUF 'VERSTANDEN' GESETZT
*
      MVC   VERGLF,='VERSTANDEN'
      VPASS 1
      B     CSWJV
*
* DER BEREICH JVEW WIRD MIT DEM WERT VON JV.E.1 VERSORGT
* VERGLEICH, OB JVEW DEN WERT 'PROGRAMM FORTSETZEN' ENTHAELT
*
FORTSETZ EQU   *
      GETJV JV.E.1,JVEW,30,VERSION=1
      CLC   JVEW+4(19),='PROGRAMM FORTSETZEN'
      BNE   FORTSETZ
*
* AUSGABE DER MELDUNG *** PROGRAMM FORTGESETZT ***
*   UND MELDUNG UND ZEITSTEMPEL IN MONJV GESETZT
*
      WROUT MELD2,TERM
      TIMJV MF=S,MONJV=*SMONJVJ,TIMESTAMP=*SET,DESCRIPTOR='TESTJV', -
          INFO=C'*** PROGRAMM FORTGESETZT ***'
      CMD   'SHOW-JV','JV=*LINK(SMONJVJ)'
*
* AUSGABE DER MELDUNG *** PROGRAMM ENDSPURT ***
*   UND MELDUNG UND ZEITSTEMPEL IN MONJV GESETZT
*
BEENDEN EQU   *
      WROUT MELD3,TERM
      TIMJV MF=S,MONJV=*SMONJVJ,TIMESTAMP=*SET,DESCRIPTOR='TESTJV', -
          INFO=C'*** PROGRAMM ENDSPURT ***'
      CMD   'SHOW-JV','JV=*LINK(SMONJVJ)'
*
* DER JOBVARIABLEN JV.E.1 WIRD DER WERT 'NORMALES ENDE' ZUGEORDNET
*   UND ZEITSTEMPEL VOM PROGRAMMENDE IN MONJV GESETZT
*
      SETJV JV.E.1,EWERT,VERSION=1
      TIMJV MF=S,MONJV=*SMONJVJ,TIMESTAMP=*SET,DESCRIPTOR='TESTJV', -
          INFO=C'*** PROGRAMM BEENDET ***'
TERM     TERM

```

```
*
*      DEFINITIONEN
*
VERGL  DC    Y(END1-VERGL)
       DS    CL2
VERGLF DC    'VERSTANDEN'
END1   EQU   *
*
NEUWERT DC    Y(END2-NEUWERT)
       DS    CL2
       DC    'WEITER GEHTS'
END2   EQU   *
*
EWERT  DC    Y(END3-EWERT)
       DS    CL2
       DC    'NORMALES ENDE'
END3   EQU   *
*
MELD1  DC    Y(END4-MELD1)
       DS    CL2
       DC    X'01'
MELD1TXT DC  '*** CSWJV AUSGEFUEHRT ***'
END4   EQU   *
*
MELD2  DC    Y(END5-MELD2)
       DS    CL2
       DC    X'01'
MELD2TXT DC  '*** PROGRAMM FORTGESETZT ***'
END5   EQU   *
*
MELD3  DC    Y(END6-MELD3)
       DS    CL2
       DC    X'01'
MELD3TXT DC  '*** PROGRAMM ENDSPURT ***'
END6   EQU   *
*
JV DOW DS    CL30
*
JVEW  DS    CL30
*
LNKBER DS    CL63
      END
```

Ablaufprotokoll LST.PROC der Prozedur PROC.BSP9

```

(IN) /          MODIFY-JOB-OPTION LOGGING=*PARAMETER(LISTING=*YES)
(IN) /          SET-JV-LINK JV-NAME=JV.E.1
(IN) /          MODIFY-JV   JV=JV.E.1,SET-VALUE=C'***'
(IN) /          SET-JV-LINK JV-NAME=JV.DO.1
(IN) /          MODIFY-JV   JV=JV.DO.1,SET-VALUE=C'***'
(IN) /          SET-JV-LINK JV-NAME=JV.MON
(IN) /          ENTER-JOB  FROM-FILE=TEST. ENTER,MONJV=JV.MON, JOB-CLASS=
JCBATCH
(OUT) % JMS0066 JOB 'TESTJV' ACCEPTED ON 10-08-08 AT 09:22, TSN = OFFJ
(IN) / .SKIP1  SHOW-JV  JV=JV.MON
(OUT) $S 00FFJ4V05  J0672010-08-08072201
( )
(IN) /          WAIT-EVENT *JV(CONDITION=((JV.MON,1,2)=C'$R'),
TIME-LIMIT=90,TIMEOUT-LABEL=SKIP1)
(OUT) % CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 09:22:01
(OUT) % CJC0021 WAIT COMMAND: CONDITION = TRUE AT 09:22:01
(IN) / .PRUEF  MODIFY-JV-CONDITIONALLY JV=(JV.DO.1),
IF-VALUE=C'GESTARTET',SET-VALUE=C'VERSTANDEN',LABEL=WEIT1
(IN) /          SKIP-COMMANDS TO-LABEL=PRUEF
(IN) / .PRUEF  MODIFY-JV-CONDITIONALLY JV=(JV.DO.1),
IF-VALUE=C'GESTARTET',SET-VALUE=C'VERSTANDEN',LABEL=WEIT1
(IN) /          SKIP-COMMANDS TO-LABEL=PRUEF
(IN) / .PRUEF  MODIFY-JV-CONDITIONALLY JV=(JV.DO.1),
IF-VALUE=C'GESTARTET',SET-VALUE=C'VERSTANDEN',LABEL=WEIT1
(IN) / .WEIT1  SHOW-JV  JV=JV.DO.1
(OUT) VERSTANDEN
(IN) /          SHOW-JV  JV=JV.E.1
(OUT) ***
(IN) /          SHOW-JV  JV=JV.MON
(OUT) $R 00FFJ4V05  J0672010-08-08072201
( )
(IN) /          SHOW-JOB-STAT *MONJV(JV.MON)
(OUT) TSN:      OFFJ      TYPE:    2 BATCH      NOW:      2010-08-08.092201
(NL) JOBNAME:  TESTJV    PRI:     9 255      SPOOLIN: 2010-08-08.0922
(NL) USERID:  COGNITAS  JCLASS:  JCBATCH   LOGON:   2010-08-08.0922
(NL) ACCNB:   89001     CPU-MAX:  200      CPU-USED:000000.0239
(NL) REPEAT:  NO        RERUN:   NO        FLUSH:   NO
(NL) MRSCAT:          HOLD:    NO        START:   SOON
(NL) TID:     0001005F  UNP/Q#:  00/001
(NL) CMD:     EXECUTE
(NL) ORIGFILE::4V05:$COGNITAS.TEST. ENTER
(NL) MONJV:   :4V05:$COGNITAS.JV.MON
(IN) /          WAIT-EVENT *JV(CONDITION=((JV.DO.1)=C'WEITER GEHTS'),
( )
TIME-LIMIT=15,TIMEOUT-LABEL=WEIT1)
(OUT) % CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 09:22:01
(OUT) % CJC0021 WAIT COMMAND: CONDITION = TRUE AT 09:22:01

```

```

(IN)      /.WEIT2  MODIFY-JV JV=JV.E.1,SET-VALUE=C'PROGRAMM FORTSETZEN'
(IN)      /.SKIP3  SKIP-COMMANDS TO-LABEL=ENDE,
( )      IF=*JV(CONDITION=((JV.E.1,12,4)=C'ENDE'))
(OUT)    % CJC0011 SKIP COMMAND: CONDITION = FALSE
(IN)      /        SHOW-USER-STATUS
(OUT)    NAME      TSN TYPE      PRI      CPU-USED CPU-MAX ACCOUNT#
(NL)     OFFF 3 DIALOG  0 240      0.1275  32767 89001
(NL)     COG2  OFFG 3 DIALOG  0 240      0.2489  32767 89001
(NL)     TESTJV OFFJ 2 BATCH  9 255      0.0394   200 89001
(OUT)    % SPS0171 NO LOCAL SPOOLOUT JOB PRESENT
(OUT)    % SPS0420 RSO WARNING : SOME RSO PRINT-JOBS CANNOT BE DISPLAYED
(IN)      /        SHOW-JV JV=JV.E.1
(OUT)    NORMALES  ENDE
(IN)      /        SHOW-JV JV=JV.MON
(OUT)    $R 00FFJ4V05 J0672010-08-080722012010-08-08072201TESTJV
*** PROGRAMM BEENDET ***
( )
(IN)      /        WAIT-EVENT UNTIL=*JV(TIME-LIMIT=45,TIMEOUT-LABEL=SKIP3)
(OUT)    % CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 09:22:01
(OUT)    % CJC0022 WAIT COMMAND: TIMEOUT AT 09:22:46, SKIP TO TIMEOUT LABEL
OR NEXT STEP
(IN)      /.SKIP3  SKIP-COMMANDS TO-LABEL=ENDE,
IF=*JV(CONDITION=((JV.E.1,12,4)=C'ENDE'))
(OUT)    % CJC0010 SKIP COMMAND: CONDITION = TRUE
(IN)      /.ENDE  REMARK **** TESTJOB BEENDET ****
(IN)      /        SHOW-JV JV=JV.MON
(OUT)    $R 00FFJ4V05 J0672010-08-080722012010-08-08072201TESTJV
*** PROGRAMM BEENDET ***
( )
(IN)      /        ASSIGN-SYSLST TO=*PRIMARY

```


Ablaufprotokoll OUT.E.TESTJV des Jobs TEST.ENTER

