

IBM Workload Scheduler
Version 9 Release 4

*Terminierung mit dem Agenten für
z/OS*

IBM

IBM Workload Scheduler
Version 9 Release 4

*Terminierung mit dem Agenten für
z/OS*

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 209 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9, Release 4, Modifikationsstufe 0 von IBM Workload Scheduler (Programmnummer 5698-WSH) und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

© Copyright IBM Corporation 2011, 2016. © Copyright HCL Technologies Limited 2016, 2017

Inhaltsverzeichnis

Abbildungsverzeichnis	v
--	----------

Tabellen	vii
---------------------------	------------

Informationen zu dieser Veröffentlichung	ix
---	-----------

Zielgruppe	ix
Behindertengerechte Bedienung	ix
Technische Schulung	ix
Unterstützungsinformationen	ix

Kapitel 1. Übersicht.	1
--	----------

Kapitel 2. Installation und Konfiguration	3
--	----------

Software für den Agenten für z/OS laden	4
Installationshilfe EELINST ausführen	6
Exits für die SMF- und JES-Ereignisüberwachung hinzufügen	12
SMF-Exits	13
JES2-Exits	13
JES3-Exits	14
Makro EEEXIT aufrufen	15
SYS1.PARMLIB aktualisieren	18
Subsysteme definieren	18
Lademodulbibliothek autorisieren	20
SMF-Parameter aktualisieren	21
z/OS-Speicherausgangsoptionen aktualisieren	22
z/OS-Verbindungsbibliotheksdefinition aktualisieren	22
Produkt automatisch starten	23
RACF für die gestartete Task des Agenten für z/OS aktualisieren	23
SSL-Umgebung einrichten	24
SYS1.PROCLIB aktualisieren	25
Installation abschließen	27
Agenten starten und Verbindung prüfen	27
Sicherstellen, dass alle Installationstasks ausgeführt wurden	28
Nachrichtenprotokoll prüfen	28
Überwachungsereignisse überprüfen	28
Problembestimmung für Überwachungsereignisse ausführen	30
Empfehlungen für die Zuordnung der Jobbibliotheksdatei (EELJBLIB)	32
Anpassungsparameter	32
Laufzeitoptionen für den Ereignisschreiber angeben	33
Exitrichtlinie für den Agenten angeben	33
HTTP-Verbindungsoptionen definieren	35
Allgemeine Laufzeitoptionen für den Agenten angeben	38

Exits des Agenten für z/OS konfigurieren	43
Exit EELUX000 (Start/Stop) konfigurieren	43
Exit EELUX002 (Lesen in der Jobbibliothek) konfigurieren	44
Exit EELUX004 (Ereignisfilterung) konfigurieren	48
Agent in einer Sysplex-Umgebung ausführen	49

Kapitel 3. Verwendung	51
--	-----------

Computer- und Workstationname des Agenten	51
Agenten für z/OS auflisten	52
Jobs definieren	52
Jobs in Dynamic Workload Console definieren	53
Definition in Composer	58
JCL definieren	61
HFS- oder ZFS-Dateiauslösung	62
Syntax	62
Argumente	63
Beispiel	64
Jobs übergeben	64
Variablen in Jobs verwenden	69
Variablen, die von IBM Workload Scheduler aufgelöst werden	69
Variablen, die vom Agenten für z/OS aufgelöst werden	75
Jobinstanzen verwalten	96
Jobs überwachen	97
Steuern, wie der Ereignisschreiber Jobbeendigungs-codes für bestimmte Jobs aufzeichnet	101
Jobprotokolle anzeigen	102
Systembefehle zur Steuerung des Agenten verwenden	103
Domänenmanager wechseln	105

Kapitel 4. Fehlerbehebung und Referenz	107
---	------------

Informationen zu Resynchronisationsnachrichten	107
Komponentenversionen müssen zur vollständigen Nutzung der aktuellen Funktionalität gleichen Stand haben	109
Vollständige Füllung des DB2-Transaktionsprotokolls stoppt Jobverarbeitung	110
Datenbereiche	111
Nachrichten	207

Bemerkungen	209
------------------------------	------------

Marken	211
Bedingungen für Produktdokumentation	211

Index	213
------------------------	------------

Abbildungsverzeichnis

1. Agent für z/OS - Rolle 1
2. Agent für z/OS in einer SYSPLEX-Konfiguration. 50
3. Seite 'Allgemein' mit einer neuen Definition für den Agenten für z/OS 56
4. Nach Verweis angegebene JCL 57
5. Nach Definition angegebene JCL 58
6. Route eines Jobs innerhalb des Agenten für z/OS. 66
7. Route eines Statusereignisses innerhalb des Agenten für z/OS, wenn es von JES auf dem Weg zu IBM Workload Scheduler zurückgegeben wird 68

Tabellen

1. Prüfliste für die Installation des Agenten für z/OS	3	12. Unterstützte Variablen in JSDL-Definitionen	70
2. Beispielbibliotheksmitglied für die Dateien des Agenten für z/OS.	5	13. Eigenschaften für dynamische Jobs im IBM Workload Scheduler-Agenten für z/OS	72
3. Von SMP/E geladene Bibliotheken des Agenten für z/OS.	5	14. Symbole, die das Ende von Variablen markieren	79
4. Von der Installationshilfe erstellte Beispieljobs	8	15. Vordefinierte jobstrombezogene Variablen	80
5. Mit dem Agenten für z/OS verwendete Dateien	9	16. Vordefinierte jobbezogene Variablen	82
6. Liste der Beispiexitis für die Ereignisüberwachung	13	17. Vordefinierte datumsbezogene Variablen	82
7. Beispiele für MAXECSA-Speicherwerte	19	18. Vordefinierte Variablen mit dynamischem Format	83
8. Erforderliche Dateien für den Agenten für z/OS.	26	19. Substitutionsergebnisse unter Verwendung des dynamischen Formats	86
9. Optionale Dateien für den Agenten für z/OS	26	20. Jobereignisse und -status in den verschiedenen Komponenten.	97
10. Vom Agenten für z/OS generierte Ereignisse	29	21. Zurückgegebene Fehlercodes nach der Übergabe eines Jobs	100
11. Typen fehlender Ereignisse und entsprechende Aktionen zur Problembestimmung.	30		

Informationen zu dieser Veröffentlichung

In dieser Veröffentlichung wird die Installation, Konfiguration und Verwendung des verteilten IBM Workload Scheduler-Agenten für z/OS beschrieben.

Zielgruppe

Informationen zur Zielgruppe dieser Veröffentlichung.

Die vorliegende Veröffentlichung richtet sich an Benutzer, die mithilfe des Agenten für z/OS Arbeit von IBM Workload Scheduler aus im JES2- oder JES3-Subsystem von z/OS terminieren wollen.

Behindertengerechte Bedienung

Mithilfe der Funktionen zur behindertengerechten Bedienung können Benutzer mit einer körperlichen Behinderung, z. B. mit eingeschränkter Mobilität oder mit eingeschränktem Sehvermögen, Softwareprodukte erfolgreich nutzen.

Mit diesem Produkt erhalten Sie Unterstützungstechnologie für Sprachausgabe und Navigation in der Schnittstelle. Sie können alle Funktionen in der grafischen Benutzerschnittstelle auch über die Tastatur statt mit der Maus bedienen und nutzen.

Vollständige Informationen finden Sie im Anhang zur behindertengerechten Bedienung in *IBM Workload Scheduler Benutzerhandbuch und Referenz*.

Technische Schulung

Cloud & Smarter Infrastructure stellt technische Schulungen bereit.

Informationen zur technischen Schulung über Cloud & Smarter Infrastructure finden Sie unter <http://www.ibm.com/software/tivoli/education>.

Unterstützungsinformationen

IBM bietet mehrere Möglichkeiten, über die Sie beim Auftreten eines Problems Unterstützung erhalten können.

Wenn Sie ein Problem mit Ihrer IBM Software haben, wollen Sie es schnell lösen. IBM bietet folgende Möglichkeiten, damit Sie die erforderliche Unterstützung erhalten:

- Wissensbasen durchsuchen: Sie können eine umfangreiche Sammlung mit bekannten Problemen, Problemumgehungen, technischen Hinweisen und anderen Informationen durchsuchen.
- Programmkorrekturen anfordern: Sie können nach den neuesten, für Ihr Produkt bereits verfügbaren Programmkorrekturen suchen.
- IBM Software Support kontaktieren: Wenn Sie Ihr Problem nach wie vor nicht beheben können und Sie Hilfe von einem IBM Mitarbeiter benötigen, können Sie sich über verschiedene Wege mit dem IBM Software Support in Verbindung setzen.

Weitere Informationen zu diesen drei Ansätzen zur Problembehebung finden Sie im Anhang mit den Unterstützungsinformationen in *IBM Workload Scheduler: Troubleshooting Guide*.

Kapitel 1. Übersicht

Der Agent für z/OS wird verwendet, um Verarbeitungsoptionen von IBM Workload Scheduler aus auf dem JES2- oder JES3-Subsystem von z/OS zu planen.

Der Agent für z/OS wird im z/OS-System installiert und konfiguriert. Sofort nach seiner Konfiguration stellt der Agent automatisch über das Protokoll HTTP oder HTTPS eine Verbindung zur Komponente Dynamic Workload Broker von IBM Workload Scheduler her (diese wurde beim Konfigurationsprozess definiert).

Der Agent setzt die Installation von IBM Workload Scheduler Version 8.6 oder höher in Ihrer Umgebung voraus.

Mit dem Agenten für z/OS können Sie Jobs und Zeitpläne in IBM Workload Scheduler definieren und einen Teil der Jobs an ein z/OS-System übergeben. Sie führen die Planungstasks unter IBM Workload Scheduler aus, während die Ausführung beim z/OS-System angefordert wird.

Wie die nächste Abbildung zeigt, agiert der Agent für z/OS als Proxy zwischen Dynamic Workload Broker, d. h. der IBM Workload Scheduler-Komponente, die die Workload übergibt, und JES, d. h. der Komponente im z/OS-System, die die Workload ausführt. Der Agent übergibt die Workload von IBM Workload Scheduler an JES und gibt alle Updates zur Ausführung der Workload in Form von Ereignissen an IBM Workload Scheduler zurück.

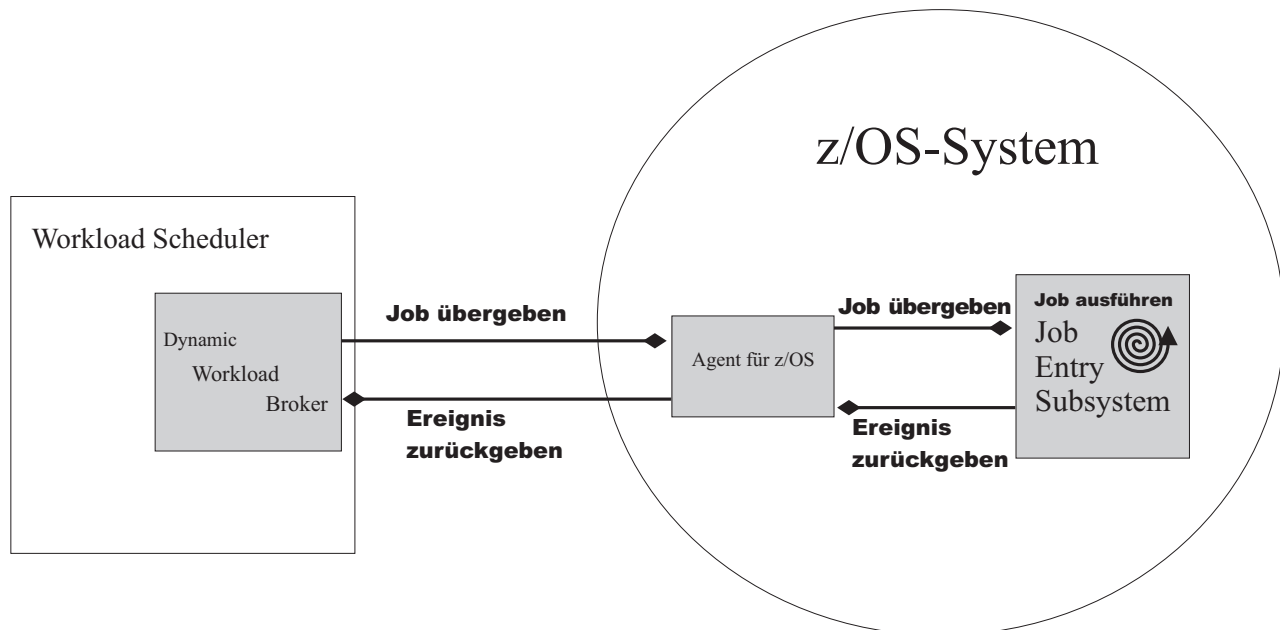


Abbildung 1. Agent für z/OS - Rolle

Der Agent für z/OS ist eine schlanke End-to-End-Terminierungslösung, mit der von JES zu verarbeitende Workloads vollständig von IBM Workload Scheduler aus definiert und verwaltet werden können, ohne dass Definitionen auf der z/OS-Seite erforderlich sind.

Der Agent verwendet das HTTP-Übergabeprotokoll, um Jobübergabeanforderungen zu empfangen und um Benachrichtigungen zum Jobstatus asynchron zurückzusenden. Bei der Konfiguration des Agenten muss der Installationsverantwortliche den Hostnamen des Dynamic Workload Broker angeben, zu dem der Agent eine Verbindung herstellen soll.

Die an einen Agenten für z/OS übergebenen Jobs sind anderen dynamischen IBM Workload Scheduler-Jobs ähnlich. In der Jobdefinition wird die XML-Syntax von JSDL (Job Submission Description Language) verwendet, wobei der Anwendungsname JCL lautet, und sie enthält einen der folgenden Informationstypen:

- Die gesamte JCL-Definition der Task, die von JES ausgeführt werden soll. In diesem Fall ist die JCL Teil der Jobdefinition, die in der IBM Workload Scheduler-Datenbank gespeichert und zusammen mit den übrigen Teilen des Jobs bei der Übergabe an den Agenten gesendet wird. Dies wird als Übergabe nach Definition bezeichnet.
- Die Namen der Datei und des Members, in denen sich die JCL-Definition in dem z/OS-System befindet, auf dem der Agent aktiv ist. Wenn der Job übergeben wird, verwendet der Agent diese Informationen, um die JCL-Definition im z/OS-System zu ermitteln. Dies wird als Übergabe nach Verweis bezeichnet.

Zum Definieren dieser Jobs können Sie die folgenden Schnittstellen verwenden:

- Composer- oder Dynamic Workload Broker-Befehlszeile
- Workload Designer von Dynamic Workload Console
- Grafische Dynamic Workload Broker-Benutzerschnittstelle

Der Agent unterstützt die Terminierung von Jobs, jedoch nicht von gestarteten Tasks, auf dem z/OS-System.

Die JCL-Anpassung (z/OS-Begriff: Variablensubstitution) wird ausgeführt, bevor die JCL mit den Variablenwerten, die in den Variablen Tabellen in IBM Workload Scheduler angegeben sind, an JES übergeben wird.

Die Jobs des Agenten für z/OS werden ebenso verwaltet wie andere IBM Workload Scheduler-Jobs. Mit Dynamic Workload Console können Sie grafische oder tabellarische Anzeigen oder Hostlisten dieser Jobs darstellen. Sie können auch die zugehörigen Jobprotokolle anzeigen sowie die meisten Aktionen (wie rerun und cancel, jedoch nicht kill) ausführen, die normalerweise für andere Jobs verfügbar sind.

Von Dynamic Workload Console oder der IBM Workload Scheduler-Befehlszeile aus kann das Jobprotokoll nur so lange angezeigt werden, wie die Jobausgabe sich im JES-Spoolprogramm befindet.

Kompatibilitätsanforderungen zwischen dem Agenten für z/OS und den IBM Workload Scheduler for z/OS-Trackern

Der Agent für z/OS und die IBM Workload Scheduler for z/OS-Tracker können nur koexistieren, wenn Sie die Standardexits der IBM Workload Scheduler for z/OS-Tracker ausführen. Wenn Sie Probleme bei der Koexistenz befürchten, dürfen Sie daher keine der Beispiexitis installieren, die mit dem Agenten bereitgestellt werden.

Koexistenz mit der Zugriffsmethode für z/OS und das Gateway

Der Agent für z/OS und das Gateway können auf demselben System koexistieren.

Kapitel 2. Installation und Konfiguration

In diesem Kapitel sind die Tasks beschrieben, die Sie zum Herunterladen, Installieren und Konfigurieren der Software für den Agenten für z/OS auf dem z/OS-Zielsystem ausführen müssen.

Der Agent für z/OS wird im z/OS-System installiert und konfiguriert. Sofort nach seiner Konfiguration stellt der Agent automatisch über das Protokoll HTTP oder HTTPS eine Verbindung zur Komponente Dynamic Workload Broker von IBM Workload Scheduler her (diese wurde beim Konfigurationsprozess definiert).

Prüfliste für die Installation

In der folgenden Tabelle sind die Installations- und Konfigurationstasks für den Agenten für z/OS zusammengefasst:

Tabelle 1. Prüfliste für die Installation des Agenten für z/OS

Task	Beschreibung
1	„Software für den Agenten für z/OS laden“ auf Seite 4 Führen Sie die folgenden Schritte aus: <ol style="list-style-type: none">1. Führen Sie SMP/E aus, um die Software für den Agenten für z/OS zu empfangen.2. Wenden Sie die Wartung für den Agenten für z/OS an.
2	„Installationshilfe EELINST ausführen“ auf Seite 6 Führen Sie EELINST für jede zu erstellende Instanz des Agenten für z/OS aus, sobald die Software für den Agenten für z/OS geladen ist. Das Programm unterstützt Sie bei Folgendem: <ul style="list-style-type: none">• Beispieljob-JCL erstellen, um angepasste Beispiele im Dialog EELINST zu generieren.• Dateien für den Agenten zuordnen.• SSL-Zertifikate hinzufügen.• Initialisierungsanweisungen definieren, um Member in der Parameterbibliothek zu erstellen (diese wird durch die Datendefinitionsanweisung EELPARM in der gestarteten Task des Agenten für z/OS angegeben).
3	Exits für die SMF- und JES-Ereignisüberwachung hinzufügen Anmerkung: Führen Sie diesen Schritt aus, wenn noch kein Tracker für IBM Workload Scheduler for z/OS im System ausgeführt wird. Wenn er vorhanden ist, müssen Sie die Exits für die aktuelle Produktversion anwenden. Wenn Sie z. B. Version 9.2 des Trackers für IBM Workload Scheduler for z/OS ausführen, müssen Sie die Exits für IBM Workload Scheduler for z/OS Version 9.2 anwenden. Bei einer Ausführung in einer JES3-Umgebung müssen Sie allerdings immer den Exit IATUX09 anwenden, der mit dem Agenten für z/OS bereitgestellt wird.

Tabelle 1. Prüfliste für die Installation des Agenten für z/OS (Forts.)

Task	Beschreibung
4	<p>SYS1.PARMLIB aktualisieren</p> <p>Führen Sie die folgenden Tasks aus, wenn sie auf Ihre Installation anwendbar sind:</p> <ul style="list-style-type: none"> • Subsystem für den Agenten für z/OS definieren (IEFSSNnn). • Lademodulbibliothek für den Agenten für z/OS autorisieren (IEAAPFnn oder PROgnn). • Definitionen für den Speicherausgangsinhalt aktualisieren. • Definition für die z/OS-Verbindungsbibliothek aktualisieren (LNKLSTnn). • SMF-Parameter aktualisieren (SMFPRMnn). Führen Sie diesen Schritt nur aus, wenn noch kein Tracker für IBM Workload Scheduler for z/OS im System ausgeführt wird. • Auswählen, ob der Agent automatisch gestartet werden soll (COMMNDnn).
5	„RACF für die gestartete Task des Agenten für z/OS aktualisieren“ auf Seite 23
6	<p>SSL-Umgebung einrichten</p> <p>Führen Sie die folgenden Schritte aus:</p> <ol style="list-style-type: none"> 1. Erstellen Sie alle privaten Schlüssel, Zertifikate und Ketten für anerkannte Zertifizierungsstellen, die Sie in Ihrem Netz verwenden wollen. 2. Geben Sie die SSL-Schlüsselwörter in der Initialisierungsanweisung HTTPOPTS an.
7	<p>SYS1.PROCLIB aktualisieren</p> <p>Erstellen Sie eine JCL-Prozedur für den Adressraum.</p>
8	<p>Installation abschließen</p> <ol style="list-style-type: none"> 1. Führen Sie einen IPL auf dem System aus, auf dem Sie den Agenten installiert haben. 2. Überprüfen Sie die Installation.

Software für den Agenten für z/OS laden

Als ersten Installationsschritt verwenden Sie SMP/E, um die Software für den Agenten für z/OS vom Programmband herunterzuladen.

Verarbeiten Sie das Softwareverteilungsband mit den Funktionen von System Modification Program Extended (SMP/E), um die Software für den Agenten für z/OS auf Ihr z/OS-System zu laden. Dadurch werden die erforderlichen Softwarebibliotheken auf Ihrem System erstellt oder aktualisiert.

Zum Herunterladen der Agentensoftware können Sie Folgendes verwenden:

- Die SMP/E-Dialoge.
- Die Beispieljobs, die auf dem Programmband bereitgestellt werden. Verwenden Sie diese mit den SMP/E-Befehlen RECEIVE, APPLY und ACCEPT. Bei den SMP/E-Jobs wird vorausgesetzt, dass alle für die SMP/E-Ausführung erforderlichen DDDEF-Einträge in den entsprechenden Zonen definiert wurden.

In den folgenden Abschnitten sind die Beispiele aufgelistet und beschrieben, die zum Ausführen der Befehle RECEIVE, APPLY und ACCEPT bereitgestellt wurden. Weitere Details enthält das Programmverzeichnis.

Umgebung einrichten

Sie können die Beispielbibliotheksmember EELALSMP und EELALLOC verwenden, um die SMP/E-Umgebung und die Produktbibliotheken für den Agenten für z/OS zu erstellen und zu initialisieren, die zur Unterstützung der Installation und der kontinuierlichen Wartung des Agenten erforderlich sind.

Der Job EELALSMP initialisiert ein SMP/E-CSI (und fügt dabei eine globale Zone hinzu) sowie die FMID des Agenten für z/OS.

Der Job EELALLOC ordnet alle Ziel- und Verteilungsbibliotheken des Agenten für z/OS zu.

Beispiele für die RECEIVE-, APPLY- und ACCEPT-Verarbeitung verwenden

In der folgenden Tabelle sind die Beispielbibliotheksmember aufgelistet, die auf dem Programmband für Sie bereitgestellt werden:

Tabelle 2. Beispielbibliotheksmember für die Dateien des Agenten für z/OS

Beispielmember	Funktion
EELACPT	Für die Ausführung der SMP/E-ACCEPT-Verarbeitung für die Dateien des Agenten für z/OS
EELALLOC	Für die Zuordnung aller Dateien des Agenten für z/OS
EELAPPLE	Für die Ausführung der SMP/E-APPLY-Verarbeitung für die Dateien des Agenten für z/OS
EELDDDEF	Für die Einrichtung von Datendefinitionen für alle Dateien des Agenten für z/OS
EELRECVE	Für die Ausführung der SMP/E-RECEIVE-Verarbeitung für die Dateien des Agenten für z/OS

Möglicherweise müssen Sie die Verteilungsbibliothek und den Zonennamen so ändern, dass sie mit den Namen übereinstimmen, die im CSI des Agenten für z/OS definiert sind.

Weitere Details enthält das Programmverzeichnis.

Ergebnisse

In der folgenden Tabelle sind die Verteilungs- und Zielbibliotheken beschrieben, die von SMP/E erstellt oder aktualisiert werden.

Tabelle 3. Von SMP/E geladene Bibliotheken des Agenten für z/OS

SMP/E-Datendefinitionsname		Beschreibung
Verteilung	Ziel	
AEELCLIB	SEELCLIB	CLISTs
AEELDATA	SEELDATA	Standard-SSL-Zertifikate
AEELMAC0	SEELMAC0	Makros für Assembler

Tabelle 3. Von SMP/E geladene Bibliotheken des Agenten für z/OS (Forts.)

SMP/E-Datendefinitionsname		Beschreibung
Verteilung	Ziel	
AEELMISC	SEELMISC	Lizenzinformationen
AEELMOD0 (Objekt)	SEELMOD0 (Laden)	Module des Agenten für z/OS
AEELMSG0	SEELMSG0	Nachrichten
AEELPNL0	SEELPNL0	Anzeigen für die Installationshilfe EELINST
AEELSAMP	SEELSAMP	Beispiele für Exits, Programme und JCL

Sie sollten die Lademodule in einer separaten Bibliothek anordnen. Erstellen Sie die Bibliothek, bevor Sie die SMP/E-Jobs ausführen.

Sie können die neuen Lademodule jedoch auch in einer Ihrer vorhandenen Lademodulbibliotheken anordnen, beispielsweise in SYS1.LINKLIB. Die übrigen von SMP/E geladenen Dateien sind neue Dateien, die Sie vor der Ausführung der SMP/E-Jobs ausführen müssen. Das Programmverzeichnis für den Agenten für z/OS enthält die JCL und die Anweisungen zum Laden der Software.

Nachdem Sie die Software für den Agenten für z/OS geladen haben, wenden Sie sämtliche empfohlene Wartung an, die im PSP-Bucket beschrieben ist.

Installationshilfe EELINST ausführen

EELINST ist ein CLIST-gesteuerter ISPF-Dialog, der Sie bei der Einrichtung einer Instanz des Agenten für z/OS unterstützt. Richten Sie EELINST ein, sobald die Software für den Agenten für z/OS installiert ist.

EELINST unterstützt Sie wie folgt bei der Installation:

- Sie erstellt die Stapeljob-JCLs, die an Ihre Anforderungen angepasst sind und die Sie für eine vollständige Installation verwenden können.
- Sie erstellt die Dateien, die für Ihre Instanz des Agenten für z/OS verwendet werden.
- Sie definiert die Initialisierungsanweisungen in der Parameterbibliothek (EELPARM).
- Sie lädt die Standard-SSL-Zertifikate in die RACF-Schlüsselspeicherdatenbank.
- Sie listet die übrigen manuellen Schritte auf, die zum Abschließen der Installation erforderlich sind.

Installationshilfe EELINST einrichten

EELINST liest Entwurfs-JCLs aus der Bibliothek SEELSAMP, passt die JCLs an und schreibt die angepassten JCLs in eine Ausgabebibliothek, die Sie angeben. Die Komponenten von EELINST befinden sich in den folgenden Bibliotheken:

SEELCLIB

CLIST für die Steuerung des Dialogs

SEELPNL0

EELINST-Anzeigen

SEELSAMP

Beispiel-JCL

Damit EELINST ausgeführt werden kann, müssen Sie diese Bibliotheken den Datendefinitionsanweisungen in Ihrer TSO-Sitzung zuordnen:

- SEELCLIB zu SYSPROC
- SEELPNL0 zu ISPLIB
- SEELSAMP zu ISPSLIB

EELINST ausführen

Zum Aufrufen von EELINST geben Sie den TSO-Befehl EELINST in einer ISPF-Umgebung ein. Die folgende Anzeige wird geöffnet:

```
                Welcome to the IBM Workload Scheduler distributed
                Agent für z/OS Installation

The installation program can assist you with the following operations:

△ Building a batch-job JCL that is tailored to your requirements and that can be
  used for a complete installation.

△ Creating the data sets used for your Agent for z/OS address space.

△ Defining initialization statements in the parameters library (EELPARM).

△ Loading the default SSL certificates on the RACF keystore database.

△ Providing a list of all the other manual steps you must apply on the system to
  complete the installation.

Press ENTER to continue the installation. Press PF3 to quit.
```

Wenn Sie die Eingabetaste drücken, wird die folgende Anzeige geöffnet, in der einige Felder bereits mit Standardwerten ausgefüllt sind:

```

Job statement information:
//ZAGEINST JOB JOBNN _____
_____
_____

Product libraries (Steplib is optional):
Steplib library   ===> _____
Message library   ===> ZAGE.INST.SEELMSG0 _____
Data library name ===> ZAGE.INST.SEELDATA _____

Agent for z/OS subsystem information:
Datasets prefix   ===> TWSSD _____
Subsystem name    ===> ZAGE          Subsystem name
Unit name         ===> 3390         Default unit name
Volume serial     ===> EELVOL_      Default volume serial

Input library: specified in the logon procedure (ISPSLIB)
Output library:
Output dsn name   ===> _____

SYSOUT class      ===> *           SYSOUT class for reports

Press ENTER to continue, PF12 for previous panel, PF3 to quit.

```

Die Länge des Subsystemnamens kann höchstens vier Zeichen betragen.

Die Länge des Dateipräfix kann höchstens 26 Zeichen betragen.

Wenn Sie die Eingabetaste drücken, werden die folgenden Beispieljobs in der Datei *Output dsn name* erstellt:

Tabelle 4. Von der Installationshilfe erstellte Beispieljobs

Beispielname	Beschreibung
EELAGT	Generiert ein Prozedurbeispiel für die gestartete Task des Agenten für z/OS.
EELAGTP	Generiert Standardparameter für das Prozedurbeispiel für die gestartete Task des Agenten für z/OS.
EELALLDS	Ordnet die Dateien zu, die für die gestartete Task des Agenten für z/OS verwendet werden.
EELFLWAS	Ruft das Dienstprogramm filewatch auf, das zum Überprüfen der HFS- oder ZFS-Dateien verwendet wird.
EELJER2V	Stellt die Benutzermodifikation der JES2-Exits wieder her.
EELJER3O	Stellt die JES3-Benutzermodifikation EELUX091 wieder her.
EELJER3U	Stellt die Exits EELUX091 und EELUX191 als JES3-Benutzermodifikationen wieder her.
EELJES2	Assembliert und stellt eine Programmverbindung zu JES2 EXIT7 her.
EELJES2U	Installiert die Benutzermodifikation JES2 EXIT7.
EELJES2V	Installiert die Benutzermodifikation JES2 EXIT51.
EELJES21	Assembliert und stellt eine Programmverbindung zu JES2 EXIT51 her.

Tabelle 4. Von der Installationshilfe erstellte Beispieljobs (Forts.)

Beispielname	Beschreibung
EELJES3	Assembliert und stellt eine Programmverbindung zu den JES3-Exits her.
EELJES3O	Installiert die JES3-Benutzermodifikation EELUX091.
EELJES3U	Installiert die JES3-Benutzermodifikation.
EELRCERT	Kopiert Standardzertifikate für die SSL-Kommunikation (Member EELCERCL in Bibliothek MISC) in RACF.
EELRETWT	Beispielprogramm zum Simulieren von Abbrüchen, Rückgabecodes und Wartezuständen.
EELRMD	Eine Readme-Datei, die die aufeinanderfolgenden manuellen Schritte auflistet.
EELSMF	Assembliert und installiert die SMF-Exits.

Anschließend wird die folgende Anzeige geöffnet:

```

                                Create the data sets used for the Started task (EELALLDS)          2/5

The data sets in EELALLDS will be named:
DATASET PREFIX.SUBSYSTEM NAME.NAME

SUBMIT/EDIT JOB    ==> E          S to submit JOB, E to edit
                                N to skip to the next panel

Press ENTER to continue the installation. Press PF3 to quit.

```

Nach der Übergabe von EELALLDS sind alle Dateien für die neue gestartete Task verfügbar. Dies umfasst Folgendes:

Tabelle 5. Mit dem Agenten für z/OS verwendete Dateien

Von EELALLDS zugeordnete Datei	Datendefinitionsname	Dateibeschreibung
-	EELBRDS	Interner Leser.
dateipräfix.subsystemname.EELDUMP dateipräfix.subsystemname.SYSDUMP	EELDUMP SYSDUMP	Diagnosedateien, in die der Agent für z/OS Fehlerbehebungsinformationen schreibt, wenn bei der Gültigkeitsprüfung interne Fehlerbedingungen erkannt werden.

Tabelle 5. Mit dem Agenten für z/OS verwendete Dateien (Forts.)

Von EELALLDS zugeordnete Datei	Datendefinitionsname	Dateibeschreibung
<i>dateiprefix.subsystemname.EV</i> <i>dateiprefix.subsystemname.HT</i> <i>dateiprefix.subsystemname.HTREF</i>	EELEVDS EELHTDS EELHTREF	Ereignisdateien mit Datensätzen, die Ereignisse beschreiben, die von den Jobüberwachungsfunktionen des Agenten für z/OS erstellt wurden. EELHTDS zeichnet Ereignisse auf, die von Dynamic Workload Broker kommen, während EELEVDS von JES und SMF erstellte Ereignisse aufzeichnet, die an Dynamic Workload Broker übertragen werden sollen. EELHTREF ist eine Servicedatei, die für Jobs verwendet wird, die nach Verweis übergeben werden oder bei denen die JCL eine Variablenuflösung erfordert.
-	EELJBLIB	Jobbibliotheksdatei für die JCLs, die nach Verweis übergeben werden. Wenn der Dateiname nicht in einer Jobdefinition des Agenten für z/OS angegeben wird, durchsucht der Agent EELJBLIB nach dem Membernamen, mit dem die JCL gespeichert wurde.
-	EELMLIB	Nachrichtenbibliothek.
<i>dateiprefix.subsystemname.MLOG</i>	EELMLOG	Nachrichtenprotokoll.
<i>dateiprefix.subsystemname.PARM</i>	EELPARM	Parameterbibliothek mit Initialisierungsanweisungen, die Laufzeitoptionen für das Subsystem des Agenten für z/OS definieren.

Eine Beispielkopie der gestarteten Task des Agenten (beispielsweise AGT1) wird in die Ausgabedatei eingefügt.

Prüfen Sie die Jobausgabe, um sicherzustellen, dass dieser Schritt ausgeführt wurde, bevor Sie mit den folgenden Schritten fortfahren.

Im nächsten Schritt werden die Standardinitialisierungsanweisungen in die Parameterbibliothek eingefügt, die von der gestarteten Task des Agenten verwendet wird.

Define initialization statements in the parameters library 3/5

INITIALIZATION PARAMETERS ==> AGT1P ___ Member in the EELPARM library
CREATE/EDIT PARAMETERS ==> E C to create, E to edit

Press ENTER to continue the installation. Press PF3 to quit.

Die Länge des Membernams in der Bibliothek EELPARM kann höchstens acht Zeichen betragen.

Wählen Sie E aus, um die Initialisierungsanweisungen jetzt mit gültigen Werten zu definieren, oder wählen Sie C aus, um die Anweisungen nur zu erstellen und ihre Funktionsdefinition zu einem späteren Zeitpunkt auszuführen (dazu ist ein Neustart des Agenten erforderlich). In „Anpassungsparameter“ auf Seite 32 finden Sie weitere Informationen zu den Initialisierungsanweisungen.

In der nächsten Anzeige haben Sie die Möglichkeit, die Standard-SSL-Zertifikate in die RACF-Schlüsselspeicherdatei zu importieren.

More: +
Run the EELRCERT job to import the default certificates to RACF 4/5

To use them, configure the HTTPPTS initialization statement to activate the SSL communication, setting parms TDWSSL and/or SSL and add the following configurations:
SSLKEYRINGTYPE(SAF)
SSLKEYRING(EELRING)

Note: If you already run default certificates with IBM Workload Scheduler for z/OS, skip this step and configure accordingly.

SUBMIT/EDIT JOB ==> N S to submit JOB, E to edit
N to skip to the next panel

Wenn Sie auf demselben System auch IBM Workload Scheduler for z/OS und die zugehörigen Standard-SSL-Zertifikate ausführen, überspringen Sie diesen Schritt. Konfigurieren Sie jedoch den Parameter SSLKEYRING mit dem Wert, der bereits für diesen Scheduler verwendet wird. Wenn Sie diese Konfiguration nicht ausführen, hat die Übergabe von EELCERT keine Auswirkungen und es wird ein RC4-Fehlercode zurückgegeben.

Schließlich wird eine Liste aller verbleibenden Schritte angezeigt, die Sie in Ihrer z/OS-Umgebung manuell ausführen müssen, um die Installation abzuschließen:

- o Add SMF and JES event tracking exits.
Complete this step only if you do not already have an IBM Workload Scheduler for z/OS tracker active on the z/OS system. Otherwise, see the Scheduling with the Agent for z/OS manual.
- o Update SYS1.PARMLIB.
 - Define the Agent for z/OS subsystem (IEFSSNnn).
 - Authorize the Agent for z/OS load module library (IEAAPFnn or PROgnn).
 - Update dump-content definitions.
 - Update the z/OS link-library definition (LNKLSTnn).
 - Update SMF parameters (SMFPRMnn). Apply this change only if you do not already run an IBM Workload Scheduler for z/OS tracker on the z/OS system.
 - Choose whether to start the Agent for z/OS automatically (COMMNDnn).
- o Set up the RACF environment.
- o Set up the SSL environment.
 - Create as many private keys, certificates, and trusted certification authority (CA) chains as you plan to use in your network.
 - Configure the scheduler, by specifying the HTTPPTS statement for each component of your network.
- o Update SYS1.PROCLIB.
 - Create a JCL procedure for the z/OS address space.
- o Activate TCP/IP connections.
 - Add TCP/IP network definitions. Define an IP address for the Agent for z/OS.
 - Add TCP/IP initialization options. Include initialization statement options in the parameters library for all the Agent for z/OS started tasks.
- o Complete the installation.
 - IPL each system where you have installed the Agent for z/OS.
 - Verify the installation.

This information is also listed in the \$EELRMD member of the output data set

Press ENTER to complete, PF12 to return to the previous panel, PF3 to quit.

Die Schritte werden in den folgenden Abschnitten beschrieben.

Exits für die SMF- und JES-Ereignisüberwachung hinzufügen

Der Agent für z/OS überwacht den Fortschritt von Jobs über das z/OS-System mithilfe von JES2-, JES3- und SMF-Exitpunkten. Fügen Sie diese Exits auf dem z/OS-System hinzu, auf dem Sie den Agenten installiert haben, es sei denn, Sie führen bereits einen Tracker mit einer unterstützten Version von IBM Workload Scheduler for z/OS aus. (In diesem Fall siehe „JES2-Exits“ auf Seite 13 und „JES3-Exits“ auf Seite 14.)

Mehrere Beispielexits zur Ereignisüberwachung, die die Installation der Ereignisüberwachung erleichtern, sind in der Beispielbibliothek SEELSAMP verfügbar. Zum Assemblieren und Installieren der Exits können Sie die bereitgestellte Beispiel-JCL verwenden, um die Exits als SMP/E-Benutzermodifikationen zu installieren. Sie können die Exits jedoch auch selbst assemblieren und per Link-Edit verknüpfen. Wenden Sie für JES-Exits Benutzermodifikationen in dem CSI an, in dem JES enthalten ist: dies ist die beste Methode. Sie hat den Vorteil, dass SMP die Exits automatisch erneut assembliert, wenn Wartung an den JES-Steuerblöcken angewendet wird, von denen der Agent für z/OS abhängig ist.