```

/          SET-JV-LINK JV-NAME=JV.DO.1
/          SET-JV-LINK JV-NAME=JV.PROG
/          MODIFY-JV JV=JV.DO.1,SET-VALUE=C 'GESTARTET'
/          START-EXE FROM-FILE=(LIB=ASS.PLAMLIB,ELEM=TESTJV),
TEST-OPTIONS=*AID,MONJV=JV.PROG
% BLS0517 MODULE 'TESTJV' LOADED
*** TID: 0001005F *** TSN: OFFJ
*****
*****
SRC_REF:    78 SOURCE: TESTJV  PROC: TESTJV
*****
LNKBER      = |.*ENTER
:4V05:$COGNITAS.JV.E.1.....|
*** CSWJV AUSGEFUEHRT ***
$R 00FFJ4V05    J0672010-08-080722012010-08-08072201TESTJV          ***
CSWJV AUSGEFUEHRT ***
*** PROGRAMM FORTGESETZT ***
$R 00FFJ4V05    J0672010-08-080722012010-08-08072201TESTJV          ***
PROGRAMM FORTGESETZT ***
*** PROGRAMM ENDSPURT ***
$R 00FFJ4V05    J0672010-08-080722012010-08-08072201TESTJV          ***
PROGRAMM ENDSPURT ***
/          SHOW-JV JV=JV.MON
$R 00FFJ4V05    J0672010-08-080722012010-08-08072201TESTJV          ***
PROGRAMM BEENDET ***
/          SHOW-JV JV=JV.PROG
$T          P
/          SKIP-COMMANDS TO-LABEL=ENDE
/.ENDE  WAIT-EVENT UNTIL=*JV(TIME-LIMIT=120,TIMEOUT-LABEL=TIME)
% CJC0020 WAIT COMMAND: TASK ENTERED WAIT STATE AT 09:22:01
% CJC0022 WAIT COMMAND: TIMEOUT AT 09:24:01, SKIP TO TIMEOUT LABEL OR NEXT
STEP
/.TIME  MODIFY-JV JV=(JV.E.1,12,4),SET-VALUE=C 'ENDE'
/          EXIT-JOB SYSTEM-OUTPUT=*NONE
% EXC0419 /LOGOFF AT 0924 ON 10-08-08 FOR TSN 'OFFJ'
% EXC0421 CPU TIME USED: 0.0540

```

6 Meldungen

6.1 Kommandoebene

Auf Kommandoebene werden die Fehlermeldungen, angeführt von einem aus sieben Zeichen bestehenden Meldungsschlüssel, ausgegeben. Dieser Meldungsschlüssel setzt sich zusammen aus drei Zeichen für die Meldungsklasse und aus vier Zeichen für den Fehlercode. Bei den Meldungen für die Jobvariablen-Funktionen können JVS, JPM und CJC als Meldungsklasse auftreten.

Die folgende Übersicht enthält die Bereiche der Meldungsschlüssel mit der jeweils zugehörigen Ursprungskomponente.

Meldungsklasse	Bereich der Fehlercodes	Meldung durch
CJC	0000 - 0200	Bedingungsabhängige Auftragssteuerung
JPM	0200 - 0500	Auftrags-/Programmüberwachung
JVS	0400 - 040F	Catalog Management System
JVS	0410 - 041F	Catalog Management System
JVS	0420 - 042F	PAM
JVS	0430 - 043F	Catalog Management System
JVS	0440 - 044F	Verarbeitung durch CREATE-JV/MODIFY-JV-ATTRIBUTES (CATJV)
JVS	0450 - 045F	Verarbeitung durch SET-/REMOVE-/SHOW-JV-LINK (DCLJV)
JVS	0460 - 046F	Verarbeitung durch DELETE-JV (ERAJV)
JVS	0470 - 047F	Verarbeitung durch SHOW-JV/MODIFY-JV-CONDITIONALLY (GETJV)
JVS	0480 - 048F	Verarbeitung durch MODIFY-JV/MODIFY-JV-CONDITIONALLY (SETJV)
JVS	0490 - 049F	Verarbeitung durch SHOW-JV-ATTRIBUTES (STAJV)
JVS	04A0 - 04AF	allgemeine Kommando-Verarbeitung

Tabelle 31: Meldungsschlüssel und Ursprungskomponente

JVS	04B0 - 04BF	verschiedene JV-Verarbeitung
JVS	04C8 - 04CA	Catalog Management System
JVS	04D0 - 04D8	MONJV-Handler
JVS	04E0 - 04E1	Kommandoprozessor

Tabelle 31: Meldungsschlüssel und Ursprungskomponente

Mit dem Kommando HELP-MSG-INFORMATION wird der Meldungstext für den angegebenen Meldungsschlüssel nach SYSOUT ausgegeben in der aktuellen Task-Sprache aus der für JV aktivierten Meldungsdatei ausgegeben. Mit INFORMATION-LEVEL=*MAXIMUM werden zusätzlich Bedeutungs- und Maßnahmetext ausgegeben. Im Operanden LANGUAGE kann auch ein Sprachschlüssel (D für Deutsch bzw. E für Englisch) für die gewünschte Ausgabesprache angegeben werden.

Die Meldungen von JV finden Sie auch über eine HTML-Anwendung auf dem Manual-Server (URL: <http://manuals.ts.fujitsu.com>) und auf der DVD „BS2000/OSD SoftBooks“.

6.2 Makroebene

Auf Makroebene wird standardmäßig ein Returncode in den beiden rechten Bytes von Register 15 bzw. in einem dafür vorgesehenen Feld der Parameterliste des aufgerufenen Makros (im Standardheader) abgelegt. Die Bedeutung der einzelnen Returncodes ist in dem Makro IDEJVS (linkes Byte SI, rechtes Byte RS) bzw. im Standardheader des aufgerufenen Makros kommentiert:

```

MELD      IDEJVS
1 *****
1 *
1 *      VERSION 401
1 *      #INTF REFTYPE=REQUEST,                                C
1 *      INTNAME=ERRJV,INTCOMP=000
1 *****
1 *      JVS RETURN CODES *****
1 IDRC0000 EQU  X'0000' REQUESTED JVS FUNCTION PROCESSED SUCESSFULLY.
1 *
1 *      JVS RETURN CODES FROM CMS
1 *
1 IDRC0401 EQU  X'0401' REQUESTED CATALOG NOT ACCESSIBLE.
1 IDRC0402 EQU  X'0402' REQUESTED CATALOG IN QUIET MODE.
1 IDRC0403 EQU  X'0403' MRSCAT CONTAINS ILLEGAL INFORMATION.
1 IDRC0404 EQU  X'0404' SYSTEM ERROR IN CMS.                                202
1 IDRC0405 EQU  X'0405' SYSTEM ERROR IN MC.
1 IDRC0406 EQU  X'0406' MASTER EXCH OPERATION ABORT.                        103
1 IDRC0407 EQU  X'0407' MASTER EXCH WRITE DENIED.                          103
1 IDRC0410 EQU  X'0410' RC FROM SUBFUNCTION.                                103
1 IDRC041F EQU  X'041F' TASK RPO1 NO DUMP.                                  201
1 IDRC0412 EQU  X'0412' REQUESTED CATALOG NOT FOUND.                        202
1 IDRC0430 EQU  X'0430' SYSTEM ERROR IN CMS ($GETMEM).
1 IDRC0431 EQU  X'0431' INVALID PARAMETER.
1 IDRC0432 EQU  X'0432' SYSTEM ERROR IN CMS.
1 IDRC0433 EQU  X'0433' REQUESTED JOB VARIABLE NOT CATALOGED.
1 IDRC0434 EQU  X'0434' SYSTEM ERROR IN CMS.
1 IDRC0435 EQU  X'0435' JOB VARIABLE NOT SHARABLE.
1 IDRC0436 EQU  X'0436' SYSTEM ERROR IN CMS.
1 * IDRC0437 EQU  X'0437' SYSTEM ERROR IN CMS.
1 IDRC0438 EQU  X'0438' 6 OR LESS BLOCKS REMAINING FOR CATALOG.
1 IDRC0439 EQU  X'0439' NO MORE SPACE AVAILABLE FOR CATALOG.
1 IDRC043B EQU  X'043B' SYSTEM ERROR IN CMS.
1 IDRC043C EQU  X'043C' CATALOG FILE SPACE EXHAUSTED.
1 IDRC043D EQU  X'043D' SYSTEM ERROR: TSOSCAT DESTROYED.
1 IDRC043E EQU  X'043E' JV TO BE CREATED ALREADY EXISTS.
1 *
1 *      JVS RETURN CODES FROM DQPAM
1 *
1 * IDRC0421 EQU  X'0421' SYSTEM ERROR: NO I/O SLOT AVAILABLE.            005
1 IDRC0422 EQU  X'0422' SYSTEM ERROR IN CMS.

```

```

1 IDRC0423 EQU X'0423' SYSTEM ERROR IN PAM.
1 IDRC0424 EQU X'0424' SYSTEM ERROR: SYSTEM ADDRESS SPACE EXHAUSTED.
1 IDRC0425 EQU X'0425' SYSTEM ERROR IN CMS.
1 IDRC0426 EQU X'0426' SYSTEM ERROR IN CMS.
1 IDRC0427 EQU X'0427' SYSTEM ERROR IN PAM.
1 IDRC0428 EQU X'0428' SYSTEM ERROR IN PAM. 005
1 IDRC0429 EQU X'0429' SYSTEM ERROR IN PAM.
1 * IDRC042B EQU X'042B' SYSTEM ERROR IN PAM. 005
1 IDRC042C EQU X'042C' SYSTEM ERROR IN PAM.
1 IDRC042D EQU X'042D' SYSTEM ERROR IN PAM. 005
1 IDRC042F EQU X'042F' SYSTEM ERROR: RESIDENT PAGE SPACE EXHAUSTED.
1 *
1 * JVS RETURN CODES FROM JVCATEX
1 *
1 IDRC0440 EQU X'0440' INVALID JVNAME1.
1 IDRC0441 EQU X'0441' INVALID JVNAME2.
1 IDRC0442 EQU X'0442' NEGATIVE RETENTION PERIOD.
1 * IDRC0443 EQU X'0443' STATE=NEW AND ACCESS=READ. 100
1 IDRC0444 EQU X'0444' JVNAME1 OR JVNAME2 ALREADY CATALOGED.
1 IDRC0445 EQU X'0445' INVALID RETENTION PERIOD SPECIFICATION.
1 * IDRC0446 EQU X'0446' RENAME JV NOT SUCCESSFUL. 005
1 IDRC0447 EQU X'0447' JVNAME1 IS IN USE BY CJC. 001
1 IDRC0448 EQU X'0448' MONJV MAY NOT BE CHANGED. 001
1 IDRC0449 EQU X'0449' ONLY STANDARD ATTRIBUTES FOR TEMPJV ALLOWED 800
1 IDRC044A EQU X'044A' GUARD NAME INVALID 200
1 IDRC044B EQU X'044B' MANAGEMENT-CLASS INVALID 200
1 IDRC044C EQU X'044C' ERROR IN DEFAULT PROTECTION 300
1 *
1 * JVS RETURN CODES FROM JVDCLEX, JVRELEX, JVLNKEX AND JVCSEX
1 *
1 IDRC0450 EQU X'0450' INVALID LINKNAME.
1 IDRC0451 EQU X'0451' NO TFT EXISTING. 800
1 * IDRC0452 EQU X'0452' INVALID JOBVAR NAME. 003
1 IDRC0453 EQU X'0453' NO USER AREA. 800
1 * IDRC0454 EQU X'0454' INVALID PARAMETER. 003
1 IDRC0455 EQU X'0455' AREA SIZE TOO SMALL 800
1 IDRC0456 EQU X'0456' COMPARISON IS FALSE 801
1 IDRC0457 EQU X'0457' REMOVE ALL ENTRIES ? 103
1 IDRC0458 EQU X'0458' REQU WITHDRAWN BY USER. 103
1 *
1 * JVS RETURN CODES FROM JVERAEX
1 *
1 IDRC0460 EQU X'0460' INVALID ERASE REQUEST.
1 IDRC0461 EQU X'0461' ERASE OF SOME JOB VARIABLES IN ERROR.
1 IDRC0462 EQU X'0462' SEVER PROCESSING BY P1 MACRO CALLER REQUESTED.
1 IDRC0463 EQU X'0463' ERASE ERROR ON JV.
1 IDRC0464 EQU X'0464' ERAJV ERROR. USER HAS STILL JOB VARIABLE
1 * CATALOG ENTRIES.

```

```

1 IDRC0465 EQU X'0465' ERASE ALL JV'S ON USERID ? 950
1 IDRC0466 EQU X'0466' MONJV IS PROTECTED 800
1 IDRC0468 EQU X'0468' ERASE ALL JV'S ON PUBSET ? 950
1 IDRC0469 EQU X'0469' ERASE A SINGLE JV ? 950
1 IDRC046A EQU X'046A' ERASE REQUEST WITHDRAWN BY CALLER 950
1 *
1 * JVS RETURN CODES FROM JVGETEX
1 *
1 IDRC0470 EQU X'0470' JV VALUE CONTAINS FEWER BYTES THAN REQUESTED.
1 IDRC0471 EQU X'0471' JOB VARIABLE $SYSJV.LASTMSG NOT ACCESSIBLE. 401
1 IDRC0472 EQU X'0472' INVALID SPECIAL JOB VARIABLE.
1 IDRC0474 EQU X'0474' JV VALUE TRUNCATED.
1 IDRC0475 EQU X'0475' THE SIZE SPECIFIED FOR THE AREA IN THE GETJV
1 * MACRO IS LESS THAN 4 BYTES.
1 *
1 * JVS RETURN CODES FROM JVSETEX
1 *
1 IDRC0480 EQU X'0480' SPECIAL JOB VARIABLE MAY NOT BE SET.
1 IDRC0481 EQU X'0481' A NON-PRIVILEGED CALLER HAS REQUESTED A
1 * PRIVILEGED SETJV FUNCTION.
1 IDRC0482 EQU X'0482' THE ADDRESS OF THE JV VALUE IS MISSING.
1 IDRC0483 EQU X'0483' THE LENGTH OF THE JV VALUE EXCEEDS 256 BYTES.
1 IDRC0484 EQU X'0484' MORE THAN 1 PRIVILEGED REQUEST HAS BEEN
1 * SPECIFIED IN A SINGLE CALL.
1 IDRC0485 EQU X'0485' JVNAME2 AND VALUE ADDRESS ARE BOTH SPECIFIED.
1 * IDRC0486 EQU X'0486' JVNAME1 AND JVNAME2 ARE NOT OF SAME TYPE120
1 IDRC0487 EQU X'0487' FIRST BYTES OF MONJV ARE PROTECTED. 800
1 *
1 * JVS RETURN CODES FROM JVSTAEX
1 *
1 IDRC0490 EQU X'0490' THE AREA SIZE PARAMETER IS TOO SMALL TO CONTAIN
1 * THE CATALOG ENTRY.
1 IDRC0491 EQU X'0491' INVALID OPERAND IN SELECTION LIST
1 *
1 * JVS RETURN CODES FROM JVXXXSY, JVSV133, JVSV190, JVSYSM, JVSYSP2
1 *
1 IDRC04A0 EQU X'04A0' FUNCTIONAL UNIT "JOB VARIABLE SERVICES" (JVS)
1 * NOT SELECTED.
1 IDRC04A1 EQU X'04A1' SYNTAX ERROR IN COMMAND PARAMETER.
1 IDRC04A2 EQU X'04A2' JV HAS BEEN ERASED.
1 IDRC04A3 EQU X'04A3' ERASE ERROR ON JV.
1 IDRC04A4 EQU X'04A4' INVALID FUNCTION CODE SPECIFIED.
1 IDRC04A5 EQU X'04A5' A NON-PRIVILEGED CALLER HAS REQUESTED A
1 * PRIVILEGED ENCRYPTION OPTION.
1 IDRC04A6 EQU X'04A6' SYSTEM ERROR IN WR0UT
1 IDRC04A7 EQU X'04A7' INVALID ADDRESS IN REGISTER 1 DURING SVC133.
1 * IDRC04A8 EQU X'04A8' INVALID INTERACTIVE MODE. 007
1 IDRC04A9 EQU X'04A9' LABEL NOT FOUND. 801

```

1	IDRC04AA	EQU	X'04AA'	INVALID PL IN CURRENT MODE (GET,SET,STA)	902	
1	IDRC04AB	EQU	X'04AB'	INVALID PARAMETER IN MACRO PARAMETER LIST	950	
1	*			(INVALID FLAGS OR OLD INTERFACE USED NEW FUNCTIONS		
1	IDRC04AC	EQU	X'04AC'	INVALID CONVERSION OF TIME UTC TO LT	101	
1	IDRC04AD	EQU	X'04AD'	ERROR DURING VARIABLE PRODUCTION	120	
1	*					
1	*			JVS RETURN CODES FROM DIFFERENT JVS MODULES		
1	*					
1	IDRC04B0	EQU	X'04B0'	SIZE FIELD ZERO OR AREA ADDRESS NOT SPECIFIED.		
1	IDRC04B1	EQU	X'04B1'	PASSWORD HAS NOT BEEN PROVIDED.		
1	IDRC04B2	EQU	X'04B2'	REQUESTED JOB VARIABLE OR REQUESTED SUBSTRING		
1	*			IS EMPTY.		
1	IDRC04B3	EQU	X'04B3'	INCORRECT SYNTAX-JVNAME.		
1	IDRC04B4	EQU	X'04B4'	LINKNAME NOT PREVIOUSLY DEFINED.		
1	IDRC04B5	EQU	X'04B5'	SYSTEM ERROR: \$GETMEM ERROR.	999	
1	IDRC04B6	EQU	X'04B6'	EXPIRATION DATE ERROR.		
1	IDRC04B7	EQU	X'04B7'	SYSTEM ERROR: \$RETMEM ERROR.	999	
1	IDRC04B8	EQU	X'04B8'	ONLY READ ACCESS IS ALLOWED.		
1	IDRC04B9	EQU	X'04B9'	ILLEGAL SUBSTRING IN GET- OR SETJV.		
1	*	IDRC04BA	EQU	X'04BA'	ILLEGAL SYNTAX FOR NON-NUMERIC JV.	120
1	IDRC04BB	EQU	X'04BB'	ILLEGAL SYNTAX FOR NUMERIC JV.	750	
1	IDRC04BC	EQU	X'04BC'	NOT ALL JVS ARE ERASED OR DISPLAYED	950	
1	IDRC04BD	EQU	X'04BD'	ERROR OCCURED WHILE USING ACCESS-FUNCTION	001	
1	*			TO SYSTEM-TABLE (E.G. TCB)	001	
1	IDRC04BE	EQU	X'04BE'	USERID DOES NOT EXIST	002	
1	IDRC04BF	EQU	X'04BF'	JV PROTECTED BY ACL	004	
1	*					
1	*			JVS RETURN CODES FROM CMS		
1	*					
1	IDRC04C8	EQU	X'04C8'	JV LIMIT EXCEEDED	100	
1	IDRC04CA	EQU	X'04CA'	JV LIMIT ERROR	102	
1	*					
1	*			JVS RETURN CODES FROM JVSMJVH	800	
1	*					
1	IDRC04D0	EQU	X'04D0'	JV NOT ACCESSIBLE.	800	
1	IDRC04D1	EQU	X'04D1'	JV NOT ASSIGNED.	800	
1	IDRC04D2	EQU	X'04D2'	JV ALREADY ASSIGNED.	800	
1	IDRC04D3	EQU	X'04D3'	JV CANNOT BE CREATED.	800	
1	IDRC04D4	EQU	X'04D4'	JV INCORRECTLY SPECIFIED.	800	
1	IDRC04D5	EQU	X'04D5'	JV CATALOG ENTRY IS LOCKED.	800	
1	IDRC04D6	EQU	X'04D6'	JV NOT SUPPLIED WITH TSN.	800	
1	IDRC04D7	EQU	X'04D7'	GCF ERROR OCCURRED.	400	
1	IDRC04D8	EQU	X'04D8'	NO LINKNAME FOR MONJV.	801	
1	IDRC04D9	EQU	X'04D9'	GCF ERROR TO CONSOLE.	400	
1	*					
1	*			JVS RETURN CODES TO CMD-PROCESSOR	101	
1	*					
1	IDRC04E0	EQU	X'04E0'	CORRECT AND RETRY.	101	