Alle Beispielexits verwenden das Makro EEEXIT, um Code zur Ereignisgenerierung zu erstellen. Weitere Details finden Sie in „Makro EEEXIT aufrufen“ auf Seite 15.

In der folgenden Tabelle sind die Beispiele beschrieben, die Sie zum Generieren und Installieren der Exits verwenden können. Der Beispielexit, die Entwurfs-JCL und die usermod-Einträge geben die Member in der Bibliothek SEELSAMP an.

Tabelle 6. Liste der Beispielexits für die Ereignisüberwachung

Exitname	Exittyp	Beispielexit	Beispiel-JCL/-usermod	Unterstütztes Ereignis
IEFACTRT	SMF	EELACTR1	EELSMF	Abschluss des Jobs und des Abschnitts
IEFUJI	SMF	EELUJI1	EELSMF	Jobstart
IEFU83	SMF	EELU831	EELSMF	Jobausgabeende
EXIT7	JES2	EELXIT74	EELJES2/ EELJES2U	JCT-Ein-/Ausgabeexit für JES2 bereinigen
EXIT51	JES2	EELXIT51	EELJES21/ EELJES2V	JES2-Exit für QMOD-Phasenänderung, z/OS Version 1.7 oder höher
IATUX09	JES3	EELUX091	EELJES3/ EELJES3O/ EELJES3U	Jobwarteschlange
IATUX19	JES3	EELUX191	EELJES3/ EELJES3U	Ausgabeverarbeitung beendet

SMF-Exits

In diesem Abschnitt finden Sie Details zum Hinzufügen der SMF-Exits für die Ereignisüberwachung, die vom Agenten für z/OS verwendet werden.

Sie müssen die Beispiel-JCL an die Anforderungen Ihrer Installation anpassen. Sie können beliebige Member der Bibliothek SEELSAMP in eine Ihrer eigenen Bibliotheken kopieren und die JCL manuell anpassen.

Wenn Sie mit der Aktivierung von SMF-Exits nicht vertraut sind, finden Sie weitere Informationen in „SMF-Parameter aktualisieren“ auf Seite 21 und in der Dokumentation zu SMF.

JES2-Exits

In diesem Abschnitt finden Sie Details zum Hinzufügen der JES2-Exits für die Ereignisüberwachung, die vom Agenten für z/OS verwendet werden. Wenn der Agent für z/OS und der Tracker für IBM Workload Scheduler for z/OS auf demselben System ausgeführt werden, sind die JES2-Exits bereits installiert.

Die Beispielbibliothek EELSAMP enthält eine Reihe von Membern, die Sie zum Assemblieren und Verknüpfen (Link-Edit) von JES-Exits verwenden können. EELJES2 und EELJES21 enthalten Beispiel-JCL zum Assemblieren und Verknüpfen der JES2-Exits. Es wird jedoch empfohlen, dass Sie die Member EELJES2U und EELJES2V verwenden. Mit der JCL, die in diesen Beispielen enthalten ist, werden die JES2-Exits als SMP/E-Benutzermodifikationen installiert. Die Benutzermodifikationen werden so definiert, dass die Zielzonen von JES und des Agenten für z/OS

über die Abhängigkeiten informiert werden. Damit ist sichergestellt, dass die zukünftige Wartung für beide Komponenten (d. h. JES2 und Agent für z/OS) ordnungsgemäß verarbeitet wird.

Das Beispiel EELJER2V wird bereitgestellt, um die JES2-Exits zurückzuweisen und bei Bedarf als SMP/E-Benutzermodifikationen wiederherzustellen.

Die Lademodule für die JES2-Exits (dies sind EXIT7 und EXIT51) heißen TWSEXIT7 und TWSXIT51 und ihre Eingangspunkte TWSENTR7 bzw. TWSENT51.

Das Beispielbibliotheksmitglied EELXIT74 enthält den Assembler Quellcode eines JES2-JCT-E/A-Exits, JESEXIT7. EELXIT74 wird für JES2 verwendet. Der Agent für z/OS verwendet JESEXIT7 zur Erkennung neuer Jobs im internen Leser und auch zum Erkennen der Bereinigung von Ausgabegruppen.

Das Beispielbibliotheksmitglied EELXIT51 enthält den Assembler Quellcode des JES2-Exits für die QMOD-Phasenänderung, JES2-EXIT51. Der Agent für z/OS verwendet JES2-EXIT51 zur Erkennung von Jobfehlern, die in der JES2-Eingangsphase auftreten.

Schließen Sie diese Datensätze in das JES2-Initialisierungsmitglied ein:

JES2-Initialisierungsanweisungen

Fügen Sie die folgenden Datensätze dem JES2-Initialisierungsmitglied hinzu:

```
Load Agent für z/OS exit mod */
EXIT(7)  ROUTINES=TWSENTR7,STATUS=ENABLED /*
Define EXIT7 entry point */
```

Und außerdem:

```
LOAD(TWSXIT51) /*
Load Agent für z/OS exit mod */
EXIT(51)  ROUTINES=TWSENT51,STATUS=ENABLED /*
Define EXIT51 entry point */
```

Zur dynamischen Installation der JES2-Exits verwenden Sie die folgenden Befehle, wenn die Module in der LNKLIST verfügbar sind:

```
$ADD LOADMOD(TWSEXIT7),STORAGE=PVT
$T EXIT(7),ROUTINES=TWSENTR7,
  STATUS=ENABLED
$ADD LOADMOD(TWSXIT51),STORAGE=PVT
$T EXIT(51),ROUTINES=TWSENT51,
  STATUS=ENABLED
```

Zum Aktualisieren eines Exits (der bereits installiert ist) verwenden Sie die folgenden Befehle, wenn die Module in der LNKLIST verfügbar sind:

```
$TLOADMOD(TWSEXIT7),REFRESH
$TLOADMOD(TWSXIT51),REFRESH
```

Weitere Informationen zu JES2-Initialisierungsanweisungen finden Sie im Handbuch *JES2 Initialization and Tuning Reference*.

JES3-Exits

In diesem Abschnitt finden Sie Details zum Hinzufügen der JES3-Exits für die Ereignisüberwachung, die vom Agenten für z/OS verwendet werden.

Bei einer Ausführung in einer JES3-Umgebung sollten Sie sicherstellen, dass Sie APAR PI47613 angewendet haben, der JES3 für den Agenten für z/OS unterstützt. Wenn auch ein Tracker für IBM Workload Scheduler for z/OS im selben System ausgeführt wird, stellen Sie sicher, dass Sie APAR PI57699 angewendet haben, um die JES3-Kompatibilität für den Tracker zu aktivieren.

Anmerkung: Der Exit IATUX29 wird nur durch den Tracker für IBM Workload Scheduler for z/OS bereitgestellt und verwendet. Der Exit IATUX09 wird nur durch den Agenten für z/OS bereitgestellt und verwendet. Der Exit IATUX19 wird durch den Tracker und durch den Agenten für z/OS bereitgestellt und verwendet.

Die Beispielbibliothek EELSAMP enthält eine Reihe von Mitgliedern, die Sie zum Assemblieren und Verknüpfen (Link-Edit) von JES-Exits verwenden können. In EELJES3 wird eine Beispiel-JCL bereitgestellt, um die JES3-Exits zu assemblieren und Programmverbindungen herzustellen. Es wird jedoch empfohlen, dass Sie die Member EELJES3O oder EELJES3U verwenden. Mit der JCL, die in diesen Beispielen enthalten ist, werden die JES3-Exits als SMP/E-Benutzermodifikationen installiert. Die Benutzermodifikationen werden so definiert, dass die Zielzonen von JES und des Agenten für z/OS über die Abhängigkeiten informiert werden. Damit ist sichergestellt, dass die zukünftige Wartung für beide Komponenten (d. h. JES3 und Agent für z/OS) ordnungsgemäß verarbeitet wird.

Das Beispiel EELJES3U wird bereitgestellt, um die Benutzermodifikationen IATUX09 und IATUX19 zu empfangen und anzuwenden. Das Beispiel EELJES3O wird bereitgestellt, um nur die Benutzermodifikation des Exits IATUX09 zu empfangen und anzuwenden. Das Beispiel EELJER3U wird bereitgestellt, um die beiden Benutzermodifikationen IATUX09 und IATUX19 zurückzuweisen und bei Bedarf wiederherzustellen. Das Beispiel EELJER3O wird bereitgestellt, um nur die Benutzermodifikation des Exits IATUX09 zurückzuweisen und bei Bedarf wiederherzustellen.

Sie können die Exits für ein JES3-System aktivieren, indem Sie sie mit einer Bibliothek verlinken, die vor SYS1.JES3LIB verknüpft ist. Sie können auch die vorhandenen Exits in SYS1.JES3LIB durch die mit dem Agenten für z/OS ausgelieferten Exits IATUX09 und IATUX19 ersetzen. Weitere Informationen finden Sie in *JES3 Initialization and Tuning Reference*.

Wenn Sie RC=4 und die Warnung ASMA303W Multiple address resolutions may result empfangen, wenn Sie den Exit IATUX19 assemblieren, der die Beispiele für JES3-Benutzermodifikationen ausführt, können Sie die Nachricht ignorieren. Wenn die Version ASMA90 des Compilers Fehler zurückmeldet und die Anweisung RMODE=ANY definiert ist, entfernen Sie die Anweisung RMODE=ANY aus dem Beispielexit.

Makro EELEXIT aufrufen

Die Beispielexits für die Ereignisüberwachung, die mit dem Agenten für z/OS geliefert werden, sind in Assemblersprache geschrieben. Der Ereignisüberwachungscode in diesen Exits wird durch ein Assemblermakro mit dem Namen EELEXIT generiert. In den folgenden Abschnitten ist beschrieben, wie das Makro EELEXIT aufgerufen wird.

EELEXIT in SMF-Exits aufrufen

EELEXIT erstellt seine eigene Adressierbarkeit in SMF-Exits. Das Makro speichert alle verwendeten Register und stellt sie wieder her. Damit dies funktioniert, erwartet das Makro, dass Register 13 auf einen z/OS-Standardsicherungsbereich zeigt.

Es gibt zwei Verfahren zum Aufrufen des Makros EELEXIT in einem SMF-Exit:

- EELEXIT aufrufen und dabei alle Register seit dem Aufruf des Exits unverändert lassen (mit Ausnahme von Register 15).
- Alle Register beim Eingang in den Exit speichern und anschließend EELEXIT durch Angabe der Adresse des ursprünglichen Sicherungsbereichs aufrufen.

In beiden Fällen muss EELEXIT im Supervisorstatus, PSW-Schlüssel 0, aufgerufen werden.

EELEXIT in JES-Exits aufrufen

In JES-Exits muss EELEXIT im Supervisorstatus, PSW-Schlüssel 1, aufgerufen werden. EELEXIT erwartet, dass die Codeadressierbarkeit bereits eingerichtet ist. Das Makro erwartet außerdem, dass die Register wie folgt eingerichtet sind:

- **EXIT7**
 - R0** JCT-Anzeiger für Lesen/Schreiben (JES2 SP Version 3 und niedriger); Adresse einer durch das JES2-Makro \$XPL zugeordneten Parameterliste (JES2 SP Version 4 und höher).
 - R1** Adresse der JCT, die gelesen oder geschrieben wird.
 - R13** Adresse des aktuellen PCE.
- **EXIT51**
 - R1** Adresse einer durch das JES2-Makro \$XPL zugeordneten Parameterliste (JES2 mit z/OS Version 1.7 oder höher).
- **IATUX09**
 - R11** Adresse des aktuellen FCT-Eintrags.
 - R11** Adresse des TVTABLE-Eintrags.
 - R13** Adresse des Eingabeservicedatenbereichs für die aktuelle Funktion.
- **IATUX19**
 - R8** Adresse des aktuellen JDS-Eintrags.
 - R9** Adresse des aktuellen RESQUEUE-Eintrags.
 - R11** Adresse des aktuellen FCT-Eintrags.
 - R12** Adresse des aktuellen TVTABLE-Eintrags.

Beachten Sie, dass diese Registerkonventionen bereits eingerichtet sind, wenn der Exit aufgerufen wird. Sie müssen EELEXIT so aufrufen, dass diese Register unverändert bleiben.

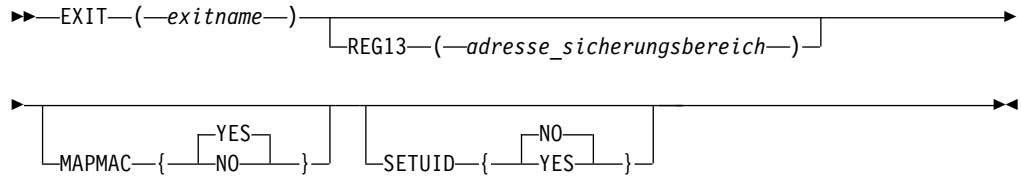
Wenn ein im Lieferumfang enthaltener JES-Beispielexit (oder das Makro EELEXIT) vom Benutzer geändert wird, müssen Sie sicherstellen, dass der Exit die Überwachung des Agenten für z/OS selbst nicht verhindert oder filtert.

Im Abschnitt NOTES des Prologs von EELEXIT finden Sie Informationen zum Registerinhalt, der durch EELEXIT in JES-Exits gelöscht wird.

Makroaufrufsyntax für EEEXIT

EEEXIT erstellt Exit-Code zur Ereignisüberwachung durch Generieren von Assemblercode, der in einem SMF- oder JES-Exit ausgeführt wird.

Syntax



Parameter

EXIT = *exitname*

Ein erforderliches Schlüsselwort, das den Namen des Exits definiert, in dem das Makro verwendet wird. Die folgenden Namen können angegeben werden: IEFACTRT, IEFUJI, EXIT7 und EXIT51. Für die Exits IEFACTRT und IEFUJI wird eine Warnung ausgegeben, wenn der Name des aktuellen Programmabschnitts sich von dem Namen unterscheidet, der durch das Schlüsselwort EXIT angegeben wird.

REG13 = *adresse_sicherungsbereich*

Ein optionales Schlüsselwort, das die Adresse des aktuellen Sicherungsbereichs für das Register angibt, wenn der SMF- oder JES2-Exit aufgerufen wurde. Der Standardwert für dieses Schlüsselwort ist von dem Namen abhängig, der durch das Schlüsselwort EXIT angegeben wird. Ist der aktuelle Exit EXIT7, lautet der Standardwert PCELPV. In allen anderen Fällen ist der Standardwert das zweite Vollwort im aktuellen Sicherungsbereich (falls der aktuelle Sicherungsbereich ordnungsgemäß verkettet ist und der vorherige Sicherungsbereich die Register beim Eingang in den Exit enthält).

Ist der Standardwert nicht anwendbar, muss das Schlüsselwort REG13 angegeben werden. Sein Wert muss ein Vollwort sein, das auf den Sicherungsbereich zeigt, der zum Speichern aller Register beim Eingang in den Exit verwendet wurde.

MAPMAC = {*YES*|*NO*}

Ein optionales Schlüsselwort, das angibt, ob das Makro die erforderlichen Assemblerzuordnungsmakros generieren soll. Der Standardwert ist die Generierung dieser Zuordnungsmakros. Die folgenden Zuordnungsmakros sind für den EEEXIT-Code erforderlich: CVT, IEFJESCT, IEFJSSOB und IEFJSSIB. Der Exit IEFACTRT erfordert darüber hinaus das Makro IEFJMR.

SETUID = {*YES*|*NO*}

Ein optionales Schlüsselwort, das angibt, ob das Makro Code generieren soll, der die aktuelle Benutzer-ID in das Feld JMRUSEID stellt, wenn der Exit IEFUJI verwendet wird. Geben Sie YES an, um diesen Code zu generieren. Wenn Sie NO angeben (dies ist der Standardwert), wird das Feld JMRUSEID nicht aktualisiert. Es wird empfohlen, dass Sie YES angeben, wenn Sie die aktuelle Benutzer-ID zum Filtern von Schließereignissen für Dateien verwenden. Sie benötigen die folgenden Zuordnungsmakros, wenn Sie YES angeben: IHAPSA, IHAASCB, IHAASXB und IHAACEE.

Rückgabecodes

Die folgenden Rückgabecodes werden möglicherweise zur Assemblierungszeit generiert:

- 4 Eingabe ungültig, Warnungen überprüfen.
- 12 Nicht unterstützter Exit für das Schlüsselwort EXIT angegeben.

Nachrichten

Die folgenden Nachrichten werden möglicherweise zur Assemblierungszeit generiert:

- WARNING: EXIT NAME DIFFERS FROM CURRENT CSECT NAME
- WARNING: MAPMAC VALUE MAPMAC IS NOT RECOGNIZED
- EXIT NAME EXIT IS NOT SUPPORTED

SYS1.PARMLIB aktualisieren

In den folgenden Abschnitten sind die Aktualisierungen für SYS1.PARMLIB beschrieben, die für Ihre Umgebung erforderlich sind.

Subsysteme definieren

Sie müssen den Namen jedes neuen Subsystems für den Agenten für z/OS in dem Member mit der Tabelle für aktive Subsystemnamen von SYS1.PARMLIB definieren.

Beachten Sie Folgendes, wenn Sie den Subsystemnamen definieren:

- Der Name des Subsystems und der Name der gestarteten Task für den Agenten für z/OS müssen identisch sein.
- Instanzen des Agenten für z/OS, die mit demselben Dynamic Workload Broker verbunden sind, dürfen nicht denselben System-/STC-Namen aufweisen.
- Da die Subsystemnamen auf einer logischen Partition eindeutig sein müssen und da alle gestarteten Tasks des Agenten für z/OS denselben Namen wie die zugehörigen Subsysteme aufweisen müssen, müssen alle gestarteten Tasks auf jeder logischen Partition eindeutige Namen haben. (Das heißt, jede Instanz des Agenten für z/OS innerhalb eines z/OS-Image muss einen eindeutigen Subsystem-/STC-Namen haben.)

Aktualisieren Sie zum Definieren der Subsysteme das aktive Member IEFSSNnn in SYS1.PARMLIB. Schließen Sie Datensätze wie im folgenden Beispiel ein:

```
Subsystem definition record  
SUBSYS SUBNAME(subsystemname) INITRTN(modulname) INITPARM ('maxecsa,suffix')
```

Dabei gilt Folgendes:

subsystemname

Der Name, der einem Subsystem des Agenten für z/OS zugeordnet wird. Der Name muss 2 bis 4 Zeichen umfassen. Alle Subsystemnamen, die im Member IEFSSNnn von SYS1.PARMLIB definiert sind, müssen innerhalb eines GRS-Komplexes eindeutig sein. Außerdem müssen die Subsystemnamen innerhalb Ihres SYSPLEX (sowohl lokale als auch ferne Systeme) eindeutig sein. Der Name der gestarteten Task für einen Adressraum des Agenten für z/OS muss genau mit dem Namen des zugehörigen Subsystems übereinstimmen.

modulname

Der Name des Subsysteminitialisierungsmoduls, EELINITJ.

maxecsa

Definiert die maximale Kapazität des ECSA (Extended Common Service Area), die für Warteschlangen für Jobüberwachungsereignisse verwendet wird. Der Wert wird in Kilobyte ausgedrückt (1 KB entspricht 1024 Byte). Der Standardwert ist 4. Dies bedeutet, dass maximal 4 KB (4096 Byte) ECSA-Speicher für Warteschlangen für Jobüberwachungsereignisse benötigt wird. Der zulässige Maximalwert für MAXECSA ist 2816.

suffix

Der Modulnamensuffix für das Modul EELSSCM, das EELINITJ in den gemeinsamen Speicher lädt. EELSSCM ist das Kommunikationsmodul für das Subsystem. Das Suffix muss ein einzelnes Zeichen sein. Da der Name des mit dem Agenten für z/OS gelieferten Moduls EELINITJ lautet, geben Sie J als Suffixwert an. Wenn Sie kein Suffix bereitstellen, versucht EELINITJ, ein Modul mit dem Namen EELSSCMJ zu laden. Sie können den Namen eines Kommunikationsmoduls für das Subsystem auch im Schlüsselwort SSCM-NAME der Initialisierungsanweisung TWSOPTS angeben, um eine aktualisierte Version des Moduls vor einem geplanten IPL zu laden.

Weitere Informationen zu EELSSCM-Modulen finden Sie in „z/OS-Verbindungsbibliotheksdefinition aktualisieren“ auf Seite 22.

Das nächste Beispiel zeigt einen Datensatz, den Sie in das Member IEFSSN nn von SYS1.PARMLIB einfügen können:

```
/*Subsystem definition example*/
SUBSYS SUBNAME(ZAG3) INITRTN(eelinitj) INITPARM ('100,J')
```

Der Datensatz definiert ein Subsystem für den Agenten für z/OS mit dem Namen ZAG3. Sein Initialisierungsmodul ist EELINITJ. Die zugeordnete ECSA-Kapazität, 101104 Byte, reicht für 1136 Jobüberwachungsereignisse aus. Da der Suffixwert J angegeben ist, lädt EELINITJ das Modul EELSSCMJ.

MAXECSA-Werte berechnen

Der Agent für z/OS ordnet ECSA-Speicher für Jobüberwachungsereignisse in Blöcken mit 1424 Byte zu. Jeder Block entspricht 16 Ereignissen. Jeder Job erstellt mindestens sechs Ereignisse. In Tabelle 7 sind Beispiele für den benötigten Speicher, den tatsächlich zugeordneten Speicher und die jeweils speicherbaren Ereignisse für verschiedene MAXECSA-Werte aufgeführt.

Wenn Sie Werte für einen bestimmten MAXECSA-Wert berechnen möchten, der nicht in der Tabelle enthalten ist, verwenden Sie die folgende Methode:

- Angeforderter Speicherplatz = MAXECSA * 1024
- Blöcke = angeforderter Speicherplatz / 1424 (auf eine ganze Zahl abrunden)
- Zugeordneter Speicherplatz = Blöcke * 1424
- Speicherbare Ereignisse = Blöcke * 16

Tabelle 7. Beispiele für MAXECSA-Speicherwerte

MAXECSA-Wert	Kapazität des angeforderten MAXECSA-Speichers	Zugeordnete Blöcke des ECSA-Speichers (Byte)	Anzahl der speicherbaren Ereignisse
0	0	0 (0)	0
4	4096	2 (2848)	32

Tabelle 7. Beispiele für MAXECSA-Speicherwerte (Forts.)

MAXECSA-Wert	Kapazität des angeforderten MAXECSA-Speichers	Zugeordnete Blöcke des ECSA-Speichers (Byte)	Anzahl der speicherbaren Ereignisse
8	8192	5 (7120)	80
16	16384	11 (15664)	176
36	36864	25 (35600)	400
72	73728	51 (72624)	816
100	102400	71 (101104)	1136
200	204800	143 (203632)	2288
400	409600	287 (408688)	4592
500	512000	359 (511216)	5744

Wichtig:

- Ordnen Sie genügend ECSA-Speicher zu, damit die Jobüberwachungsereignisse nicht verloren gehen, wenn die Ereignisschreibersubtask des Agenten für z/OS nicht aktiv ist. Ist der Ereignisschreiber aktiv, ist die Anzahl der Ereignisse in der Warteschlange im ECSA fast immer 0. Ordnen Sie eine ausreichende ECSA-Kapazität für die erwartete Höchstdauer der Inaktivität des Ereignisschreibers zu. Beispielsweise können nach dem IPL eines z/OS-Systems Jobüberwachungsereignisse auftreten, bevor der Adressraum des Agenten für z/OS aktiv ist. Wenn Sie erwarten, dass während dieser Zeit maximal 50 Ereignisse eintreten, sollten Sie den MAXECSA-Wert 8 festlegen, wie aus der Tabelle hervorgeht. Wenn der Ereignisschreiber aktiv wird, werden die Ereignisse in der Warteschlange verarbeitet und aus dem ECSA entfernt.
Wenn Ereignisse verloren gehen, wird die Nachricht EELZ035E in das Nachrichtenprotokoll geschrieben.
- Der gesamte ECSA-Speicher wird im Bereich über 16 MB zugeordnet.

Lademodulbibliothek autorisieren

In diesem Abschnitt wird erläutert, wie die Lademodulbibliothek für den Agenten für z/OS aktiviert wird.

Sie müssen das aktive Member für Authorized Program Facility (IEAAPF nn oder PROG nn) aktualisieren, um die Lademodulbibliothek zu autorisieren. Jeder Datensatz mit Ausnahme des letzten endet mit einem Komma. Bei dem folgenden Beispiel wird vorausgesetzt, dass Sie die Lademodule des Agenten für z/OS in der Datei TWS.SEELLM00 installiert haben und dass diese Datei sich auf dem Datenträger ABC123 befindet. Fügen Sie zum Autorisieren dieser Bibliothek den folgenden Datensatz vor dem letzten Eintrag in IEAAPF nn ein:

```
TWS.SEELLM00 ABC123,
```

Sie können auch das Member PROG nn aktualisieren.

Beachten Sie, dass die Bibliotheken, die im Member IEAAPF nn oder PROG nn definiert sind, nur so lange autorisiert sind, wie sie sich auf dem angegebenen Datenträger befinden. Wird DFHSM auf Ihrem System verwendet, ändern Sie die DFHSM-Parameter so, dass die neue autorisierte Bibliothek nicht von DFHSM migriert wird.

SMF-Parameter aktualisieren

Die SMF-Parameter müssen aktualisiert werden, um die Exits zu aktivieren, die vom Agenten für z/OS für die Ereignisüberwachung verwendet werden.

Im Member SMFPRM nn sind Parameter für die System Management Facilities (SMF) definiert. Sie müssen überprüfen, ob im aktiven Member für SMF-Parameter, SMFPRM nn , angegeben ist, dass alle vom Agenten für z/OS verwendeten SMF-Exits für die Ereignisüberwachung aktiviert sind und dass die erforderlichen SMF-Datensätze erfasst werden. Ist dies nicht der Fall, müssen Sie das aktive Member für SMF-Parameter aktualisieren. Für die Ereignisüberwachung sind die folgenden SMF-Exits erforderlich:

IEFACTRT

Exit für Jobende und Schrittlende.

IEFUJI

Exit für Jobstart.

IEFU83

Exit für das Schreiben von Datensätzen. Für Jobausgabe verwendet.

Der Agent für z/OS verwendet die folgenden SMF-Datensatztypen:

26 Für die gesamte Jobüberwachung

30 Für die gesamte Jobüberwachung

Aktive Exits sind durch den Parameter EXITS der Schlüsselwörter SYS und SUBSYS definiert. Beispiel für diese Schlüsselwörter:

```
/*SYS and SUBSYS keywords*/
SYS(TYPE(6,26,30,60,62),EXITS(IEFACTRT,IEFUJI,IEFU83))
SUBSYS(STC,EXITS(IEFUJI,IEFACTRT,IEFU83))
SUBSYS(JESn,EXITS(IEFUJI,IEFACTRT,IEFU83))
```

Wichtig:

- JES n ist entweder JES2 oder JES3. Dieser Parameter bezieht sich nicht auf JES selbst, sondern auf Stapeljobs, die von JES verarbeitet werden. Unterdrücken Sie also nicht den Exitaufruf. Stellen Sie sicher, dass Sie nicht TYPE6=NO und TYPE26=NO in den Anweisungen JOBCLASS und STCCCLASS der JES2-Initialisierungsparameter angeben.
- Für die Installation ist es möglicherweise hilfreich, wenn Sie zwei SMFPRM nn -Member codieren: eines, in dem die Exits aktiv sind, und eines, in dem die Exits inaktiv sind. Dann können Sie den z/OS-Befehl SET SMF= nn verwenden, um Ihre aktuellen SMF-Parameter auf das neue Member umzuschalten. Durch Zurückschalten mit dem Befehl SET SMF= nn können Sie einen erneuten IPL vermeiden, falls ein Problem auftritt.
- Exits für SUBSYS STC sind nur erforderlich, wenn Sie auf demselben System auch IBM Workload Scheduler for z/OS ausführen. Wenn Sie nur den Agenten für z/OS ausführen, ist die folgende Zeile nutzlos:
SUBSYS(STC,EXITS(IEFUJI,IEFACTRT,IEFU83))

Verwenden Sie das PARMLIB-Member PROG nn , um Installationsexits anzugeben und deren Verwendung zu steuern. Mit PROG nn können Sie (beim IPL oder während das System aktiv ist) mehrere Exitroutinen zu Installationsexits zuordnen. Ziehen Sie in Erwägung, PROG nn neben SMFPRM nn für die Angabe von Exits zu verwenden, unabhängig davon, ob Sie diese Funktionen nutzen wollen.

Das folgende Beispiel zeigt, wie Sie SMF-Exits in einem PARMLIB-Member PROG xx angeben können. Angenommen, Sie geben Folgendes in SMFPRM nn an:
SYS(...EXITS(IEFACTRT,IEFUJI,IEFU83))

Sie müssen Folgendes hinzufügen, um die äquivalente Verarbeitung in PROG nn zu erzielen:

```
EXIT ADD EXITNAME(SYS.IEFACTRT) MODNAME(IEFACTRT)
EXIT ADD EXITNAME(SYS.IEFUJI) MODNAME(IEFUJI)
EXIT ADD EXITNAME(SYS.IEFU83) MODNAME(IEFU83)
```

Wenn Sie über PROG nn oder den Befehl SETPROG neue Exitroutinen zu SMF-Exits zuordnen, müssen Sie die folgenden Namenskonventionen befolgen:

- Für Exits, die im Schlüsselwort EXITS der Anweisung SYS in SMFPRM nn aufgelistet sind, hat jeder Exit den Namen SYS. $xxxx$ (wobei $xxxx$ einer der aufgelisteten Exits ist).
- Für Exits, die im Schlüsselwort EXITS der Anweisung SUBSYS von SMFPRM nn aufgelistet sind, hat jeder Exit den Namen SYS $zzzzz$. $xxxx$ (wobei $zzzz$ der Name des Subsystems und $xxxx$ einer der aufgelisteten Exits ist).

Wenn Sie in SYS1.PARMLIB zwei Member mit unterschiedlichen Namen definieren, beispielsweise PROG03 mit der Anweisung EXIT ADD EXITNAME(SYS.1EFACTRT) MODNAME(EQQACTR1), können Sie ohne IPL zu der Version EQQACTR1 umschalten, indem Sie den folgenden Befehl absetzen: /SET PROG=03.

Weitere Informationen zur Steuerung der Verwendung von Exits und Exitroutinen mit PROG nn finden Sie im Handbuch *IBM z/OS MVS Initialization and Tuning Reference*.

z/OS-Speicherauszugsoptionen aktualisieren

In diesem Abschnitt ist beschrieben, wie die z/OS-Speicherauszugsoptionen für den Agenten für z/OS aktualisiert werden.

Die Beispiel-JCL-Prozedur für einen Adressraum des Agenten für z/OS umfasst eine Datendefinitionsanweisung und eine Speicherauszugsdatei wird von dem Beispiel EELRETWT zugeordnet, das von EELINST erstellt wurde. SYSMDUMP ist das von der Serviceorganisation bevorzugte Speicherauszugsformat.

Stellen Sie sicher, dass die Speicherauszugsoptionen für SYSMDUMP auf Systemen, auf denen ein Adressraum für den Agenten für z/OS ausgeführt wird, die Angaben RGN, LSQA, TRT, CSA und GRSQ umfassen. Zum Anzeigen der aktuellen SYSMDUMP-Optionen setzen Sie den z/OS-Befehl DISPLAY DUMP,OPTIONS ab. Mit dem Befehl CHNGDUMP können Sie die SYSMDUMP-Optionen ändern. Beachten Sie, dass die Parameteränderung nur bis zur Ausführung des nächsten IPL in Kraft ist.

Zum Erstellen eines Speicherauszugs für einen Adressraum des Agenten für z/OS mit dem z/OS-Befehl DUMP sollten die SDUMP-Optionen RGN, LSQA, TRT, CSA und GRSQ angegeben werden. Möglicherweise empfiehlt es sich, diese Optionen als Systemstandardwert festzulegen.

z/OS-Verbindungsbibliotheksdefinition aktualisieren

In diesem Abschnitt sind die Schritte zur Aktualisierung der z/OS-Verbindungsbibliotheksdefinition beschrieben, die Sie ausführen sollten, wenn Sie den Agenten für z/OS in einer separaten Lademodulbibliothek installiert haben.

Wenn Sie den Agenten für z/OS in einer separaten Lademodulbibliothek installiert haben, sollten Sie diese Bibliothek im aktiven Member LNKLST nn definieren.

Wenn Sie die Lademodule in der Datei TWS.SEELLM00 installiert haben und diese Datei im Hauptkatalog katalogisiert ist, fügen Sie den folgenden Datensatz vor dem letzten Eintrag in das Member LNKSTnn ein, um diese Bibliothek zur Verbindungsbibliotheksverknüpfung hinzuzufügen:

```
Adding LINKLIB  
TWS.SEELLM00
```

Wenn Sie die Lademodulbibliothek von IBM Workload Scheduler for z/OS nicht im Member LNKSTnn definieren wollen, *müssen* Sie wie folgt vorgehen:

- Kopieren Sie die Tracker-Module, EQQINITM und EQQSSCMM, in eine Bibliothek in der z/OS-Verbindungsbibliotheksverknüpfung. EQQINITM wird von der Funktion für die Master-Scheduler-Initialisierung verwendet, wenn das z/OS-System über IPL gestartet wird. EQQINITM lädt dann EQQSSCMM in den gemeinsamen Speicher. EQQSSCMM ist ungefähr 23 KB groß und wird in den Bereich oberhalb des Grenzwerts von 16 MB gestellt. Sie müssen die Module bei jeder Aktualisierung durch die Verwaltung von IBM Workload Scheduler for z/OS erneut kopieren. Dies ist besonders für das Modul EQQSSCMM wichtig, das sich auf derselben Aktualisierungsstufe befinden muss wie der Rest des Codes von IBM Workload Scheduler for z/OS.
- Definieren Sie die Lademodulbibliothek von IBM Workload Scheduler for z/OS in einer Datendefinitionsanweisung STEPLIB in der JCL der gestarteten Task.

Produkt automatisch starten

Das Member COMMNDnn von SYS1.PARMLIB listet z/OS-Befehle auf, die bei der Systeminitialisierung automatisch ausgegeben werden. Zur Vermeidung von Verzögerungen beim Starten des Agenten für z/OS, wenn das z/OS-System gestartet wird, empfiehlt es sich möglicherweise, die Namen der gestarteten Task für den Agenten für z/OS in dieses Member aufzunehmen. Informationen zum Einschließen von Startbefehlen für einen Adressraum finden Sie im Handbuch *IBM z/OS MVS Initialization and Tuning Reference*.

RACF für die gestartete Task des Agenten für z/OS aktualisieren

In diesem Abschnitt ist die Definition des Agenten für z/OS für Ihr Sicherheitssystem beschrieben.

Wenn in Ihrer Installation Daten und Ressourcen vor unbefugter Verwendung geschützt sind, müssen Sie den Agenten für z/OS für Ihr Sicherheitssystem definieren. In diesem Abschnitt wird vorausgesetzt, dass RACF (Resource Access Control Facility) auf Ihrem z/OS-System installiert und aktiv ist. Der Abschnitt enthält eine Beschreibung der Aktivitäten, die Sie ausführen müssen, um die Sicherheitsumgebung für den Agenten für z/OS zu definieren und zu aktivieren.

RACF steuert die Interaktion zwischen Benutzern und Ressourcen. Über RACF-Profilen definieren Sie Ressourcen und die Zugriffsebenen, die den Benutzern für diese Ressourcen erteilt werden. Unter einem Benutzer versteht man eine alphanumerische Benutzer-ID, die RACF dem Benutzer zuordnet.

Der Agent für z/OS benötigt Zugriff auf z/OS-Ressourcen, um Arbeit zu terminieren. Die Benutzer-ID, die dem Agenten zugeordnet ist, kann aus den folgenden Quellen abgerufen werden:

- Adressraum des Agenten für z/OS, der auf Dateien zugreift, die für die terminierte Arbeit verwendet werden, und der Arbeit übergibt und JES-Befehle absetzt.

- Parameter USER auf der JOB-Karte eines Stapeljobs, der übergeben werden soll.

Benutzer-ID des Adressraums steuern

Da der Agent für z/OS als gestartete Task ausgeführt wird, müssen Sie den katalogisierten Prozedurnamen einem RACF-Benutzer zuordnen, der über ausreichende Berechtigungen verfügt. Die Benutzer-ID muss in der Ressourcenklasse STARTED definiert sein.

Benutzer-ID der übergebenen Jobs steuern

Der Agent für z/OS kann zwei Typen von Jobs an JES übergeben:

- Normale Produktionsjobs, die von einem IBM Workload Scheduler-Plan übergeben werden.
- Ad-hoc-Jobs, die Sie mithilfe von Dynamic Workload Console oder Conman direkt übergeben können.

Der Agent übergibt Produktions- und Ad-hoc-Jobs an den internen Leser, wenn alle Voraussetzungen erfüllt sind. Die einem Job zugeordnete Berechtigung können Sie wie folgt festlegen:

- Sie können Arbeit mit der Berechtigung des Adressraums des Agenten für z/OS übergeben. Der Job erhält dieselbe Berechtigung wie der Agent für z/OS.
- Sie können ein Kennwort in die JCL einschließen, um die Berechtigung eines bestimmten Benutzers weiterzugeben.

Dateien schützen

Für die grundlegende Datensicherheit sollten Sie den Zugriff auf die folgenden Produktdateien einschränken:

- Interner Leser (EELBRDS)
- Diagnosedateien (EELDUMP und SYSDUMP)
- Ereignisdateien (EELEVDS und EELHTDS)
- Servicedatei (EELHTREF)
- Nachrichtenbibliothek (EELMLIB)
- Nachrichtenprotokoll (EELMLOG)
- Parameterbibliothek (EELPARM)

Darüber hinaus müssen die Mitarbeiter in der Softwareunterstützung in der Lage sein, Fehler zu beheben und Dateien zu reorganisieren. Es empfiehlt sich, diesen Mitarbeitern Änderungszugriff auf alle Produktdateien zu erteilen.

SSL-Umgebung einrichten

In diesem Abschnitt ist beschrieben, wie SSL-Schutz für die Verbindung zwischen Ihrem Agenten für z/OS und IBM Workload Scheduler eingerichtet wird.

Zum Bereitstellen von SSL-Sicherheit für die HTTP-Verbindung zwischen dem Agenten für z/OS und dem Dynamic Workload Broker von IBM Workload Scheduler führen Sie folgende Schritte mit der Initialisierungsanweisung HTTPOPTS aus:

- Setzen Sie das Schlüsselwort SSL und/oder TDWBSSL auf Yes.
- Geben Sie Werte für die zu SSL gehörigen Schlüsselwörter an.