```

1 IDRC04E1 EQU   X'04E1' WAIT AND RETRY.                101
1 *
1 *   JVS RETURN CODES FROM CMS
1 *
1 IDRC14A4 EQU   X'14A4' CATALOG INDEX MAX SIZE.         400
1 IDRC14A5 EQU   X'14A5' CATALOG INDEX DESTROYED.        400
1 *****
1           SPACE

```

Die aufgelisteten Returncodes entsprechen in den meisten Fällen den Fehlercodes von Systemmeldungen und lassen sich durch Hinzunahme der jeweiligen Meldungsklasse zu einem Meldungsschlüssel ergänzen. Ausführliche Informationen gewinnen Sie somit wie im vorhergehenden [Abschnitt „Kommandoebene“](#) beschrieben.

Für die folgenden Returncodes existiert keine entsprechende Systemmeldung. Sie haben folgende Bedeutung:

X'0000'	Ausführung ohne Fehler.
X'0450'	Link-Name ist ungültig.
X'0453'	Der Benutzerbereich konnte nicht versorgt werden.
X'0455'	Der Benutzerbereich ist zu klein.
x'0456'	Der Jobvariablenwert ist ungleich dem Vergleichswert.
X'0461'	Fehler beim Löschen einiger Jobvariablen. Die im ERAJV angegebene jvid passt (z.B. durch Angabe von Musterzeichen) auf mehr als ein Objekt (Katalog, Benutzerkennung und/oder JV). Nicht alle diese Objekte (u. U. auch keine) konnten korrekt bearbeitet werden.
X'0462'	Vom Benutzer wurde eine REMOVE-USER-Verarbeitung gefordert (In der ERAJV-Parameterliste war das Bit IDJESEVR gesetzt).
X'0475'	Die Länge des Bereichs beträgt weniger als 4 Bytes.
X'0481'	Ein nicht-privilegierter Benutzer hat eine privilegierte SETJV-Funktion gefordert.
X'0482'	Die Adresse für den Jobvariablen-Wert fehlt.
X'0483'	Die Länge im Satzlängengeld des Jobvariablen-Wertes überschreitet 260 Bytes oder unterschreitet 4 Bytes.
X'0484'	In einem Aufruf wurde mehr als eine privilegierte Anforderung gegeben.
X'0485'	Es wurden gleichzeitig „jvname2“ und Adresse des Wertes angegeben.
X'0490'	Die Länge des definierten Bereichs reicht nicht aus für den Katalogeintrag.
X'0491'	Angabe eines ungültigen Selektions-Operanden (JVSEL).

X'04A4'	Angabe eines ungültigen Funktionscodes (SVC 133).
X'04A5'	Der nicht-privilegierte Benutzer wünscht eine privilegierte Verschlüsselungsfunktion.
X'04A7'	Ungültige Adresse in Register 1 beim SVC 133 oder SVC 190.
X'04AA'	Der Anwender befindet sich im 31-Bit-Mode und hat eine 24-Bit-Parameterliste verwendet (GETJV, SETJV oder STAJV).
X'04AB'	Ein neuer Parameter wurde in einer alten Parameterliste verwendet oder der Anwender hat das Flag Byte auf einen unzulässigen Wert gesetzt (ERAJV, STAJV, etc.).
X'04B0'	Länge des Feldes beträgt null oder die Adresse des Bereichs ist nicht definiert.
X'04B9'	Angabe eines unzulässigen Wertebereichs.

Folgende Returncodes werden bei fehlerhafter Initialisierung des Standardheaders (PARMOD=31) zurückgegeben:

X'0001FFFF':	Unit- und/oder Funktionsnummer im Header wird nicht unterstützt.
X'0003FFFF':	Versionsnummer im Header bezeichnet eine Schnittstellenversion, die nicht unterstützt wird.
X'0004FFFF':	Die Parameterliste ist nicht auf Wortgrenze ausgerichtet.

7 Der privilegierte Benutzer

Der privilegierte Benutzer ist der Systembetreuer, der üblicherweise unter der Benutzerkennung TSOS arbeitet. Mit Einsatz des Softwareproduktes SECOS werden die Funktionen der Systembetreuung auf verschiedene Benutzerkennungen aufgeteilt. Soweit keine andere Angabe gemacht wird, ist in den nachfolgenden Abschnitten mit dem privilegierten Benutzer immer die Systembetreuung unter der Benutzerkennung TSOS gemeint.

7.1 Installation

Das Softwareprodukt Jobvariablen (JV) V15.0 kann in den aktuell freigegebenen Versionen von BS2000/OSD-BC eingesetzt werden. Es handelt sich um ein kostenpflichtiges Softwareprodukt, das nicht im Grundausbau des BS2000 enthalten ist. Jobvariablen V15.0 wird als nachladbares Subsystem (im Sinne von DSSM) ausgeliefert.

Folgende Bestandteile werden mit JV V15.0 ausgeliefert:

Dateiname	Inhalt
SYSFGM.JV.150.D	Freigabemitteilung deutsch
SYSFGM.JV.150.E	Freigabemitteilung englisch
SYSRME.JV.150.D	Readme-Datei deutsch
SYSRME.JV.150.E	Readme-Datei englisch
SYSLNK.JV.150	Bindelademodul (LLM) für S-Server
SPMLNK.JV.150	Bindelademodul (LLM) für SX-Server
SKMLNK.JV.150	Bindelademodul (LLM) für SQ-Server
SYSLIB.JV.150	Benutzer-Makrobibliothek
SIPLIB.JV.150	TPR-Makrobibliothek
SYSMES.JV.150	Meldungsdatei (siehe auch Kapitel „Meldungen“ auf Seite 227)
SYSRMS.JV.150	Liefermenge für RMS
SYSSDF.JV.150	Syntaxdatei (siehe auch „Syntaxdateien“ auf Seite 236)
SYSSSC.JV.150	Subsystemdeklaration (siehe auch „Subsystemdeklarationen“ auf Seite 236)

Die Installation muss mit IMON erfolgen.

Subsystemdeklarationen

Das Subsystem JV wird von DSSM geladen und steht zum Zeitpunkt „system ready“ zur Verfügung. Das Subsystem JV muss dazu im Subsystemkatalog deklariert sein. Die benötigten Subsystemdeklarationen sind in der Datei SYSSSC.JV.150 enthalten. Die Erstellung des Subsystemkatalogs ist in dem Handbuch „Verwaltung von Subsystemen“ [7] beschrieben.

Bei seiner Initialisierung öffnet das Subsystem JV die Systemdatei \$TSOS.SYSCAT.JV auf dem Home-Pubset für interne Nutzung. Falls die Datei noch nicht existiert, wird sie vorher neu angelegt. Die Datei wird erst mit Beendigung des Subsystems JV geschlossen.

Syntaxdateien

Mit JV wird die Syntaxdatei SYSSDF.JV.150 ausgeliefert, die von IMON als Subsystem-Syntaxdatei in die SDF-Parameterdatei eingetragen wird.

Für selbsterstellte Gruppen- und Benutzer-Syntaxdateien gilt bei Versionswechsel:

- Enthalten diese JV-Kommandos, die mit SDF-A modifiziert wurden, müssen sie neu erstellt werden.
- Gruppen-Syntaxdateien, die mit HIERARCHY=NO eingesetzt werden und JV-Kommandos enthalten, müssen ebenfalls neu erstellt werden.

Die Verwaltung von System- und Gruppensyntaxdateien und das Dienstprogramm SDF-I sind im Handbuch „SDF-Schnittstelle“ [2] beschrieben.

7.2 Privilegierte Zugriffsrechte

Der privilegierte Benutzer wird hinsichtlich der Jobvariablen anderer Benutzer wie der Eigentümer behandelt (kann auch eingeschränkt werden, siehe „[Eingeschränkte TSOS-Miteigentümerschaft](#)“ auf Seite 24). Dies gilt auch für den Operator, der an der Konsole die Kommandos SHOW-JV, MODIFY-JV und SHOW-CJC-STATUS absetzen kann. Bei Schutz mit BASIC-ACL gelten die für OWNER vereinbarten Zugriffsrechte.