- Wählen Sie für SSL aktivierte Ports für die beiden Schlüsselwörter für die Verbindung aus: PORTNUMBER (für den Agenten) und TDWBPORTNUMBER (für Dynamic Workload Broker).

Sicherheitszertifikate verwenden

Wenn Sie den Agenten installieren, werden die folgenden Standardsicherheitszertifikate automatisch in der Bibliothek SEELDATA gespeichert:

EELCERCL

Das Sicherheitszertifikat für den HTTP-Client (Dynamic Workload Broker).

EELCERSR

Das Sicherheitszertifikat für den HTTP-Server (Agent für z/OS).

Sie müssen entscheiden, ob Sie diese Standardzertifikate verwenden oder eigene Zertifikate erstellen; es sei denn, Sie haben diese Entscheidung bereits bei der Ausführung der Installationshilfe EELINST getroffen (Anzeige 4/5) oder Sie verwenden bereits SSL mit IBM Workload Scheduler for z/OS. In beiden Fällen müssen Sie sie manuell in Ihr Sicherheitssystem importieren. Wenn Sie RACF verwenden, steht Ihnen der Beispieljob EELRCERT zur Verfügung, der die Zertifikate importiert. Stellen Sie sicher, dass Sie zum Ausführen dieses Jobs die Benutzer-ID verwenden, die RACF der gestarteten Task des Agenten für z/OS zuordnet.

Der Job EELRCERT führt die folgenden Schritte aus:

- Er kopiert die Zertifikate EELCERCL und EELCERSR in temporäre sequenzielle Dateien.
- Er importiert EELCERCL und EELCERSR in RACF.
- Er löscht die temporären sequenziellen Dateien.
- Er erstellt den SAF-Schlüsselring, der zum Verbinden der importierten Zertifikate verwendet wird.
- Er aktualisiert die RACF-Datenbank mit den neuen Zertifikaten und dem neuen Schlüsselring.

SYS1.PROCLIB aktualisieren

In diesem Abschnitt ist die Definition einer JCL-Prozedur für den Adressraum des Agenten für z/OS beschrieben.

Sie müssen eine JCL-Prozedur oder einen Stapeljob für den Adressraum des Agenten für z/OS definieren.

Die Installationshilfe EELINST generiert die folgenden Member in der Ausgabebibliothek, die Sie im Dialog Create customized sample jobs angeben haben, um Sie bei dieser Aufgabe zu unterstützen:

EELAGT

Beispielprozedur für die gestartete Task des Agenten.

EELAGTP

Beispielparameter für die gestartete Task des Agenten.

Diese Member enthalten JCL für die gestartete Task, die mit den von Ihnen in diesem Dialog eingegebenen Werten angepasst ist. Passen Sie diese Member weiter für die Dateien an, die Sie benötigen. Sie können auch ein Member aus der Bibliothek SEELSAMP in eine Ihrer eigenen Bibliotheken kopieren und manuell anpassen.

Wenn Sie eine neue Bibliothek für Ihre Prozeduren für die gestartete Task des Agenten für z/OS erstellen, vergessen Sie nicht, die Bibliothek in der JES-PROCLIB-Verkettung anzugeben. Anschließend müssen Sie JES erneut starten, um die neue Bibliothek einzuschließen.

Erforderliche Dateien

Schließen Sie die folgenden erforderlichen Dateien in Ihre JCL-Prozedur ein:

Tabelle 8. Erforderliche Dateien für den Agenten für z/OS

Datendefinitionsname	Definiert
EELBRDS	Internen JES-Leser
EELVDS	Ereignisdatei für die Funktion für das Übergabepunktverfahren und für die Ereignisschreibertask
EELHTDS	Ereignisdatei für das Speichern von Ereignissen, die aus Dynamic Workload Broker stammen
EELHTREF	Servicedatei für die Verarbeitung von JCLs <i>nach Verweis</i> und für die Variablensubstitution
EELMLIB	Nachrichtenbibliothek
EELMLOG	Ausgabenachrichtenprotokoll
EELPARM	Parameterbibliothek

Optionale Dateien

Die folgende Tabelle enthält die Dateien, die Sie optional in Ihre JCL-Prozedur einschließen können. Geben Sie diese Dateien nur an, wenn Sie die zugehörige Funktion verwenden wollen.

Tabelle 9. Optionale Dateien für den Agenten für z/OS

Datendefinitionsname	Definiert
EELDUMP	Speicherausgangsausgabe für die Diagnose.
STDENV	Enthält Umgebungsvariablen. Der Datendefinitionsname STDENV kann auf ein sequenzielles DS- oder ein PDS-Member (beispielsweise ein Member von PARMLIB) zeigen, in dem Sie die Umgebungsvariablen zur Initialisierung von Language Environment definieren können. STDENV muss das Format F oder FB und eine Datensatzlänge größer oder gleich 80 aufweisen. In dieser Datei bzw. in diesem Member können Sie Ihre Umgebungsvariablen durch die Angabe VARNAME=wert definieren. In jeder Zeile können Sie nur 1 Variable angeben; Zeichen nach Spalte 71 werden ignoriert. Wenn Sie mehr als 71 Zeichen benötigen, können Sie ein beliebiges Zeichen in Spalte 72 einfügen und die Eingabe in der nächsten Zeile fortsetzen (das Zeichen in Spalte 72 wird ignoriert).
STEPLIB	Lademodulbibliothek.

Tabelle 9. Optionale Dateien für den Agenten für z/OS (Forts.)

Datendefinitionsname	Definiert
SYSMDUMP	Speicherauszugsdatei.

Installation abschließen

In diesem Abschnitt sind die abschließenden Schritte beschrieben, die Sie ausführen müssen, um die Installation des Agenten zu Ende zu führen und zu überprüfen.

Gehen Sie wie folgt vor, wenn Sie die Installationstasks für den Agenten ausgeführt haben:

1. Führen Sie einen IPL auf jedem System aus, auf dem Sie den Agenten installiert haben.
2. Überprüfen Sie die Installation.

Führen Sie die folgenden Tasks aus, um den Agenten zu überprüfen:

1. Sicherstellen, dass Sie alle erforderlichen Installationstasks ausgeführt haben.
2. Agent starten und Verbindung zu Dynamic Workload Broker prüfen.
3. Nachrichtenprotokoll (EELMLOG) prüfen.
4. Überprüfen, ob Überwachungsereignisse in der Ereignisdatei (EELEVDS) erstellt werden.
5. Problembestimmung für Überwachungsereignisse ausführen, wenn Ereignisse in der Ereignisdatei fehlen.

Agenten starten und Verbindung prüfen

Als ersten Schritt zur Überprüfung des Installationserfolgs starten Sie den Agenten und prüfen, ob er eine Verbindung zu dem angegebenen Dynamic Workload Broker von IBM Workload Scheduler herstellt.

Zum Starten des Agenten für z/OS verwenden Sie den z/OS-Befehl START mit dem Namen des Subsystems oder der gestarteten Task, den Sie in den EELINST-Anzeigen für den Agenten definiert haben.

Wenn der Agent startet und erfolgreich eine Verbindung zu Dynamic Workload Broker herstellt, wird die Nachricht Der Agent ist für E-Business geöffnet auf der z/OS-Konsole angezeigt.

Wenn der Agent zum ersten Mal startet und erfolgreich eine Verbindung zu dem Dynamic Workload Broker herstellt, dessen Hostname und Port Sie in der Initialisierungsanweisung HTTOPTS angegeben haben, wird er automatisch in der IBM Workload Scheduler-Datenbank mit dem folgenden Workstationnamen definiert:

subsystemname-systemname

Dabei gilt Folgendes:

subsystemname

Dies ist der Name der gestarteten Task von z/OS, die den Agenten startet.

systemname

Dies ist der Name des z/OS-Systems.

Sie können diesen Workstationnamen jetzt verwenden, um eine Workload für z/OS zu entwerfen, zu übergeben und zu überwachen. Weitere Informationen finden Sie in „Computer- und Workstationname des Agenten“ auf Seite 51.

Sicherstellen, dass alle Installationstasks ausgeführt wurden

Stellen Sie sicher, dass alle Installationstasks ausgeführt wurden.

Stellen Sie sicher, dass Sie alle Installationstasks ausgeführt haben, die erforderlich sind, damit Ihr Agent für z/OS ordnungsgemäß ausgeführt wird. D. h., Sie sollten die folgenden Tasks ausgeführt haben:

- Entsprechende Prozeduren für das Subsystem des Agenten für z/OS ausführen, das Sie installieren.
- Erforderliche JES- und SMF-Exits installieren und überprüfen, ob sie aktiv sind.
- JCL-Prozedur für den Tracker erstellen.
- Erforderliche Dateien zuordnen.
- Sicherheitszugriff einrichten, damit das Subsystem auf die Dateien zugreifen kann.
- Initialisierungsanweisungen in der Parameterbibliothek (EELPARM) angeben.

Nachrichtenprotokoll prüfen

In diesem Abschnitt ist die Überprüfung des Nachrichtenprotokolls beschrieben.

Nach dem Start des Agenten prüfen Sie das Nachrichtenprotokoll:

- Prüfen Sie, ob der Rückkehrcode für alle Initialisierungsoptionen 0 ist (Nachricht EELZ016I).
- Stellen Sie sicher, dass alle erforderlichen Subtasks aktiv sind.
 - Die Tasks für den Datenrouter und die Übergabe werden immer gestartet. Die folgenden Nachrichten sollten angezeigt werden:

```
EELZ005I SUBTASK DATA ROUTER IS BEING STARTED
EELF001I DATA ROUTER TASK INITIALIZATION IS COMPLETE

EELZ005I SUBTASK JOB SUBMIT IS BEING STARTED
EELSU01I THE SUBMIT TASK HAS STARTED
```
 - Überprüfen Sie außerdem, ob der Agent einen Ereignisschreiber gestartet hat. Die folgenden Nachrichten sollten angezeigt werden:

```
EELZ005I SUBTASK EVENT WRITER IS BEING STARTED
EELW065I EVENT WRITER STARTED
```
- Prüfen Sie Fehlermeldungen.

Wichtig: Beim ersten Start des Ereignisschreibers formatiert dieser die Ereignisdatei. Ignorieren Sie den Abbruchcode SD37, der während des Formatierungsprozesses ausgegeben wird.

- Überprüfen Sie, ob das Protokoll vollständig ist. Setzen Sie dazu einen Testbefehl MODIFY wie den folgenden ab: F ss-name,xx. Nach der Verarbeitung des Befehls wird die Nachricht EELZ049E in das Protokoll geschrieben. Ist diese Nachricht der letzte Eintrag im Protokoll, bedeutet dies, dass das Protokoll ordnungsgemäß funktioniert.

Überwachungseignisse überprüfen

In diesem Abschnitt ist beschrieben, wie geprüft wird, ob der Agent Informationen zu Überwachungseignissen erfasst und in die Ereignisdatei (EELEVDS) schreibt.

Die Jobüberwachung funktioniert nur dann ordnungsgemäß, wenn der Agent für z/OS Informationen zu allen Statusänderungen der Jobs empfängt, die er übergeben hat. Die Jobüberwachung ruft diese Informationen aus SMF- und JES-Exits ab. Diese Exits stellen die erforderlichen Informationen zusammen und ein Exitdatensatz wird über ECSA-Puffer der Ereignisschreiberwarteschlange des Agenten hinzugefügt. Die Ereignisschreiberwarteschlange ist auch aktiv, wenn der Agent nicht aktiv ist.

Ereignisschreiber

Der Ereignisschreiber entfernt das Ereignis aus seiner Warteschlange und erstellt einen Ereignisdatensatz, der in eine Ereignisdatei geschrieben wird. Der Ereignisschreiber leitet das Ereignis auch weiter, wenn er mit einer Ereignisleserfunktion gestartet wurde.

Ereignisdatei

Die Ereignisdatei wird benötigt, um Unterschiede zwischen der Geschwindigkeit der Generierung und der Verarbeitung von Ereignissen auszugleichen und um zu verhindern, dass Ereignisse verloren gehen, wenn der Agent für z/OS erneut gestartet werden muss.

Das erste Byte in einem Exitsatz ist A, wenn das Ereignis auf einem JES2-System erstellt wird, oder B, wenn das Ereignis auf einem JES3-System erstellt wird. Dieses Byte befindet sich an der Position 21 des Standardereignissatzes bzw. an der Position 47 eines Fortsetzungereignisses (Typ N). Die Bytes 2 und 3 in dem Exitdatensatz definieren den Ereignistyp. In Tabelle 10 werden die Ereignistypen gezeigt, die vom Agenten für z/OS generiert werden.

Tabelle 10. Vom Agenten für z/OS generierte Ereignisse

Ereignistyp	Beschreibung	Generiert von...
KJ1	Jobübergabeereignis. Ein Job wurde vom Agenten für z/OS an JES übergeben.	Agent
A1	Leserereignis. Ein Job ist in das JES2-System eingetreten.	JES2-Exits EXIT7 und EXIT51
B1	Leserereignis. Ein Job ist in das JES3-System eingetreten.	JES3-Exit IATUX09
A2 oder B2	Jobstartereignis. Die Ausführung eines Jobs hat begonnen.	SMF-Exit IEFUJ1
A3J oder B3J	Jobendereignis. Die Ausführung eines Jobs wurde fertig gestellt.	SMF-Exit IEFACRT
A3P	Jobabschlussereignis. Ein Job wurde den JES2-Ausgabewarteschlangen hinzugefügt.	JES2-Exit EXIT7
B3P	Jobabschlussereignis. Ein Job wurde den JES3-Ausgabewarteschlangen hinzugefügt.	JES3-Exit IATUX19
A5	Bereinigungsereignis. Sämtliche Ausgabe für einen Job wurde aus dem JES2-System gelöscht.	JES2-Exit EXIT7
B5	Bereinigungsereignis. Sämtliche Ausgabe für einen Job wurde aus dem JES3-System gelöscht.	SMF-Exit IEFU83

Wird nach der ersten Jobübergabe einer dieser Ereignistypen nicht in der Ereignisdatei (EELEVDS) erstellt, muss ein Fehler behoben werden, bevor der Agent für z/OS im Produktionsmodus gestartet wird.

Führen Sie die folgenden Aktionen aus, um zu überprüfen, ob Ereignisse auf Ihrem System erstellt werden:

1. Führen Sie einen Job über Conman oder Dynamic Workload Console aus:
 - a. Übergeben Sie einen Job wie den folgenden und stellen Sie sicher, dass die Ausgabe in eine nicht angehaltene Ausgabeklasse geschrieben wird:

```
Testjob
//VERIFY1 JOB STATEMENT PARAMETERS
//VERIFY EXEC PGM=IEBGENER
//*
//SYSPRINT DD DUMMY
//SYSUT2 DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD *
        SAMPLE TEST OUTPUT STATEMENT 1
//*
```

- b. Überprüfen Sie, ob der Job ausgeführt, ausgegeben und bereinigt wurde.
 - c. Zeigen Sie die Datei EELEVDS mit der ISPF/PDF-Anzeigefunktion an. Die folgenden Ereignisse sind in der Ereignisdatei enthalten:
 - Ereignis des Typs KJ1
 - Ereignis des Typs A1 (für JES2) oder Ereignis des Typs B1 (für JES3)
 - Ereignis des Typs A2 (für JES2) oder Ereignis des Typs B2 (für JES3)
 - Ereignis des Typs A3J (für JES2) oder Ereignis des Typs B3J (für JES3)
 - Ereignis des Typs A3P (für JES2) oder Ereignis des Typs B3P (für JES3)

Problembestimmung für Überwachungsereignisse ausführen

In diesem Abschnitt ist beschrieben, wie die Problembestimmung für Überwachungsereignisse ausgeführt wird, wenn Ereignisse in der Ereignisdatei fehlen.

Die Problembestimmung ist davon abhängig, welches Ereignis fehlt. In der folgenden Tabelle enthält die erste Spalte den fehlenden Ereignistyp und die zweite Spalte die auszuführende Aktion.

Tabelle 11. Typen fehlender Ereignisse und entsprechende Aktionen zur Problembestimmung

Typ	Aktionen zur Problembestimmung
Alle	<ol style="list-style-type: none"> 1. Überprüfen Sie in der Datei EELMLOG, ob der Ereignisschreiber erfolgreich gestartet wurde. 2. Überprüfen Sie, ob die Definition des Datendefinitionsnamens EELEVDS in der Prozedur für die gestartete Task des Agenten für z/OS korrekt ist (d. h., ob die Ereignisse in die korrekte Datei geschrieben werden). 3. Überprüfen Sie, ob die erforderlichen Exits installiert wurden. 4. Überprüfen Sie, ob das Member IEFSSNnm von SYS1.PARMLIB ordnungsgemäß aktualisiert wurde und ob ein IPL des z/OS-Systems nach der Aktualisierung ausgeführt wurde.
KJ1	Überprüfen Sie, ob das Subsystem des Agenten für z/OS ordnungsgemäß definiert wurde.

Tabelle 11. Typen fehlender Ereignisse und entsprechende Aktionen zur Problembestimmung (Forts.)

Typ	Aktionen zur Problembestimmung
A1	<p>Falls das Ereignis A3P ebenfalls fehlt:</p> <ol style="list-style-type: none"> 1. Überprüfen Sie, ob die Version des Agenten für z/OS für die Routinen der JES2-Exits 7 und 51 ordnungsgemäß installiert wurde. Verwenden Sie die JES-Befehle \$T EXIT(7) und \$T EXIT(51) oder \$DMODULE(TWSEXIT7) und \$DMODULE(TWSXIT51). 2. Überprüfen Sie, ob die JES2-Initialisierungsdatei eine Anweisung LOAD und eine Anweisung EXIT7 für die Version des Agenten für z/OS für den JES2-Exit 7 (TWSEXIT7) enthält. 3. Überprüfen Sie, ob der Exit einer Lademodulbibliothek hinzugefügt wurde, die für JES2 erreichbar ist, und ob JES2 nach dieser Aktion erneut gestartet wurde. <p>Falls das Ereignis A3P in der Ereignisdatei vorhanden ist, fordern Sie Unterstützung für die Programmierung bei einem IBM® Ansprechpartner an.</p>
B1	<ol style="list-style-type: none"> 1. Überprüfen Sie, ob die Version des Agenten für z/OS für die Routine des JES3-Exits IATUX09 ordnungsgemäß installiert wurde. 2. Überprüfen Sie, ob der Exit einer Lademodulbibliothek hinzugefügt wurde, auf die JES3 zugreifen kann. <p>Überprüfen Sie, ob JES3 gestartet wurde.</p>
A2 oder B2	<ol style="list-style-type: none"> 1. Überprüfen Sie, ob die Ausführung des Jobs, für den kein Ereignis des Typs A2 erstellt wurde, gestartet wurde. Ein Ereignis des Typs A2 wird nicht für einen Job erstellt, der aufgrund von JCL-Fehlern aus dem System gelöscht wird. 2. Überprüfen Sie, ob der Exit IEFUJI ordnungsgemäß installiert wurde: <ol style="list-style-type: none"> a. Überprüfen Sie, ob in dem Member für SMF-Parameter, SMFPRMm, in der Datei SYS1.PARMLIB angegeben ist, dass der Exit IEFUJI aufgerufen werden soll. b. Stellen Sie sicher, dass der Exit IEFUJI nicht durch einen Bedienerbefehl inaktiviert wurde. c. Überprüfen Sie, ob die korrekte Version von IEFUJI aktiv ist. Falls LPALIB in SYS1.PARMLIB als Verkettung mehrerer Bibliotheken definiert ist, verwendet z/OS das erste gefundene Modul IEFUJI. d. Überprüfen Sie, ob die Bibliothek, die dieses Modul enthält, durch die Version des Agenten für z/OS für IEFUJI aktualisiert wurde und ob nach dieser Änderung für das z/OS-System ein IPL ausgeführt wurde.
A3J oder B3J	<ol style="list-style-type: none"> 1. Überprüfen Sie, ob der Exit IEFACTRT ordnungsgemäß installiert wurde. 2. Überprüfen Sie, ob in dem Member für SMF-Parameter, SMFPRMm, in der Datei SYS1.PARMLIB angegeben ist, dass der Exit IEFACTRT aufgerufen werden soll. 3. Stellen Sie sicher, dass der Exit IEFACTRT nicht durch einen Bedienerbefehl inaktiviert wurde. 4. Überprüfen Sie, ob die korrekte Version von IEFACTRT aktiv ist. Falls LPALIB in SYS1.PARMLIB als Verkettung mehrerer Bibliotheken definiert ist, verwendet z/OS das erste gefundene Modul IEFACTRT. 5. Überprüfen Sie, ob diese Bibliothek durch die Version des Agenten für z/OS für IEFACTRT aktualisiert wurde und ob nach dieser Änderung für das z/OS-System ein IPL ausgeführt wurde.