Der privilegierte Benutzer kann in allen Kommandos bzw. Makroaufrufen, die im JV-Namen Musterzeichen zulassen, auch innerhalb der Benutzerkennung Musterzeichen verwenden (z.B. Kommando SHOW-JV-ATTRIBUTES bzw. STAJV-Makro).

Bei Ausgabe des JV-Eintrages (SHOW-JV-ATTRIBUTES bzw. STAJV-Makro) werden dem privilegierten Benutzer vereinbarte Kennwörter mit ausgegeben. Die Änderung eines JV-Eintrages (MODIFY-JV-ATTRIBUTES bzw. CATJV-Makro) kann er ohne Angabe von Kennwörtern durchführen. Beim Löschen von Jobvariablen (DELETE-JV bzw. ERAJV-Makro) hat er die Möglichkeit, den Kennwortschutz zu ignorieren (Operand IGNORE-PROTECTION).

Er ist außerdem berechtigt, Jobvariablen unter jeder Benutzerkennung einzurichten (CREATE-JV bzw. CATJV-Makro). Er hat auch Zugriff auf alle temporären Jobvariablen im System.

Die Systembetreuung kann Systemverwalterfunktionen auf den Operator übertragen (siehe Handbuch „Einführung in die Systembetreuung“ [3]). Das bedeutet, dass der Operator außer SHOW-JV und MODIFY-JV noch weitere Kommandos (z.B. CREATE-JV, DELETE-JV) mit denselben Privilegien an der Konsole geben darf.

7.3 Systemdatei SYSCAT.JV

Bei seiner Initialisierung öffnet das Subsystem JV die Systemdatei \$TSOS.SYSCAT.JV auf dem Home-Pubset für interne Nutzung. Falls die Datei noch nicht existiert, wird sie vorher neu angelegt. Die Datei wird erst mit Beendigung des Subsystems JV geschlossen.

7.4 Überwachung bei der Pubset-Verwaltung

Bei der Verwaltung von Pubsets kann die Systembetreuung folgende Funktionen von Jobvariablen überwachen lassen:

- „Importieren eines Pubsets“
- „Exportieren eines Pubsets“
- „Exportieren eines Pubsets erzwingen“

Beim Importieren eines Shared-Pubsets wird eine Jobvariable eingerichtet, die die Verfügbarkeit dieses Pubsets anzeigt.

Folgende Zustandswerte werden beim Importieren und Exportieren in einer überwachenden Jobvariable gesetzt:

Wert	Funktion	Bedeutung/Anlass der Wertzuweisung
\$A	Export	Exportieren fehlerhaft oder mit CANCEL-PUBSET-EXPORT abgebrochen
	Import	Import fehlerhaft abgebrochen
\$E	Export	Exportauftrag erfolgreich gestartet
\$I	Import	Importauftrag erfolgreich eingeleitet
\$R	Import	Pubset erfolgreich importiert
\$T	Export	Exportieren eines Pubsets erfolgreich abgeschlossen
\$W	Import	Bei Import eines Shared-Pubsets wartet der Rechner auf die Bestätigung des Master-Rechners

Tabelle 32: Zustandswerte beim Importieren/Exportieren einer überwachenden Jobvariable

Beim Shared-Pubset-Betrieb innerhalb eines Mehrrechnerverbundes ist auf jedem Home-Pubset eines Sharers eine shared-pubset-spezifische Jobvariable eingerichtet, welche die Vorgänge im Verbund überwacht.

Wert	Bedeutung/Anlass der Wertzuweisung
\$R	Shared-Pubset verfügbar. Master-Wechsel erfolgreich durchgeführt.
\$T	Shared-Pubset wird nicht mehr überwacht; er wird oder ist bereits exportiert.
\$A	Master-Wechsel fehlerhaft abgebrochen, z.B. bei Ausfall des Masters und einer nicht erfolgreichen Bestimmung eines neuen Masters aus der Reihe der Slave-Rechner.
\$C	Wegen Ausfall bzw. Abschalten des Masters wird ein Master-Wechsel auf dem gemeinschaftlichen Datenträger durchgeführt.

Tabelle 33: Überwachende Jobvariable im Shared-Pubset-Betrieb

In [Tabelle 34](#) auf [Seite 239](#) wird ein Überblick über die Kommandos gegeben, in denen die Systembetreuung die Verwaltung von Pubsets mit Jobvariablen überwachen kann. Mehr Informationen zur Verwaltung von Pubsets sind in den Handbüchern „Einführung in die Systembetreuung“ [3] und „HIPLEX MSCF“ [8] enthalten.

Hinweise zur Verwendung einer MONJV

- Die Jobvariable darf nicht schreibgeschützt sein.
- Ist die Jobvariable **nicht** katalogisiert, wird die Meldung DMS0383 ausgegeben und der Import-/Export-Auftrag wird ohne überwachende Jobvariable durchgeführt.
- Ist die Jobvariable mit einem Kennwort geschützt, kann die Erklärung zur überwachenden Jobvariablen nur erfolgen, wenn der Auftraggeber das Kennwort bereits in die Kennwort-Tabelle eingetragen hat (ADD-PASSWORD) oder im Operanden JV-PASSWORD der nachfolgenden Kommandos angibt.

Kommando	Operanden, die für die Überwachung der Pubset-Verwaltung relevant sind
EXPORT-PUBSET exportiert einen zuvor importierten Pubset	MONJV=*NONE/<filename 1..54 without-gen> definiert die überwachende JV. JV-PASSWORD=NONE/<c-string 1..4>/<x-string 1..8>/<integer -2147483648..21484836479> definiert ein Kennwort für die überwachende JV.
<p><i>Hinweis zu EXPORT-PUBSET:</i></p> <p>Das Kommando erzeugt einen Auftrag, der das Exportieren durchführt. Der zu exportierende Pubset wird „unerreichbar“ gesetzt. Der Auftrag wartet, bis alle belegenden Aufträge ihre Belegung freigegeben haben (geöffnete Dateien oder Reservierung von Dateien mit SECURE-RESOURCE-ALLOCATION). Der Export-Auftrag informiert über die Anzahl der Aufträge, die den Pubset noch belegen. Die Auftragsnummern können mit dem Kommando SHOW-PUBSET-PARAMETER abgefragt werden. Nach Beendigung des Wartezustands wird der Benutzerkatalog geschlossen und alle Betriebsmittel werden freigegeben. Die Meldungen des Auftrags werden an der Bedienstation ausgegeben. Der Wartezustand der Export-Verarbeitung kann mit dem Operanden TERMINATE-JOBS=YES beendet werden. Dabei werden alle noch belegenden Aufträge ordnungsgemäß beendet. Lässt sich der Wartezustand nicht beheben, kann dies mit dem Kommando FORCE-PUBSET-EXPORT erzwungen werden.</p> <p>Die Änderung der Verfügbarkeit wird an alle aktiven Rechner eines Rechnernetzes gemeldet. Der Export-Auftrag kann durch eine Jobvariable (siehe MONJV) überwacht werden. Auf die jeweiligen Verarbeitungszustände kann in einer bedingungsabhängigen Auftragssteuerung reagiert werden.</p>	

Tabelle 34: Kommandos der Pubset-Verwaltung mit MONJV-Unterstützung (Teil 1 von 2)

Kommando	Operanden, die für die Überwachung der Pubset-Verwaltung relevant sind
FORCE-PUBSET-EXPORT erzwingt das Exportieren eines Pubsets.	MONJV=* <u>NONE</u> /<filename 1..54 without-gen> definiert die überwachende JV. JV-PASSWORD= <u>NONE</u> /<c-string 1..4>/<x-string 1..8>/<integer -2147483648..21484836479> definiert ein Kennwort für die überwachende JV.
IMPORT-PUBSET importiert einen Pubset. <i>Hinweis zu IMPORT-PUBSET:</i> Der Import-Auftrag kann durch eine Jobvariable (siehe MONJV) überwacht werden. Auf die jeweiligen Verarbeitungszustände kann in einer bedingungsabhängigen Auftragssteuerung reagiert werden. Wurde ein mehrbenutzbarer Pubset importiert, wird zusätzlich die Verfügbarkeit in einer eigenen Jobvariablen protokolliert. Diese Jobvariable wird beim Importieren auf dem Home-Pubset des importierenden Rechners neu eingerichtet bzw. neu gesetzt: Standardname ist :<cat-id home>:\$TSOS.SYS.PVS.<cat-id shared-pubset>.MASTER.CONTROL Mögliche Zustandswerte siehe Tabelle 33 auf Seite 238 .	MONJV=* <u>NONE</u> /<filename 1..54 without-gen> definiert die überwachende JV. JV-PASSWORD= <u>NONE</u> /<c-string 1..4>/<x-string 1..8>/<integer -2147483648..21484836479> definiert ein Kennwort für die überwachende JV.

Tabelle 34: Kommandos der Pubset-Verwaltung mit MONJV-Unterstützung (Teil 2 von 2)

7.5 Jobvariablen als Objekt der Systemüberwachung

Bei Einsatz des Softwareproduktes SECOS gehören Jobvariablen zu den Objekten, die mit der Funktionseinheit SAT überwacht werden können (Objekt JOB VARIABLES).

Mit SAT können Zugriffe auf das Objekt Jobvariable protokolliert werden. Protokollierung eines bestimmten Zugriffs (z.B. Lesen einer Jobvariablen) erfolgt, wenn der Sicherheitsbeauftragte das Objekt Jobvariable zur Überwachung zugelassen hat. Daneben kann er die Protokollierung eines bestimmten Zugriffs vom Resultat abhängig machen: erfolgreich (SUCC) oder nicht erfolgreich (FAIL). Protokolliert wird für Jobvariablen das Resultat (SUCC oder FAIL) und der vollqualifizierte Jobvariablen-Name bzw. der teilqualifizierte Jobvariablen-Name oder eine Musterzeichenfolge. Die Rückkehrinformation von JV kann zur Protokollierung zusätzlich zugelassen werden. Sie ist mit *NONE vorbesetzt und bleibt beim Resultat „erfolgreicher Zugriff“ (SUCC) unverändert.

Allgemeine Fehler beim Zugriff werden nicht protokolliert (Syntaxfehler, Fehler in der Parameterliste, Jobvariablen-Name fehlerhaft).

Die Systemüberwachung mit SAT ist umfassend im Handbuch „SECOS“ [9] beschrieben.

Für das Objekt JOB VARIABLES können die nachfolgenden Ereignisse ausgewählt werden. Der Kurzname für das Ereignis ist jeweils vorangestellt. Anschließend folgen die Kommandos und Makroaufrufe, die das Ereignis auslösen können:

- JVC** Jobvariableneintrag (und damit Schutzattribute) erstellen:
Kommando CREATE-JV oder Makroaufruf CATJV mit STATE=NEW; auch Kommando SET-JV-LINK bzw. Makroaufruf DCLJV, wenn eine nicht existierende Jobvariable eingerichtet wird.
- JVM** Schutzattribute einer Jobvariablen ändern:
Kommando MODIFY-JV-ATTRIBUTES oder Makroaufruf CATJV mit STATE=UPDATE.
- JVR** Jobvariable umbenennen:
Im Kommando MODIFY-JV-ATTRIBUTES wird NEW-NAME angegeben oder im Makroaufruf CATJV mit STATE=UPDATE wird ein zweiter Jobvariablen-Name vergeben. Werden dabei gleichzeitig Schutzattribute geändert, so wird ein zweiter SAT-Record für das Ereignis JVM geschrieben.
- JVA** Jobvariable umbenennen beim Rekonstruieren mit dem Dienstprogramm ARCHIVE
- JVD** Jobvariableneintrag (und damit Schutzattribute) löschen:
Kommando DELETE-JV und Makroaufruf ERAJV. Bei Verwendung eines teilqualifizierten Jobvariablen-Namens bzw. von Musterzeichen wird das Auflisten der betroffenen Jobvariablen nicht protokolliert. Beim Löschen wird für eine betroffene Jobvariable ein SAT-Record geschrieben.
Wird nur der Wert der Jobvariablen gelöscht (OPTION=DATA bzw. DATA=YES), so ist dies nur ein schreibender Zugriff (siehe Ereignis JVS).

- JVQ Informationen über Jobvariablen (auch Schutzattribute) auflisten:
Kommando SHOW-JV-ATTRIBUTES oder Makroaufruf STAJV
Ein angegebener teilqualifizierter Jobvariablen-Name bzw. Musterzeichenfolge werden ebenfalls protokolliert. Für jede betroffene Jobvariable wird ein SAT-Record geschrieben.
- JVG Lesen des Jobvariablenwertes:
Kommando SHOW-JV oder Makro GETJV, bei Verwendung in bedingten Ausdrücken und bei Jobvariablen-Ersetzung.
- JVS Schreiben des Jobvariablenwertes:
Kommandos MODIFY-JV, MODIFY-JV-CONDITIONALLY, MODIFY-MONJV oder Makroaufrufe SETJV, CSWJV, TIMJV.
Wird der zu setzende Wert aus einer Jobvariablen übernommen, so wird für den lesenden Zugriff auf diese Jobvariable ein weiterer SAT-Record geschrieben (siehe Ereignis JVG).

Die Protokollierung eines bestimmten Zugriffs auf das Objekt JOB VARIABLES kann von den nachfolgenden Informationen abhängig gemacht werden:

- JVNAME Voll- oder teilqualifizierter Jobvariablen-Name
- JVPATRN Musterzeichenfolge
- NEWJV Neuer Jobvariablen-Name
- JVSRC Rückkehrinformation

Nachfolgende Tabelle zeigt, welche Informationen unbedingt (**Mandatory**), nur wahlweise (**Optional**) oder nicht vorhanden („-“) sein müssen, damit bestimmte Ereignisse für das Objekt JOB VARIABLES protokolliert werden können.

Information	Ereignis							
	JVA	JVC	JVD	JVG	JVM	JVQ	JVR	JVS
JVNAME	M	M	M	M	M	O ¹	M	M
JVPATRN	-	-	-	-	-	O ¹⁾	-	-
JVSRC	O	O	O	O	O	O	O	O
NEWJV	M	-	-	-	-	-	M	-

Tabelle 35: Abhängigkeit der Protokollierung eines Zugriffs auf das Objekt JOB VARIABLES von Informationen

¹ Einer der beiden ist notwendig.

Abkürzungen

ACL	Access Control List
BASIC-ACL	Basic Access Control List
CJC	Conditional Job Control
CMS	Catalog Management System
DSECT	Dummy Section
DSSM	Dynamic Subsystem Management
GUARDCOO	GUARD Co-Owner Protection (Miteigentümerschutz)
GUARDS	Generally Usable Access Control Administration System (Zugriffsüberwachung)
ISP	Interactive String Processor (Standard -Kommandoentschlüssler in BS2000, der vor SDF eingesetzt wurde)
JMS	Job Management System
JPM	Job Program Management
JV	Jobvariable
JVS	Job Variable System
MONJV	Überwachende (monitoring) Jobvariable
MRSCAT	Katalog der an einem Rechner bekannten Pubsets. Im Mehrrechnerverbund sind zusätzlich die am Verbund beteiligten Rechner (repräsentiert durch ihre catid's) eingetragen.
MSCF	Multi System Control Facility

Abkürzungen

MF	Macro Format
RS	Rücksprungschalter
SAT	Security Audit Trail