Tabelle 11. Typen fehlender Ereignisse und entsprechende Aktionen zur Problembestimmung (Forts.)

Typ	Aktionen zur Problembestimmung
A3P	Falls auch Ereignisse des Typs A1 fehlen, befolgen Sie die Prozedur, die für Ereignisse des Typs A1 beschrieben ist. Falls Ereignisse des Typs A1 nicht fehlen, fordern Sie Unterstützung für die Programmierung bei einem IBM Ansprechpartner an.
B3P	<ol style="list-style-type: none"> 1. Überprüfen Sie, ob die Version des Agenten für z/OS für die Routine des JES3-Exits IATUX19 ordnungsgemäß installiert wurde. 2. Überprüfen Sie, ob der Exit einer Lademodulbibliothek hinzugefügt wurde, auf die JES3 zugreifen kann. 3. Überprüfen Sie, ob JES3 erneut gestartet wurde.
A5	<ol style="list-style-type: none"> 1. Überprüfen Sie, ob JES2 den Job gelöscht hat, für den kein Ereignis des Typs A5 erstellt wurde. 2. Stellen Sie sicher, dass Sie nicht TYPE26=NO in den Anweisungen JOBCLASS und STCCCLASS der JES2-Initialisierungsparameter angegeben haben. 3. Falls auch Ereignisse des Typs A1 fehlen, befolgen Sie die Prozedur, die für Ereignisse des Typs A1 beschrieben ist. 4. Falls Ereignisse des Typs A1 nicht fehlen, fordern Sie Unterstützung für die Programmierung bei einem Produktansprechpartner an.
B5	Überprüfen Sie, ob JES3 den Job gelöscht hat, für den kein Ereignis des Typs B5 erstellt wurde.

Empfehlungen für die Zuordnung der Jobbibliotheksdatei (EELJBLIB)

Die Jobbibliotheksdatei enthält die JCL für die Jobs, die der Agent für z/OS übergibt. Sie ist für den Agenten erforderlich. Die Datei wird zuerst zugeordnet, wenn Sie die Installationshilfe EELINST ausführen.

Wenn Sie später mehr Jobbibliotheksdateien zuordnen müssen, geben Sie PDSE-Dateien (PDSE, erweiterte partitionierte Datei) den Vorzug.

Wenn Sie PDS-Dateien (PDS, partitionierte Datei) zuordnen müssen, ordnen Sie die Jobbibliotheksdatei nur mit Zuordnung des primären Bereichs zu. Wenn eine sekundäre Bereichszuordnung definiert ist und die Bibliothek in eine Erweiterung geht, wenn der Agent aktiv ist, müssen Sie den Agenten stoppen und erneut starten. Außerdem dürfen Sie Member in dieser PDS-Datei nicht komprimieren. Verwenden Sie zum Beispiel nicht den ISPF-Befehl PACK ON, da der Agent die ISPF-Services nicht zum Lesen der Datei verwendet.

Anpassungsparameter

Die Parameter, die für die Anpassung des Agenten für z/OS erforderlich sind, werden in Form von Initialisierungsanweisungen bereitgestellt. In diesem Abschnitt sind alle Initialisierungsanweisungen dokumentiert, die für den Agenten benötigt werden.

Die Initialisierungsanweisungen, die zum Ausführen Ihrer gestarteten Task des Agenten für z/OS erforderlich sind, werden mit funktionierenden Standardwerten erstellt, wenn Sie den Schritt **Define initialization statements in the parameters library** der Installationshilfe EELINST ausführen. Zu diesem Zeitpunkt haben Sie

die Option, die Standardwerte beizubehalten und später zu ändern oder die Standardwerte sofort zu ändern, bevor das zugehörige Member in der Parameterbibliothek erstellt wird. Wenn Sie die Werte später ändern wollen, müssen Sie die Datei PARMLIB des Agenten bearbeiten und den Agenten danach erneut starten.

Der Agent erfordert die folgenden Initialisierungsanweisungen:

EWTROPTS

Definiert Laufzeitoptionen für die Ereignisschreibertask.

EXITS Definiert die Exitrichtlinie für den Agenten.

HTTPOPTS

Definiert Optionen zum Herstellen der Verbindung zum Dynamic Workload Broker.

TWSOPTS

Definiert allgemeine Laufzeitoptionen für den Agenten.

Laufzeitoptionen für den Ereignisschreiber angeben

In diesem Abschnitt ist die Initialisierungsanweisung EWTROPTS beschrieben.

Verwenden Sie die Initialisierungsanweisung EWTROPTS, um Laufzeitoptionen für die Ereignisschreibertask anzugeben.

EWTROPTS

►► EWTROPTS—RETCODE—(—LAST—
HIGHEST—) —————►►

Parameter

RETCODE = (LAST|HIGHEST)

Definiert, wie der Ereignisschreiber einen Rückkehrcode für den Ereignisdatensatz für Jobende (A3J) erstellt. Wenn Sie HIGHEST angeben, erstellt der Ereignisschreiber einen Ereignisdatensatz mit dem höchsten Rückkehrcode aller ausgeführten Abschnitte. Wenn Sie LAST angeben, erstellt der Ereignisschreiber einen Ereignisdatensatz mit dem Rückkehrcode des zuletzt ausgeführten Abschnitts.

Der Standardwert ist LAST.

Wichtig: In z/OS 1.13 und höher kann der Parameter JOBRC in den JOB-Kartenanweisungen der JCL hinzugefügt werden. Wenn der Parameter JOBRC in der JCL-Jobkarte mit dem Wert MAXRC oder LASTRC angegeben wird, wird der Jobbeendigungscode, der von RETCODE bestimmt wird, durch den Wert von JOBRC überschrieben. Wenn der Parameter JOBRC mit dem Wert STEP angegeben wird, wird er vom Agenten für z/OS ignoriert und der Jobbeendigungscode, der im Ereignisdatensatz protokolliert wird, ist derjenige, der durch den RETCODE-Wert festgelegt wird.

Die Schlüsselwortwerte sind gültig, bis Sie einen anderen Wert angeben und den Agenten erneut starten.

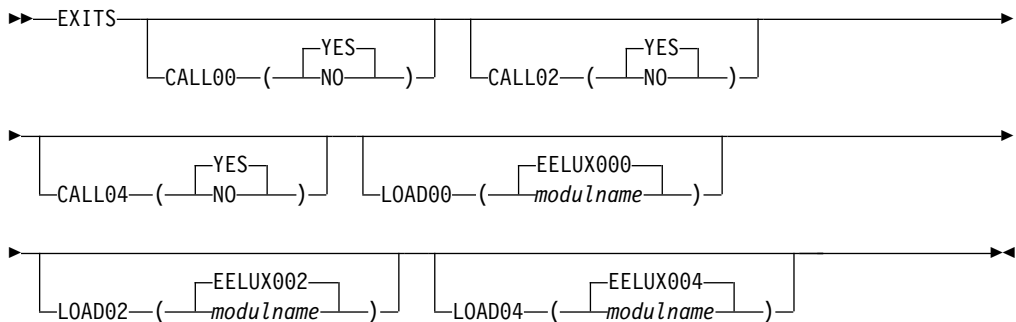
Exitrichtlinie für den Agenten angeben

In diesem Abschnitt ist die Initialisierungsanweisung EXITS beschrieben.

Diese Anweisung definiert Exitoptionen für den Agenten für z/OS. Sie wird auf die Exitprogramme, EELUX000, EELUX002 und EELUX004 angewendet, die der Agent verwendet. Mit der Anweisung EXITS können Sie verhindern, dass der Agent versucht, einen bestimmten Exit zu laden, oder den Standardnamen des Lademoduls ändern.

Weitere Informationen zu diesen Exitprogrammen finden Sie in „Exit EELUX000 (Start/Stop) konfigurieren“ auf Seite 43, „Exit EELUX002 (Lesen in der Jobbibliothek) konfigurieren“ auf Seite 44 und „Exit EELUX004 (Ereignisfilterung) konfigurieren“ auf Seite 48.

EXITS



Parameter

CALL00 = (YES|NO)

Gibt an, ob Exit EELUX000 geladen werden soll. Der Exitname ist entweder EELUX000 oder die Alternative, die durch das Schlüsselwort LOAD00 angegeben ist.

Der Standardwert ist YES.

CALL02 = (YES|NO)

Gibt an, ob Exit EELUX002 geladen werden soll. Der Exitname ist entweder EELUX002 oder die Alternative, die durch das Schlüsselwort LOAD02 angegeben ist.

Der Standardwert ist YES.

CALL04 = (YES|NO)

Gibt an, ob Exit EELUX004 geladen werden soll. Der Exitname ist entweder EELUX004 oder die Alternative, die durch das Schlüsselwort LOAD04 angegeben ist.

Der Standardwert ist YES.

LOAD00 = (EELUX000|modulname)

Gibt ein alternatives Lademodul an, das anstelle des Standardexits mit dem Namen EELUX000 aufgerufen wird.

LOAD02 = (EELUX002|modulname)

Gibt ein alternatives Lademodul an, das anstelle des Standardexits mit dem Namen EELUX002 aufgerufen wird.

LOAD04 = (EELUX004|modulname)

Gibt ein alternatives Lademodul an, das anstelle des Standardexits mit dem Namen EELUX004 aufgerufen wird.

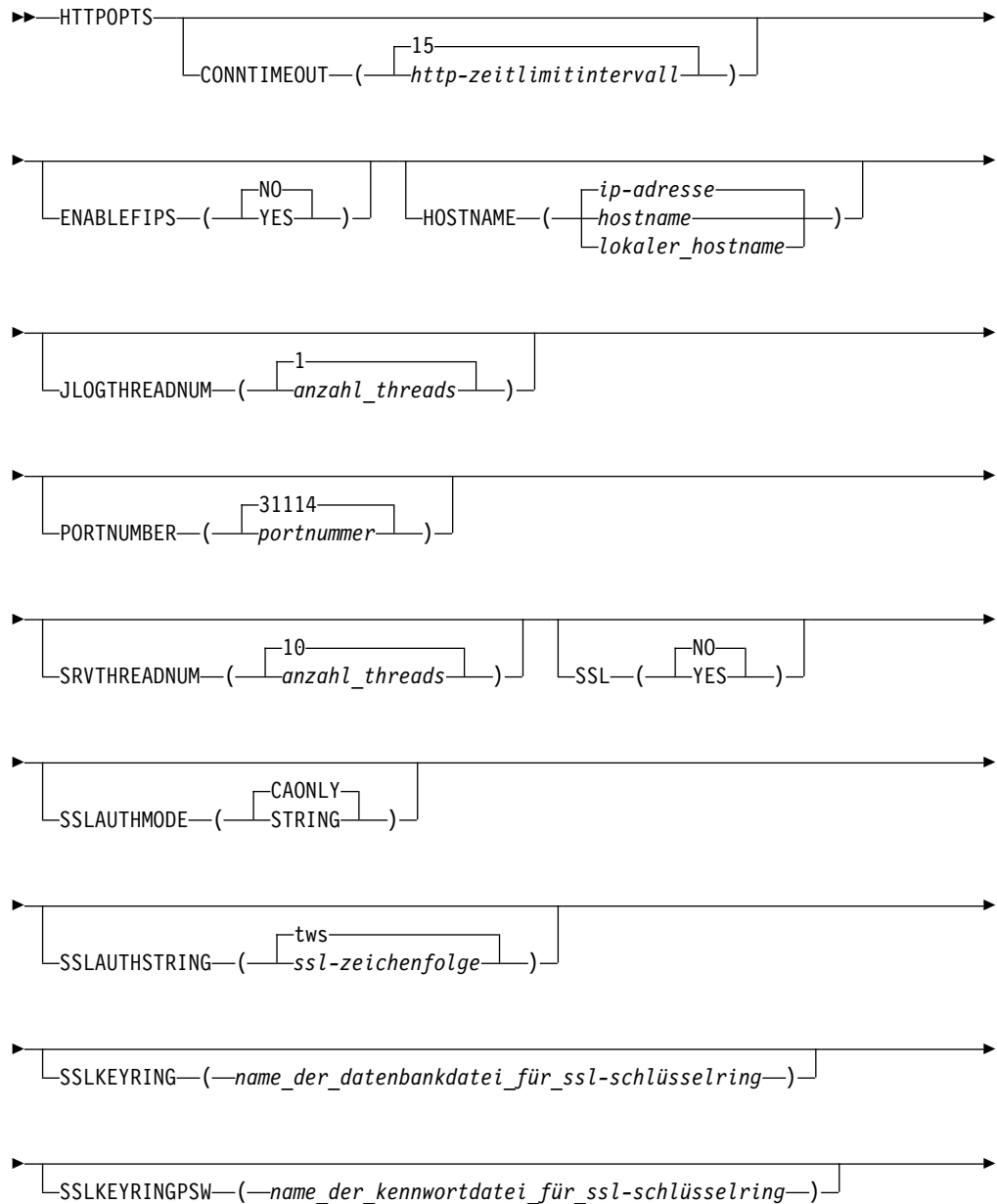
HTTP-Verbindungsoptionen definieren

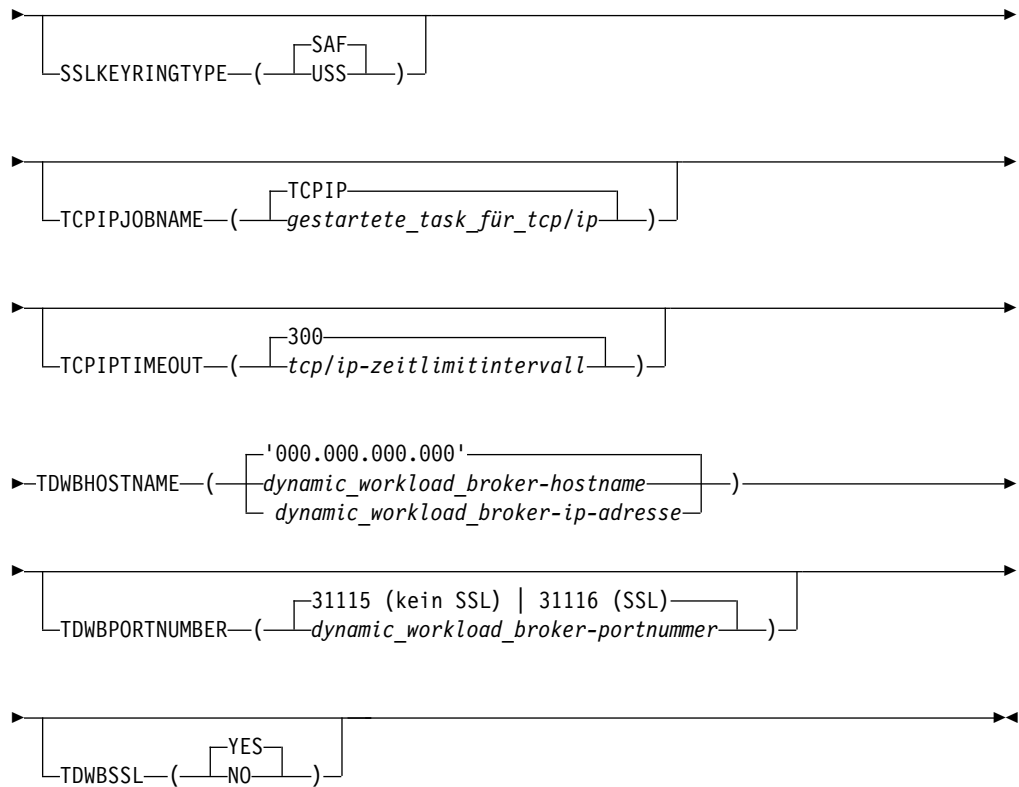
In diesem Abschnitt ist die Initialisierungsanweisung HTTPOPTS beschrieben.

Diese Anweisung definiert die Details für die Verbindung zwischen dem Agenten und Dynamic Workload Broker. Verwenden Sie sie, um Folgendes anzugeben:

- Hostname und Port des Agenten
- Hostname und Port der Gegenstelle für die Verbindung
- SSL-Sicherheitsoptionen

HTTPOPTS





Parameter

CONNTIMEOUT = (*zeitlimitintervall* | **15**)

Die Anzahl Sekunden, die eine HTTP-Verbindung wartet, bevor eine Zeitlimit-überschreitung auftritt. Gültige Werte liegen im Bereich von 1 bis 10000. Der Standardwert ist 15 Sekunden.

ENABLEFIPS (**NO** | **YES**)

Gibt an, ob die SSL-Kommunikation die FIPS-Standards einhalten muss. Geben Sie YES an, damit die SSL-Kommunikation FIPS-konform ist. Dieses Schlüsselwort wird ignoriert, wenn die SSL-Kommunikation nicht aktiviert ist. Der Standardwert ist NO.

Weitere Informationen zum Aktivieren der Unterstützung für den FIPS-Standard finden Sie in *IBM Workload Scheduler for z/OS: Planung und Installation*.

HOSTNAME = (*hostname* | *ip-adresse*)

Der Name des lokalen Hosts oder die IP-Adresse des Agenten für z/OS, der bzw. die für die Kommunikation mit Dynamic Workload Broker verwendet wird. Der Wert kann bis zu 52 alphanumerische Zeichen enthalten. Der Hostname oder die IP-Adresse können im IPV4- oder IPV6-Format angegeben werden. Schließen Sie diesen Wert in einfache Anführungszeichen ein. Der Standardwert ist die IP-Adresse, die von TCP/IP zurückgegeben wird.

JLOGTHREADNUM = (*anzahl_threads* | **1**)

Die Anzahl Threads, die von der HTTP-Servertask verwendet werden, um die Anforderungen in Bezug auf das Jobprotokoll zu verwalten. Gültige Werte liegen im Bereich von 1 bis 100. Der Standardwert ist 1.

PORTNUMBER = (*port* | **31114**)

Die Portnummer des Agenten für z/OS, die für die Kommunikation mit Dynamic Workload Broker verwendet wird. Gültige Werte liegen im Bereich von 0 bis 65535. Der Standardwert ist 31114.

SSL = (*Yes* | **No**)

Gibt an, ob SSL an PORTNUMBER konfiguriert ist, um ankommende Anforderungen zu schützen. Setzen Sie diesen Parameter auf Yes, wenn Sie SSL verwenden, um den Port des Agenten für z/OS zu schützen. Andernfalls setzen Sie diesen Parameter auf No. Der Standardwert ist No. Ist SSL aktiv, ist der Parameter SSLKEYRING obligatorisch.

SSLAUTHMODE = (*STRING* | **CAONLY**)

Der SSL-Authentifizierungstyp. Gültige Werte sind:

CAONLY

Der Scheduler überprüft die Gültigkeit des Zertifikats durch Überprüfen, ob eine anerkannte Zertifizierungsstelle das Peerzertifikat ausgegeben hat. Die in dem Zertifikat enthaltenen Informationen werden nicht geprüft.

STRING

Der Scheduler überprüft die Gültigkeit des Zertifikats wie unter der Option CAONLY beschrieben. Darüber hinaus überprüft er, ob der allgemeine Name (Common Name - CN) des Zertifikatsubjekts mit der Zeichenfolge übereinstimmt, die im Parameter SSLAUTHSTRING angegeben ist.

Der Standardwert ist CAONLY.

SSLAUTHSTRING = (*ssl-zeichenfolge* | **tws**)

Die SSL-Zeichenfolge, die für die Gültigkeitsprüfung des Zertifikats verwendet wird, wenn Sie SSLAUTHMODE auf STRING setzen. Die Zeichenfolge kann bis zu 64 Zeichen enthalten. Der Standardwert ist tws.

SSLKEYRING = (*name_der_datenbankdatei_für_ssl-schlüsselring*)

Hat SSLKEYRINGTYPE den Wert SAF (System Authorization Facility), gibt dieser Parameter den SAF-Schlüsselring an, der zum Verbinden der Sicherheitszertifikate verwendet wird.

Hat SSLKEYRINGTYPE den Wert USS (Unix System Services), gibt dieser Parameter die Datenbank an, die Schlüssel und Zertifikate enthält. Er besteht aus dem Namen eines SSL-Arbeitsverzeichnisses und einem Dateinamen in dem folgenden Format:

ssl-arbeitsverz/TWS.kbd

Bei dem Parameter muss die Groß-/Kleinschreibung beachtet werden.

SSLKEYRINGPSW = (*name_der_kennwortdatei_für_ssl-schlüsselring*)

Dieser Parameter ist erforderlich, wenn Sie SSL-Sicherheit ausführen und SSLKEYRINGTYPE den Wert USS hat. Er gibt die Datei an, die das Schlüsselkennwort enthält. Er besteht aus dem Namen eines SSL-Arbeitsverzeichnisses und einem Dateinamen in dem folgenden Format:

ssl-arbeitsverz/TWS.sth

Wird kein vorhandener und korrekter Dateiname angegeben, wird eine Fehlermeldung ausgegeben und der Agent kann nicht gestartet werden. Bei dem Parameter muss die Groß-/Kleinschreibung beachtet werden.

SSLKEYRINGTYPE = (*USS* | *SAF*)

Gibt an, ob es sich bei der Schlüsselringdatei um eine USS-Datei für die Schlüsseldatenbank oder um einen SAF-Schlüsselring handelt. Ist der Typ SAF, können Sie den Befehl RACF verwenden, um SSL-Verbindungen zu verwalten.

Wichtig: Ist der Typ USS, müssen Sie einen Kennwortdateinamen für den SSL-Schlüsselring für SSLKEYRINGPSW angeben. Andernfalls kann der Agent nicht gestartet werden.

SRVTHREADNUM = (*anzahl_threads* | 10)

Die Anzahl Threads, die die HTTP-Servertask verwenden kann, um mehrere Anforderungen zu verarbeiten, die gleichzeitig von Dynamic Workload Broker gesendet werden. Gültige Werte liegen im Bereich von 2 bis 100. Der Standardwert ist 10.

TCPIJOBNAME = (*gestartete-task_für_tcp/ip* | TCPIP)

Der Name der gestarteten Task für TCP/IP, die auf dem z/OS-System ausgeführt wird. Der Standardname ist TCPIP.

TCPIPTIMEOUT = (*tcp/ip-zeitlimitintervall* | 300)

Die Anzahl Sekunden, die eine HTTP-Anforderung auf eine Antwort wartet, bevor eine Zeitlimitüberschreitung auftritt. Gültige Werte liegen im Bereich von 1 bis 10000. Der Standardwert ist 300.

TDWBHOSTNAME = (*dynamic_workload_broker-hostname* | *dynamic_workload_broker-ip-adresse* | '000.000.000.000')

Der Name des lokalen Hosts oder die IP-Adresse des Dynamic Workload Broker, zu dem der Agent für z/OS eine HTTP-Verbindung herstellen soll. Der Wert kann bis zu 52 alphanumerische Zeichen enthalten. Der Hostname oder die IP-Adresse können im IPV4- oder IPV6-Format angegeben werden. Schließen Sie diesen Wert in einfache Anführungszeichen ein. Der Parameter ist obligatorisch.

TDWBPORTNUMBER = (*port* | 31115 | 31116)

Die Portnummer des Dynamic Workload Broker, zu dem der Agent für z/OS eine HTTP-Verbindung herstellen soll. Die Standardwerte sind 31115 für Verbindungen ohne SSL und 31116 für SSL-Verbindungen.

TDWBSSL = (Yes | No)

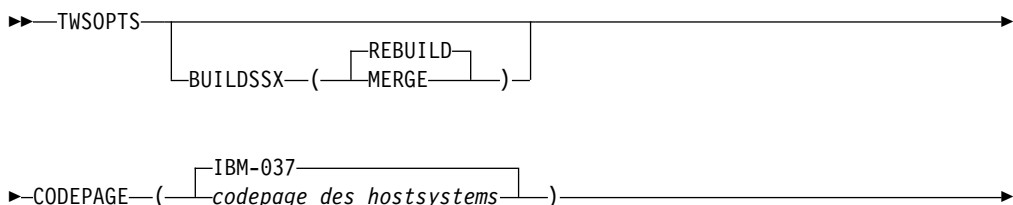
Gibt an, ob der Dynamic Workload Broker-Port, der durch TDWBPORTNUMBER definiert ist, mit SSL geschützt ist. Der Standardwert ist Yes.

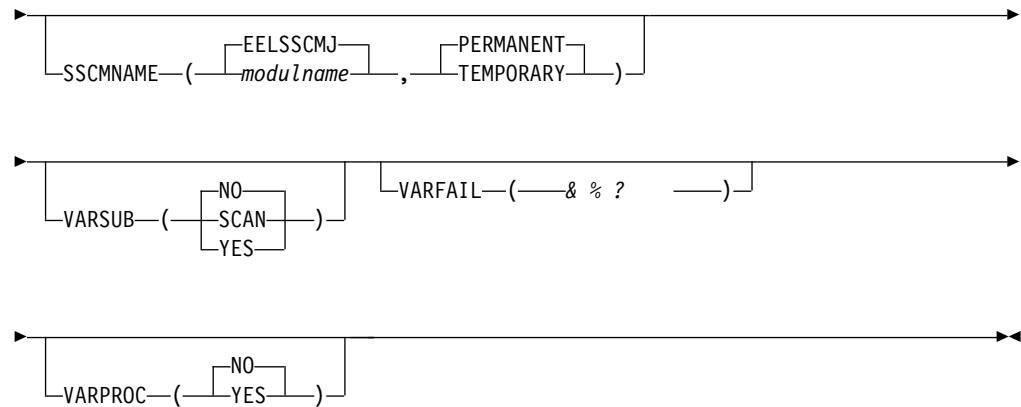
Allgemeine Laufzeioptionen für den Agenten angeben

In diesem Abschnitt ist die Initialisierungsanweisung TWSOPTS beschrieben.

Diese Anweisung definiert Laufzeioptionen für den Agenten für z/OS.

TWSOPTS





Parameter

BUILDSSX = (MERGE | REBUILD)

Definiert, ob die CVT-Erweiterung (CVT - Communication Vector Table) des Subsystems für den Agenten für z/OS, die SSX, auf einer neuen Stufe neu erstellt werden soll, wenn der Adressraum gestartet wird. Die SSX wird bei der Subsysteminitialisierung vom Modul EELINITJ erstellt. Wurde das Modul EELINITJ seitdem aktualisiert, entweder durch Wartung oder weil Sie ein neues Release oder eine neue Modifikationsstufe des Agenten für z/OS installieren, verwenden Sie das Schlüsselwort BUILDSSX, um einen IPL des z/OS-Systems zu vermeiden.

Geben Sie MERGE an, wenn operative Daten wie die Ereignisschreiberwarteschlange in die neue SSX kopiert werden sollen. So ist sichergestellt, dass die neue Ereignisschreiberwarteschlange die Ereignisse enthält, die in die Warteschlange des alten SSX-Blocks gestellt wurden. Verwenden Sie diese Option, wenn Sie einen Adressraum des Agenten für z/OS starten, nachdem Sie Wartungsaktualisierungen installiert haben.

Geben Sie REBUILD an, wenn Sie auf ein neues Release oder eine neue Modifikationsstufe des Agenten für z/OS migrieren bzw. wenn Sie zu einem älteren Release oder einer älteren Modifikationsstufe des Agenten zurückkehren. Die Ereignisschreiberwarteschlange der alten SSX wird in der neuen SSX nicht referenziert. Stellen Sie sicher, dass Sie außerdem den Namen des neuen Kommunikationsmoduls für das Subsystem mithilfe des Schlüsselworts SSCMNAME angeben.

Der Standardwert ist REBUILD.

Wichtig:

- Im Abschnitt HOLD des PTF-Begleitschreibens sind die Funktionsaktualisierungen angegeben, die eine Neuerstellung der SSX erfordern.
- MERGE kann nicht verwendet werden, wenn die alten oder neuen SSX-Blöcke für verschiedene FMIDs erstellt werden. Verwenden Sie MERGE nicht, wenn Sie auf ein neues Release oder eine neue Modifikationsstufe des Agenten für z/OS migrieren bzw. wenn Sie zu einem älteren Release oder einer älteren Modifikationsstufe des Agenten zurückkehren.
- Wenn Sie BUILDSSX(REBUILD) angeben, um auf ein neues Release oder eine neue Modifikationsstufe des Agenten für z/OS zu migrieren bzw. um

zu einem älteren Release oder einer älteren Modifikationsstufe des Agenten zurückzukehren, stellen Sie sicher, dass Sie auch das Schlüsselwort SSCM-NAME angeben.

- Der Parameter BUILDSSX sollte nach dem nächsten IPL des z/OS-Systems entfernt werden, da er nicht mehr erforderlich ist.

CODEPAGE = (codepage_des_hostsystems | IBM-037)

Der Name der Host-Codepage. Dieser Wert ist erforderlich, da er von der Überwachungstask für die Konvertierung der Überwachungsdaten verwendet wird, die an den Überwachungsagenten gesendet werden. Geben Sie eine Codepage aus der folgenden Liste der IBM-*mmm*-Werte an. Dabei ist *mmm* die EBCDIC-Codepage, die für Ihr z/OS-System verwendet wird:

IBM-037

USA, Portugal, Kanada (Französisch). Dies ist die Standardeinstellung.

IBM-273

Deutschland

IBM-274

Belgien

IBM-277

Dänemark - Norwegen

IBM-278

Schweden - Finnland

IBM-280

Italien

IBM-284

Spanien - Lateinamerika

IBM-285

Großbritannien

IBM-297

Frankreich

IBM-424

Israel

IBM-500

International

IBM-838

Thailändisch

IBM-933

Korea

IBM-935

China

IBM-937

Taiwan

IBM-939

Japan, erweitert

IBM-970

Lateinisch 2

- IBM-971**
Island
- IBM-975**
Griechenland
- IBM-1025**
Kyrillisch
- IBM-1026**
Lateinisch 5 (Türkei)
- IBM-1047**
Offene Systeme
- IBM-1112**
Baltikum
- IBM-1122**
Estland
- IBM-1388**
China

Die folgende Liste enthält EBCDIC-Codepages für EURO-Unterstützung:

- IBM-1140**
Finnland, Schweden
- IBM-1141**
Österreich, Deutschland
- IBM-1142**
Dänemark, Norwegen
- IBM-1143**
USA
- IBM-1144**
Italien
- IBM-1145**
Spanien, spanischsprachige Länder in Lateinamerika
- IBM-1146**
Großbritannien
- IBM-1147**
Frankreich
- IBM-1148**
Belgien, Schweiz
- IBM-1149**
Island

SSCMNAME = (*modulname* | **EELSSCMJ**, **PERMANENT** | **TEMPORARY**)

Der erste Schlüsselwortwert definiert den Namen des Kommunikationsmoduls für das Subsystem, das anstelle des beim IPL geladenen Moduls EELSSCMJ verwendet werden soll. Der zweite Schlüsselwortwert gibt an, wie lange das Modul das beim IPL geladene Modul ersetzen soll. Verwenden Sie dieses Schlüsselwort, um eine aktualisierte Version des Moduls vor einem geplanten IPL zu laden. Das Modul, das Sie angeben, muss sich in einer APF-autorisierten Bibliothek befinden, die entweder durch den Datendefinitionsnamen STEPLIB oder die Verkettung LNKLST*nn* definiert ist. Wird SSCMNAME nicht an-

gegeben oder wird durch SSCMNAME ein Modul angegeben, das in der autorisierten Bibliothek nicht gefunden wird, werden die Ereignisse des Agenten für z/OS weiterhin durch das beim IPL geladene Modul EELSSCMJ generiert.

Geben Sie PERMANENT als zweiten Schlüsselwortwert an, um das beim IPL geladene Kommunikationsmodul des Subsystem durch das Modul zu ersetzen, das im ersten Schlüsselwortwert angegeben ist. In diesem Fall muss das angegebene Modul sich in einer APF-autorisierten Bibliothek befinden, die durch den Datendefinitionsnamen STEPLIB definiert ist. Dies ist die Standardeinstellung.

Wird TEMPORARY angegeben bzw. als Standardwert des zweiten Schlüsselworts automatisch verwendet, generiert das von Ihnen angegebene Modul Jobüberwachungsereignisse nur, solange der Adressraum des Agenten für z/OS aktiv ist. Wenn der Adressraum gestoppt wird, werden die Ereignisse weiterhin durch das beim IPL geladene Modul EELSSCMJ generiert.

Wichtig:

- Im Abschnitt HOLD des PTF-Begleitschreibens sind die Funktionsaktualisierungen angegeben, die das Laden eines neuen Kommunikationsmoduls für das Subsystem erfordern.
- Stellen Sie sicher, dass Sie dieses Schlüsselwort angeben, wenn die Option BUILDSSX(REBUILD) verwendet wird, um auf ein neues Release oder eine neue Modifikationsstufe des Agenten für z/OS zu migrieren bzw. um zu einem älteren Release oder einer älteren Modifikationsstufe des Agenten zurückzukehren.
- Das Schlüsselwort SSCMNAME sollte nach dem nächsten IPL entfernt werden, da es nicht mehr erforderlich ist.

VARSUB(YES|NO|SCAN)

Dieses Schlüsselwort gibt an, ob die JCL-Variablensubstitution durchgeführt werden soll. YES bedeutet, dass die Variablensuche für alle Jobs durchgeführt wird. NO bedeutet, dass keine Variablensuche erfolgt. SCAN weist den Agenten an, die JCL nur dann nach Variablen zu durchsuchen, wenn die Anweisung `//*%OPC SCAN` in der JCL gefunden wird.

VARFAIL(&, %, ?)

Dieses Schlüsselwort gibt an, ob nicht aufgelöste Variablen in der JCL einen JCL-Fehler verursachen sollen. Sie können ein, zwei oder drei der folgenden Zeichen in beliebiger Reihenfolge verwenden, um einen Substitutionsfehler zu umgehen: (&, %, ?).

Wenn beispielsweise VARFAIL(&) angegeben wird, betrachtet der Agent das Fehlschlagen der Substitution von Variablen, die mit einem &-Zeichen beginnen, nicht als Fehler. Jede beliebige Kombination aus den drei Typen ist zulässig, wie zum Beispiel VARFAIL(&, %) oder VARFAIL(?). Es muss jedoch mindestens ein Wert angegeben werden. Eine Wiederholung von Zeichen wird zurückgewiesen.

Wenn VARFAIL nicht angegeben wird, werden alle nicht substituierten Variablen wie zuvor als Fehler behandelt.

VARPROC(YES|NO)

Dieses Schlüsselwort gibt an, ob Onlineprozeduren die Variablensubstitution berücksichtigen sollen. Bei der Angabe VARPROC(YES) werden Variablen in Onlineprozeduren aufgelöst.

Der Standardwert ist NO.

Exits des Agenten für z/OS konfigurieren

Die Exits EELUX000, EELUX002 und EELUX004 werden vom Agenten für z/OS aufgerufen. Ihre eigenen Programme können die von den Exits übergebenen Informationen verwenden, um eine Reihe von Funktionen auszuführen.

Jeder Exit wird geladen, wenn das Exitmodul vorhanden ist und wenn der Exit nicht inaktiviert oder in der Initialisierungsanweisung EXITS durch einen anderen Exitnamen ersetzt wurde.

Exits werden mithilfe von Standardverbindungskonventionen aufgerufen. Beim Eingang in den Exit zeigt Register 1 auf eine Parameterliste. Jede Adresse in dieser Liste zeigt auf einen Parameter, der an den Exit übergeben wird.

Der Eingang in die Exits erfolgt mit der RACF-Berechtigung des Subsystems des Agenten für z/OS.

Exit EELUX000 (Start/Stop) konfigurieren

In diesem Abschnitt ist der Exit für den Start/Stop des Agenten für z/OS (EELUX000) beschrieben.

EELUX000 wird aufgerufen, wenn der Agent für z/OS startet und wenn der Agent normal beendet wird. Sie können diesen Exit verwenden, um beim Start des Agenten Ressourcen zuzuordnen und um die Ressourcen beim Stopp des Agenten freizugeben. Auf diese Weise reduziert sich der Aufwand, der mit der Zuordnung und der anschließenden Freigabe von Ressourcen bei jeder Verwendung verbunden ist.

Die Beispielbibliothek SEELSAMP, die während der Installation erstellt wurde, enthält den Exit EELUX000. Dieser Exit ist ein Beispiel für Start-/Stoppexits.

Exit installieren

Das Lademodul, das den Start-/Stoppexit implementiert, muss in der LNKLST-Verkettung per Link-Edit mit einer APF-autorisierten Bibliothek verknüpft oder durch die Datendefinitionsanweisung STEPLIB in der JCL-Prozedur des Agenten für z/OS definiert sein. Wenn das Lademodul Eingabe- oder Ausgabeoperationen ausführt, muss es unter Beachtung der normalen z/OS-Einschränkungen per Link-Edit mit RMODE(24) verknüpft werden. Alternativ kann es mit RMODE(ANY) verknüpft werden.

Der Agent für z/OS ruft den Exit in AMODE 31 auf; der bei der Verknüpfung angegebene Parameter AMODE hat keine Auswirkungen.

Schnittstelle für den Exit

Der Start-/Stoppexit wird im Taskmodus, Problemstatus und Schlüssel 8 aufgerufen und die Jobabschnitttask ist APF-autorisiert. Die aktive Task wird mit derselben Zugriffsberechtigung wie die Jobabschnitttask ausgeführt. Der Exit muss diesen Status wiederherstellen, bevor die Steuerung an das aufrufende Modul zurückgegeben wird.

Die Steuerung wird mit der Anweisung BAL an den Exit übergeben. Bei der Rückgabe der Steuerung an das aufrufende Modul muss der Exit die Adresse und den Adressierungsmodus verwenden, die in dem allgemeinen Register 14 an den Exit übergeben wurden.

Der Eingang in den Exit erfolgt in AMODE 31. Der Exit muss jedoch in AMODE 24 umschalten, bevor er Eingabe- oder Ausgabeoperationen ausführt, und anschließend in AMODE 31 zurückschalten, bevor die Steuerung an das aufrufende Modul zurückgegeben wird.