SDF	System Dialog Facility (Standard-Kommandoentschlüssler ab BS2000 V10.0)
SECOS	Security Control System
SI	Sekundärindikator
TIAM	Terminal Interactive Access Method (BS2000-Zugriffsmethode mit der es möglich ist, so im Dialog zu arbeiten als stünde das System allein einem Anwender zur Verfügung)
TPR	Task privileged
TSOSCAT	Systemkatalog
TSN	Task Sequence Number (Auftragsfolgennummer)
TU	Task unprivileged
UTC	Universal Time Coordinate (entspricht der Greenwich-Zeit)
VSN	Volume Serial Number (Archivnummer)

Literatur

Die Handbücher sind online unter <http://manuals.ts.fujitsu.com> zu finden oder in gedruckter Form gegen gesondertes Entgelt unter <http://manualshop.ts.fujitsu.com> zu bestellen.

- [1] **BS2000/OSD-BC
Kommandos**
Benutzerhandbuch
- [2] **SDF (BS2000/OSD)
Dialogschnittstelle SDF**
Benutzerhandbuch
- [3] **BS2000/OSD-BC
Einführung in die Systembetreuung**
Benutzerhandbuch
- [4] **BS2000/OSD-BC
Makroaufrufe an den Ablaufteil**
Benutzerhandbuch
- [5] **BS2000/OSD-BC
DVS Makros**
Benutzerhandbuch
- [6] **SDF-P (BS2000/OSD)
Programmieren in der Kommandosprache**
Benutzerhandbuch
- [7] **DSSM / SSCM (BS2000/OSD)
Verwaltung von Subsystemen**
Benutzerhandbuch
- [8] **HIPLEX MSCF (BS2000/OSD)
BS2000-Rechner im Verbund**
Benutzerhandbuch

- [9] **SECOS** (BS2000/OSD)
Security Control System - Beweissicherung
Benutzerhandbuch
- [10] **SECOS** (BS2000/OSD)
Security Control System - Zugangs- und Zugriffskontrolle
Benutzerhandbuch
- [11] **ARCHIVE** (BS2000/OSD)
Benutzerhandbuch
- [12] **HSMS / HSMS-SV** (BS2000/OSD)
Hierarchisches Speicher Management System
Benutzerhandbuch
- [13] **SPOOL** (BS2000/OSD)
Benutzerhandbuch
- [14] **BLSSERV** (BS2000/OSD)
Bindelader-Starter
Benutzerhandbuch
- [15] **openFT** (BS2000/OSD)
Enterprise File Transfer in der offenen Welt
Benutzerhandbuch
- [16] **SPACEOPT** (BS2000/OSD)
Optimierung und Reorganisation von Platten
Benutzerhandbuch
- [17] **VM2000** (BS2000/OSD)
Virtuelles Maschinensystem
Benutzerhandbuch

Stichwörter

A

ACCESS (Schutzmerkmal) 37
ADD-CJC-ACTION (Kommando) 65, 67, 83
ADD-DEFAULT-PROTECTION-ATTR
(Kommando) 36
ADD-DEFAULT-PROTECTION-RULE
(Kommando) 36
AIDSYS 184
Attribut-Guard 36
Auftragsüberwachung 55, 77
 ENTER-JOB (Kommando) 77
 ENTER-PROCEDURE (Kommando) 77
 Jobvariablen 53
 Kettungsname 64
 Kommandoübersicht 77, 80
 MODIFY-JOB-OPTIONS (Kommando) 77,
 80
 PRINT-DOCUMENT (Kommando) 77
 Selektionskriterium 145
 SET-LOGON-PARAMETERS
 (Kommando) 77
 TRANSFER-FILE (Kommando) 77
Auftragsverarbeitung
 SET-SYSLST-READ-MARK (Kommando) 80
Auftragsverwaltung
 CANCEL-FILE-TRANSFER (Kommando) 80
 CANCEL-JOB (Kommando) 80
 CANCEL-PRINT-JOB (Kommando) 80
 CHANGE-TASK-PRIORITY (Kommando) 80
 FORCE-JOB-CANCEL (Kommando) 80
 HOLD-JOB (Kommando) 80
 MODIFY-JOB (Kommando) 80
 RESUME-JOB (Kommando) 80
 SHOW-FILE-TRANSFER (Kommando) 80
 SHOW-JOB-STATUS (Kommando) 80

SHOW-RESOURCE-ALLOCATION

(Kommando) 80

Ausdrücke

bedingte 83
relationale 83

B

BASIC-ACL (Schutzmerkmal) 37
 Schnittstellenübersicht 31
 Selektionskriterium 142
BBS
 Operationsliste 184
 Programmtabelle 184
bedingte Ausdrücke 83
 Großschreibung 84
 Kleinschreibung 84
Bedingungen 65
bedingungsabhängige Auftragssteuerung 7, 65,
83
 Kommandoübersicht 83
Beispiele
 zu JV-Kommandos 191
 zu JV-Makros 208, 219
 zur Auftragsüberwachung 200, 201, 202
 zur Programmüberwachung 198
 zur Verwaltung von Jobvariablen 191
Benennen von überwachenden JV
 Kettungsnamen 64
Benutzer-JV 13, 131
 Benennung 16
 Werte 42
Benutzerkennung (Benennung der JV) 16
Benutzerklasse
 GROUP 30
 OTHERS 30

Benutzerklasse (Forts.)

OWNER 30

Zugriffsrechte 31

Benutzerprogramm 131

C

CANCEL-FILE-TRANSFER (Kommando) 80

CANCEL-JOB (Kommando) 80

CANCEL-PRINT-JOB (Kommando) 80

CANCEL-PUBSET-EXPORT (Kommando) 239

CATJV (Makro) 92, 237

CHANGE-TASK-PRIORITY (Kommando) 80

CJC-Schutz 40

Selektionskriterium 146

Conditional Job Control 65

Contingency-Prozess 159

COPJV (Makro) 104

CREATE-JV (Kommando) 15, 30, 36, 237

CSWJV (Makro) 109

D

DCLJV (Makro) 114

DELETE-JV (Kommando) 15, 237

DISEI (Makro) 68, 118

DONEVT (Makro) 65, 68

E

Eigentümer der JV 23

Eingabe aus Jobvariablen 70

ENAEI (Makro) 158

END-CJC-ACTION (Kommando) 68

ENTER-JOB (Kommando) 77

ENTER-PROCEDURE (Kommando) 77

ERAJV (Makro) 123, 237

ereignisabhängige Steuerung von Aufträgen 65

Ereigniskennung 156

Ereigniskurzbezeichnung 158

Ereignisse 65

Synchronisierung 66

Ersetzen von Jobvariablen 70

Erstellungsdatum einer JV

Selektionskriterium 140

EXDATE (Schutzfrist für JV) 27

Expiration Date (Freigabedatum der Datei) 141

EXPORT-PUBSET (Kommando) 239

Exportieren eines Pubsets überwachen 238

F

FORCE-JOB-CANCEL (Kommando) 80

FORCE-PUBSET-EXPORT (Kommando) 239

Freigabedatum der JV (Expiration Date),
Selektionskriterium 141

G

GETJV (Makro) 131

Größe des reservierten Speicherplatzes 147

GROUP (Benutzerklasse) 30

GUARDS (Schutzmerkmal) 37

GUARDS Schutzmerkmal)

Selektionskriterium 145

H

HOLD-JOB (Kommando) 80

HSMS-Management-Klasse

Selektionskriterium 145

I

IMPORT-PUBSET (Kommando) 239

Importieren eines Pubsets überwachen 238

Informationsaustausch zwischen Aufträgen 53

J

Jobkette 53, 65

Jobnetz 53, 65

Jobrestart 53

Jobtrace 53

Jobvariable

Aufbau 15

Dauer der Schutzfrist 99

Definition 7

Elemente im Systemteil einer auftragsüberwa-
chenden JV versorgen 186

Jobvariablen nach Selektionskriterien
auswählen 136

JV-LINK-Eintrag ausgeben 151

JV-LINK-Eintrag löschen 163

katalogisieren 92

Kettungsname definieren 114

- Jobvariable (Forts.)
 - kopieren 104
 - Lese- und Schreibzugriff 95
 - Lesekennwort 98
 - löschen 123
 - Merkmale ausgeben 172
 - Programm/Prozedurabschnitt beenden 182
 - prüfen und setzen 109
 - Schreibkennwort 99
 - Verwaltungsteil 15
 - Wert ausgeben 131
 - Wert setzen 166
 - Zugriffsrechte der anderen Benutzer 97
 - Zugriffsrechte der Gruppe 96
 - Zugriffsrechte des Eigentümers 96
- Jobvariablen-Einsatz
 - Auftragsüberwachung 75
 - bedingungsabhängige Auftragssteuerung 75
 - Programmüberwachung 75
 - Verwaltung 75
- Jobvariablen-Eintrag löschen 23
- Jobvariablen-Ereignis 118
- Jobvariablen-Ersetzung 70
 - Builtin-Funktion 71
 - Einschränkungen 70
 - im Batchbetrieb 70
 - im ungeführten Dialog 70
 - in Prozeduren 70
 - in S-Prozeduren 71
 - S-Variablen 71
- Jobvariablen-Kettungsname 114, 151
- Jobvariablen-Name (Benennung der JV) 16
- Jobvariablenwert
 - Länge (Selektionskriterium) 147
- JV-LINK-Eintrag 163
- JV-LINK-Tabelle 20, 151, 163
- JVSEL (Makro) 136

- K**
 - Katalogisieren einer Jobvariable 92
 - Katalogkennung (Benennung der JV) 16
 - Kennwort für JV, Zeitstrafe 28
 - Kennwortkombinationen 28
 - Kennwortschutz für JV 28, 62
 - Selektionskriterium 139
 - Kettungsname 20
 - bei Auftragsüberwachung 64
 - bei JV-Ersetzung 21
 - für JV definieren 114
 - in bedingten Ausdrücken 21
 - Programmüberwachung 64
 - Verwendungsmöglichkeit (Tabelle) 21
 - Kommando
 - ADD-DEFAULT-PROTECTION-ATTR 36
 - ADD-DEFAULT-PROTECTION-RULE 36
 - CREATE-JV 36
 - MODIFY-DEFAULT-PROTECTION-ATTR 36
 - MODIFY-DEFAULT-PROTECTION-RULE 36
 - Kopieren
 - Jobvariable 104

- L**
 - Lesekennwort für JV 28
 - LNKJV (Makro) 151
 - logische Operatoren 65, 84
 - LOGON 77

- M**
 - Makros 87
 - CATJV 92
 - CSWJV 109
 - DCLJV 114
 - ERAJV 123
 - GETJV 131
 - JVSEL 136
 - LNKJV 151
 - Metazeichen 90
 - Operand MACID 89
 - Operand MF 88
 - Operand PARAM 89
 - Operand PREFIX 89
 - RELJV 163
 - SETJV 166
 - STAJV 172
 - TERM 182
 - TIMJV 186
 - Übersicht 87

- Mehrbenutzbarkeit [27](#)
 - Selektionskriterium [139](#)
- Meldungen
 - Kommandoebene [227](#)
 - Makroebene [229](#)
- Metazeichen (Makros) [90](#)
- Mit-Eigentümerschaft [23](#)
- MODIFY-DEFAULT-PROTECTION-ATTR (Kommando) [36](#)
- MODIFY-DEFAULT-PROTECTION-RULE (Kommando) [36](#)
- MODIFY-JOB (Kommando) [80](#)
- MODIFY-JOB-OPTIONS (Kommando) [77, 80](#)
- MODIFY-JV (Kommando) [44, 237](#)
- MODIFY-JV-ATTRIBUTES (Kommando) [30, 54, 237](#)
- MODIFY-JV-CONDITIONALLY (Kommando) [44, 66](#)
- MONJV-Schutz [40](#)
 - Selektionskriterium [145](#)
- Musterzeichen
 - bei temporären und permanenten JVs [18](#)
 - in JV-Kommandos/-Makros [18](#)
- Musterzeichen(Wildcards) [44](#)
- O**
- ONEVT (Makro) [65, 68](#)
- Operatoren, logische [84](#)
- OTHERS (Benutzerklasse) [30](#)
- OWNER (Benutzerklasse) [30](#)
- P**
- Parameterliste [89](#)
- permanente JV [13, 15, 16](#)
- Pfadname
 - Aufbau (Benennung der JV) [16](#)
 - Gesamtlänge (Benennung der JV) [16](#)
- POSSIG (Makro) [156](#)
- Post Code [158](#)
- PRINT-DOCUMENT (Kommando) [77](#)
- Programmüberwachung [59, 82, 199](#)
 - Kettungsname [64](#)
 - Kommandoübersicht [82](#)
- Pseudo-Benutzerkennung SYSJV [44](#)
- Pubsetverwaltung
 - CANCEL-PUBSET-EXPORT (Kommando) [239](#)
 - EXPORT-PUBSET (Kommando) [239](#)
 - FORCE-PUBSET-EXPORT (Kommando) [239](#)
 - IMPORT-PUBSET (Kommando) [239](#)
 - Kommandos zur Überwachung [239](#)
 - Überwachung [238](#)
 - Zustandswerte einer MONJV [238](#)
- R**
- READ-PASS-WORD (Schutzmerkmal) [37](#)
- Readme-Datei [10](#)
- Regel [36](#)
- Regelbehälter [36](#)
- Register R1 [89](#)
- relationale Ausdrücke [83](#)
- RELJV (Makro) [163](#)
- REMOVE-CJC-ACTION (Kommando) [65, 68](#)
- REMOVE-JV-LINK (Kommando) [21](#)
- RESUME-JOB (Kommando) [80](#)
- S**
- Schreibkennwort für JV [28](#)
- Schutz der JV
 - bei Auftragsüberwachung [40](#)
 - bei bedingungsabhängiger Auftragssteuerung [40](#)
- Schutz von überwachenden Jobvariablen [62](#)
- Schutzfrist für JV [27](#)
- Schutzmerkmal der JV
 - ändern [23](#)
 - vereinbaren über Default-Protection [36](#)
- Schutzmerkmalen [36](#)
- SET-JV-LINK (Kommando) [20, 70](#)
- SET-LOGON-PARAMETERS (Kommando) [77](#)
- SET-SYSLST-READ-MARK (Kommando) [80](#)
- SETJV (Makro) [166](#)
- Shared-Pubset-Betrieb
 - überwachen [238](#)
 - Zustandswerte einer MONJV [238](#)
- SHOW-CJC-STATUS (Kommando) [66, 237](#)
- SHOW-FILE-TRANSFER (Kommando) [80](#)

SHOW-JOB-STATUS (Kommando) 80
 SHOW-JV (Kommando) 44, 237
 SHOW-JV-ATTRIBUTES (Kommando) 237
 SHOW-JV-LINK (Kommando) 21
 SHOW-RESOURCE-ALLOCATION
 (Kommando) 80
 SKIP-COMMANDS (Kommando) 44, 65, 66, 83
 SMONJVJ (Standard-Kettungsname) 20, 64, 80
 SMONJVP (Standard-Kettungsname) 20, 64, 82
 SOLSIG (Makro) 68, 159
 Sonder-Jobvariablen 13, 15, 44, 131
 Ausgabe 44
 Katalogeintrag 44
 System Exit 033 51
 Zugriff 44
 Speicherauszug ausgeben 182, 183
 STAJV (Makro) 172, 237
 Standard-Kettungsname
 SMONJVJ 20, 64, 80
 SMONJVP 20, 64, 82
 Standard-Zugriffskontrolle 26
 Standardheader 89
 STXIT-Routine 184
 Synchronisierung v. Ereignissen 66
 SYSCAT.JV (Systemdatei) 236, 237
 SYSJV (Pseudo-Benutzerkennung) 44
 System Exit 033 (für Sonder-Jobvariablen) 51
 Systemdatei SYSCAT.JV 236, 237
 Systeminformation, Abfrage von 44
 Systemüberwachung
 Einsatz von SECOS 241
 Ereignisse für das Objekt JOB
 VARIABLES 241
 JV als Objekt der Überwachung 241

T
 Tabelle der JV-LINK-Einträge 20
 temporäre JV 13, 15, 19, 237
 Zugriff auf 22
 TERM (Makro) 182
 TIMJV (Makro) 186
 TRANSFER-FILE (Kommando) 77
 TU-Eventing 158

U

Überwachen
 Exportieren eines Pubsets 238
 Importieren eines Pubsets 238
 Pubsetverwaltung 238
 Repeatjobs 63
 Shared-Pubset-Betrieb 238
 überwachende JV
 Benennung 64
 Identifizierung eines Auftrags 61
 Werte 54
 USER-ACCESS (Schutzmerkmal) 37

V

Vererbung der Zugriffsberechtigung 62
 Vergleichsoperatoren in bedingten
 Ausdrücken 84
 Verwaltung von Jobvariablen, Kommando-/Makro-
 Übersicht 13

W

WAIT-EVENT (Kommando) 65, 66, 83
 Werte für Benutzer-JV 42
 Werte für überwachende JV 54
 bei Batchaufträgen 57
 bei Dialogaufträgen 57
 bei Programmüberwachung 59
 beim Exportieren eines Pubsets 58
 beim Importieren eines Pubsets 58
 beim Shared-Pubset-Betrieb 59
 Wildcards in JV-Kommandos/-Makros 18
 Wortgrenze 89
 WRITE-PASSWORD (Schutzmerkmal) 37

Z

Zeitstempel 55, 186
 Zeitstrafe bei Kennwortschutz 28
 Zugriffsart 26
 Selektionskriterium 139
 Zugriffsart auf eine JV 26
 Zugriffsberechtigung 23, 62
 Zugriffskontrolle 31
 Kennwort 28
 Selektionskriterium 146

Zugriffskontrollliste, einfache 31
Zugriffsrechte 31
 Kombinationen (Tabelle) 28, 31
Zugriffsverlust auf eine JV 41
Zustandswerte einer MONJV
 Pubsetverwaltung 238
 Shared-Pubset-Betrieb 238