Beim Eingang in den Exit enthält Register 1 die Adresse der Parameterliste. Jede Adresse in dieser Liste wird verwendet, um den Parameterwert zu lokalisieren. Die folgenden Parameter werden an den Exit übergeben:

EELUX000-Parameter			
ACTION	DS	CL8	(Start-/Stoppaktion)
MCAUSERF	DS	A	(Benutzerfeld)

ACTION hat den Wert START, wenn der Exit beim Start des Agenten aufgerufen wird. MCAUSERF ist null für diesen ursprünglichen Aufruf. Normalerweise führt dieser Exit Exitinitialisierungsfunktionen für den Startaufruf aus, wenn Sie den Agenten starten. Wenn der Exit Speicher zuordnen muss, der verwendet wird, während der Agent aktiv ist, sollten Sie MCAUSERF aktualisieren, um diesen Speicher zu adressieren.

ACTION hat den Wert STOP, wenn der Exit beim Beenden des Agenten aufgerufen wird. Normalerweise führt dieser Exit Exitbeendigungsfunktionen für den Stoppaufruf aus, wenn Sie den Agenten stoppen. Wird MCAUSERF durch den Startaufruf aktualisiert, wird derselbe Wert für den Stoppaufruf an den Exit übergeben.

Exit EELUX002 (Lesen in der Jobbibliothek) konfigurieren

Der Exit EELUX002 wird aufgerufen, wenn ein Job nach Verweis für die Verarbeitung ausgewählt wird, die Jobdefinition jedoch den Namen der Datei nicht enthält, in der die JCL gespeichert ist. In diesem Fall durchsucht der Agent für z/OS standardmäßig die Verkettung von Dateien, die dem Datendefinitionsnamen von EELJBLIB in der JCL-Prozedur des Agenten zugeordnet ist. Wenn der Agent jedoch andere Dateien durchsuchen soll, verwenden Sie den Exit EELUX002 zur Ausführung dieser Funktion.

Ziehen Sie eine Verwendung von EELUX002 auch in Betracht, um die Leistung zu verbessern, wenn Sie viele große partitionierte Dateien (PDS) mit EELJBLIB verketten haben. Zum Auffinden eines Members in der letzten Datei der Verkettung muss der Agent das Verzeichnis aller vorhergehenden PDS lesen, was einen erheblichen Systemaufwand bedeuten kann. Ziehen Sie in Betracht, für jede Computer-Workstation eine partitionierte Datei und einen entsprechenden Datendefinitionsnamen zu definieren.

Das SEELSAMP-Member EELUX002 enthält einen Beispiexit für das Lesen in der Jobbibliothek. Dieses Beispiel sucht einen Datendefinitionsnamen mit der Bezeichnung MYJOB LIB, bevor es EELJBLIB durchsucht.

Exit installieren

Das Lademodul, das den Exit zum Lesen in der Jobbibliothek implementiert, muss in der LNKST-Verkettung per Link-Edit mit einer APF-autorisierten Bibliothek verknüpft oder durch die Datendefinitionsanweisung STEPLIB in der JCL-Prozedur des Agenten für z/OS definiert sein.

Schnittstelle für den Exit

Der Exit für Lesen in der Jobbibliothek wird im Taskmodus, Problemstatus und Schlüssel 8 aufgerufen und die Jobabschnitttask ist APF-autorisiert. Die aktive Task wird mit derselben Zugriffsberechtigung wie die Jobabschnitttask ausgeführt. Der Exit muss diesen Status wiederherstellen, bevor die Steuerung an das aufrufende Modul zurückgegeben wird.

Die Steuerung wird mit der Anweisung BAL an den Exit übergeben. Bei der Rückgabe der Steuerung an das aufrufende Modul muss der Exit die Adresse und den Adressierungsmodus verwenden, die in dem allgemeinen Register 14 an den Exit übergeben wurden.

Wenn der Exit abgebrochen wird, wird er als *nicht ausführbar* markiert. Der Agent für z/OS versucht nicht, den Exit erneut aufzurufen.

Beim Eingang in den Exit enthält Register 1 die Adresse der Parameterliste. Jede Adresse in dieser Liste wird verwendet, um den Parameterwert zu lokalisieren. Die folgenden Parameter werden an den Exit übergeben:

Parameter für EELUX002

TYPE	DS	CL1	(Konstante = J)
FUNC	DS	CL1	(Konstante = G)
JOBNAME	DS	CL8	(Jobname)
IOAREA	DS	A	(Adresse des E/A-Bereichs)
IOAREAL	DS	F	(Größe des E/A-Bereichs)
RETCODE	DS	X	(Rückkehrcode)
DATAL	DS	F	(zurückgegebenes Datenvolumen)
ERRDATA	DS	CL78	(zurückgegebene Fehlernachricht)
ADID	DS	CL16	(Name der aktuellen Anwendung)
USRAREA	DS	A	(Benutzerfeld, 0 bei erstem Aufruf)
JCLUSER	DS	CL8	(letzter Benutzer, der diesen Job aktualisiert hat)
OPNUM	DS	F	(Operationsnummer)
IATIME	DS	CL10	(Eingangszeit der Laufeingabe, JJMMTHMM)
VAROCCP	DS	A	(Adresse der Laufdaten, wenn Operation in CP ist)
VAROPRP	DS	A	(Adresse der Operationsdaten, wenn Operation in CP ist)
VARWSP	DS	A	(Adresse der Workstationdaten, wenn Operation in CP ist)
MCAUSERF	DS	A	(Adresse, die vom Benutzer im Exit EELUX000 festgelegt wurde)
OCCPTR	DS	A	(Adresse der Laufdaten)
OPRPTR	DS	A	(Adresse der Operationsdaten)
WSPTR	DS	A	(Adresse der Workstationdaten)
AUTHGROU	DS	CL8	(Berechtigungsgruppe)
MEMPRO	DS	CL1	(Anzeiger für Hauptspeicherprobleme)
TASKPTR	DS	A	(Adresse des TCB der Aufrufertask)
XINFO	DS	A	(Adresse der erweiterten Informationen)
XJNAMLEN	DS	F	(Länge des erweiterten Jobnamens)
USRFNR	DS	F	(Anzahl Benutzerfelder)
USRFAREA	DS	A	(Bereichsadresse für Benutzerfelder)

JOBNAME

Der Name des Jobs, der übergeben werden soll.

IOAREA

Die Adresse eines Puffers, der vom Agenten für z/OS zugeordnet wurde und in den die JCL-Sätze für den aktuellen Job platziert werden müssen.

IOAREAL

Der Speicherplatz in Byte im IOAREA-Puffer.

RETCODE

Wird von dem Exit gesetzt. Die folgenden Werte sind gültig:

0 Normale Rückgabe.

- 4 Datenende für den aktuellen Job wurde erreicht.
- 16 Der Job wurde in keiner Eingabedatei gefunden.
- 20 Es ist keine JCL vorhanden, die vom Exit zurückgegeben werden kann. Der Agent für z/OS versucht, die JCL aus EELJBLIB abzurufen.
- 44 Nicht genügend Speicherplatz. Der freie Speicherplatz im IOAREA-Puffer (wie durch IOAREAL bestimmt) reicht für den nächsten Datenblock nicht aus.
- 241 Ein E/A-Fehler ist aufgetreten.
- 242 Ein Fehler beim Öffnen ist aufgetreten. Mindestens eine Eingabedatei konnte nicht geöffnet werden.

Der Exit wird erneut aufgerufen, um die Verarbeitung desselben Jobs fortzusetzen, wenn ein Rückkehrcode 0 oder 44 zurückgegeben wird. Alle anderen Rückkehrcodes beenden die Verarbeitung des aktuellen Jobs.

DATAL

Das Datenvolumen, das vom Exit zurückgegeben wird, wenn der Rückkehrcode 0 oder 4 ist.

ERRDATA

Ein Benutzernachrichtenbereich, in dem Sie ein im Exit gefundenes Problem beschreiben können. Der Text wird in der Nachricht EELJ020 ausgegeben, wenn der Rückkehrcode 242 vom Exit festgelegt wird, oder in der Nachricht EELJ024, wenn der Rückkehrcode 241 festgelegt wird.

Anmerkung: Wenn Sie den Nachrichtenbibliothekseintrag für EELJ020 oder EELJ024 ändern, um eine WTO-Nachricht zu generieren, müssen Sie sicherstellen, dass für jede Zeile nicht mehr als 70 Zeichen Nachrichtentext definiert werden. Ordnen Sie den Text bei Bedarf um. Stellen Sie außerdem sicher, dass in ERRDATA selbst nicht mehr als 70 Zeichen enthalten sind.

ADID Der Name der aktuellen Anwendung.

USRAREA

Hat beim ersten Aufruf des Exits zum Abrufen eines Jobs den Wert null. Der Exit kann diesen Parameter auf einen beliebigen Wert setzen. Der Agent für z/OS verwendet oder aktualisiert diesen Parameter nicht.

Der Exit sollte den Parameter USRAREA verwenden, wenn er einen Rückkehrcode 0 oder 44 zurückgibt. Normalerweise dient der Parameter USRAREA dazu, die Adresse eines Arbeitsbereichs aufzunehmen, für den der Exit eine GETMAIN-Operation ausgeführt hat. Dieser Arbeitsbereich sollte genügend Informationen enthalten, um dem Exit das Fortsetzen der Verarbeitung desselben Jobs zu ermöglichen.

JCLUSER

Hat beim ersten Aufruf des Exits den Wert null. Der Exit sollte diesen Parameter auf den Namen des z/OS-Benutzers setzen, der zur Berechtigungsprüfung verwendet wird, wenn die JCL Anweisungen zur automatischen Wiederherstellung enthält.

OPNUM

Die Operationsnummer der Operation, die diesen Job darstellt.

IATIME

Der Eingangszeitpunkt des Anwendungslaufs, zu dem dieser Job gehört.

MCAUSERF

Ein Benutzerfeld, über das Sie Ressourcen im Start-/Stoppexit EELUX000 zuordnen können, die dieser Exit später verwenden kann. Es ist zum Beispiel möglich, Dateien für den JCL-Abruf im Starttypaufruf von EELUX000 zu öffnen, anstatt sie bei jedem Aufruf von EELUX002 zu öffnen. Der Agent verwendet oder aktualisiert dieses Feld nicht. Das Feld MCAUSERF ist gültig, wenn der Agent aktiv ist.

AUTHGROU

Der Name der Berechtigungsgruppe.

MEMPRO

Der Anzeiger für Hauptspeicherprobleme.

TASKPTR

Die Adresse des Tasksteuerblocks der Aufrufertask.

Wenn der Exit Zugriff auf die eigenen Dateien benötigt, müssen diese Dateien beim ersten Aufruf für einen Job (USRAREA-Wert=0) geöffnet und auf eine der folgenden Arten geschlossen werden:

- Bevor die Steuerung an den Agenten zum letzten Mal zurückgegeben wird (bevor der Rückkehrcode 4 festgelegt wird).
- Wenn ein Fehler auftritt, der verhindert, dass der Agent mehr Speicherplatz abrufen, informiert der Agent den Exit, indem der Parameter MEMPRO auf den folgenden Wert gesetzt wird:

X'04' Wenn die Begrenzung von 608.000 Byte erreicht wird (Ausgabe von EELJ025)

X'08' Wenn nicht genügend Speicherplatz verfügbar ist (Ausgabe von EELJ021)

Wenn der Exit EELUX002 zum ersten Mal aufgerufen wird, um einen Job abzurufen, ist der E/A-Bereich 32.000 Byte groß. Wenn der Exit den gesamten Job abgerufen hat und dieser in den verfügbaren Pufferplatz passt, kann der Exit den E/A-Bereich aktualisieren, das Datenvolumen im Job zurückgeben und den Rückkehrcode 4 festlegen.

Wenn der Exit nicht den gesamten Job abgerufen hat, kann er den E/A-Bereich aktualisieren, das Datenvolumen im Job zurückgeben und den Rückkehrcode 0 zurückgeben, um anzugeben, dass noch mehr Daten zurückzugeben sind. Beim nächsten Aufruf des Exits werden die Adresse und die Größe des E/A-Bereichs aktualisiert, da der E/A-Bereich bereits zum Teil von Daten aus einem früheren Aufruf verwendet wird. Der Exit sollte diesen Prozess fortsetzen, bis keine zurückzugebenden Daten mehr vorhanden sind, und anschließend den Rückkehrcode 4 festlegen, um anzugeben, dass der gesamte Job abgerufen wurde.

Da sich der verfügbare Speicherplatz im Puffer bei jedem Aufruf verringert, ist es möglich, dass der Exit den Rückkehrcode 44 festlegen muss, um anzugeben, dass der freie Speicherplatz nicht ausreicht. Wenn der Rückkehrcode 44 zurückgegeben wird, wird der Exit erneut mit einem Jobnamen aus acht Gleichheitszeichen (=====) aufgerufen. Dies ist der Zurücksetzungsaufruf. Der Exit bereitet sich anschließend darauf vor, den Job von Anfang an erneut zu verarbeiten.

Beim Zurücksetzungsaufruf können keine Daten zurückgegeben werden. Wenn der Exit nach dem Zurücksetzungsaufruf wieder aufgerufen wird, ist der E/A-Bereich 32.000 Byte größer als vorher. Dieser Prozess der Rückgabe einer Bedingung "Nicht genügend Speicherplatz" kann bis zu 19-mal für einen Job wiederholt werden. Dies

bedeutet, dass die maximale Puffergröße, die durch den Exit EELUX002 angefordert werden kann, 608.000 Byte beträgt. Dies entspricht einem Job aus 7599 Kartenimages. Wenn die Begrenzung von 608.000 Byte erreicht wird, gibt der Agent die Nachricht EELJ025 aus und der Exit wird ein 20. Mal aufgerufen, wenn der Parameter MEMPRO auf 4 gesetzt wurde.

Der Exit kann außerdem mehr Pufferspeicherplatz dadurch abrufen, dass er den gesamten verfügbaren Speicherplatz im aktuellen Puffer verwendet. Wenn dies geschieht und der Rückkehrcode 0 festgelegt wird, wird der Exit erneut mit 32.000 Byte an freiem Speicherplatz im Puffer aufgerufen. In diesem Fall wird der Zurücksetzungsaufwurf nicht verwendet. Der Exit sollte die Verarbeitung des aktuellen Jobs normal fortsetzen. Diese Art der Erweiterung des Puffers kann bis zu einer maximalen Puffergröße von 608.000 Byte fortgesetzt werden.

Exit EELUX004 (Ereignisfilterung) konfigurieren

In diesem Abschnitt ist der Exit für Ereignisfilterung (EELUX004) beschrieben.

EELUX004 wird aufgerufen, wenn der Ereignisschreiber des Agenten für z/OS im Begriff ist, ein Ereignis in die Ereignisdatei zu schreiben. Mit diesem Exit können Sie von JES- und SMF-Exits erstellte Ereignisse löschen.

Dieser Exit wird häufig verwendet, um Ereignisse zu filtern, die von der Verarbeitung außerhalb der Produktionsumgebung erstellt werden. Wenn Sie eine erhebliche Anzahl von Testjobs und sonstige Verarbeitung außerhalb der Produktionsumgebung ausführen und Ihre Standards für Jobnamen eine Unterscheidung zulassen, empfiehlt sich möglicherweise die Verwendung von EELUX004 zur Filterung dieser Verarbeitung. Die Beispielbibliothek SEELSAMP, die während der Installation erstellt wurde, enthält den Exit EELUX004. Dieser Exit ist ein Beispiel für Ereignisfilterungsexits.

Exit installieren

Das Lademodul, das den Ereignisfilterungsexit implementiert, muss in der LNKLST-Verkettung per Link-Edit mit einer APF-autorisierten Bibliothek verknüpft oder durch die Datendefinitionsanweisung STEPLIB in der JCL-Prozedur des Agenten für z/OS definiert sein. Das Lademodul sollte unter Beachtung der normalen z/OS-Einschränkungen per Link-Edit mit RMODE(24) verknüpft werden.

Der Agent für z/OS ruft den Exit in AMODE 24 auf; der bei der Verknüpfung angegebene Parameter AMODE hat keine Auswirkungen.

Schnittstelle für den Exit

Der Ereignisfilterungsexit wird im Taskmodus, Problemstatus und Schlüssel 8 aufgerufen und die Jobabschnitttask ist APF-autorisiert. Die aktive Task wird mit derselben Zugriffsberechtigung wie die Jobabschnitttask ausgeführt. Der Exit muss diesen Status wiederherstellen, bevor die Steuerung an das aufrufende Modul zurückgegeben wird.

Die Steuerung wird mit der Anweisung BAL an den Exit übergeben. Bei der Rückgabe der Steuerung an das aufrufende Modul muss der Exit die Adresse und den Adressierungsmodus verwenden, die in dem allgemeinen Register 14 an den Exit übergeben wurden.

Wird der Exit abgebrochen, wird er als *nicht ausführbar* markiert; der Agent versucht nicht, den Exit erneut aufzurufen.

Beim Eingang in den Exit enthält Register 1 die Adresse der Parameterliste. Jede Adresse in dieser Liste wird verwendet, um den Parameterwert zu lokalisieren. Die folgenden Parameter werden an den Exit übergeben:

EELUX004-Parameter			
JOBNAME	DS	CL8	(Name des aktuellen Jobs)
RETCODE	DS	F	(Rückkehrcode)
EXR	DS	CL80	(Exitdatensatz)

Dabei gilt Folgendes:

JOBNAME

Der Name des Jobs, für den ein Jobüberwachungsereignis erkannt wurde und für den ein Ereignisdatensatz in die Ereignisdatei geschrieben werden soll.

RETCODE

Wird von dem Exit gesetzt. Die folgenden Werte werden vom Prüfprogramm für die Jobbeendigung erkannt:

- 0** Normale Rückgabe. Der Ereignisschreiber setzt die normale Verarbeitung fort; das Ereignis wird in die Ereignisdatei geschrieben.
- 8** Dies ist kein Schedulerereignis. Das Ereignis wird nicht in die Ereignisdatei geschrieben. Ist das Ereignis jedoch ein Leserereignis (Typ 1) und wurde der Job von einem Jobüberwachungsexit des Schedulers angehalten, wird der Job vom Ereignisschreiber freigegeben.

EXR Der Exitdatensatz, der das Jobüberwachungsereignis beschreibt. Dieser Datensatz wird vom SMF- oder JES-Exit erstellt, der das Ereignis erkannt hat. Der Jobnummeroffset im Exitdatensatz, EXRJOBID, enthält JOB als erste drei Zeichen, wenn das Ereignis für einen Job erstellt wurde.

Agent in einer Sysplex-Umgebung ausführen

In diesem Abschnitt ist dokumentiert, wie Sie den Agenten für z/OS in einem z/OS-Sysplex ausführen können.

Beachten Sie bei der Ausführung des Agenten für z/OS in einem z/OS-Sysplex die folgenden Empfehlungen:

- In jedem Image des Sysplex sollte mindestens eine Instanz des Agenten für z/OS installiert sein.
- Das z/OS-Sysplex sollte nur mit einem einzigen Dynamic Workload Broker verbunden sein, obwohl ein einziger Dynamic Workload Broker mit mehreren Sysplex-Installationen oder Systemen verbunden sein kann.

Die folgende Abbildung zeigt vier Systeme, die in einer Sysplex-Umgebung ausgeführt werden, über XCF-Kommunikationsverbindungen (XCF - Cross-System Coupling-Facility) miteinander verbunden und mit einer Dynamic Workload Broker-Instanz verknüpft sind.

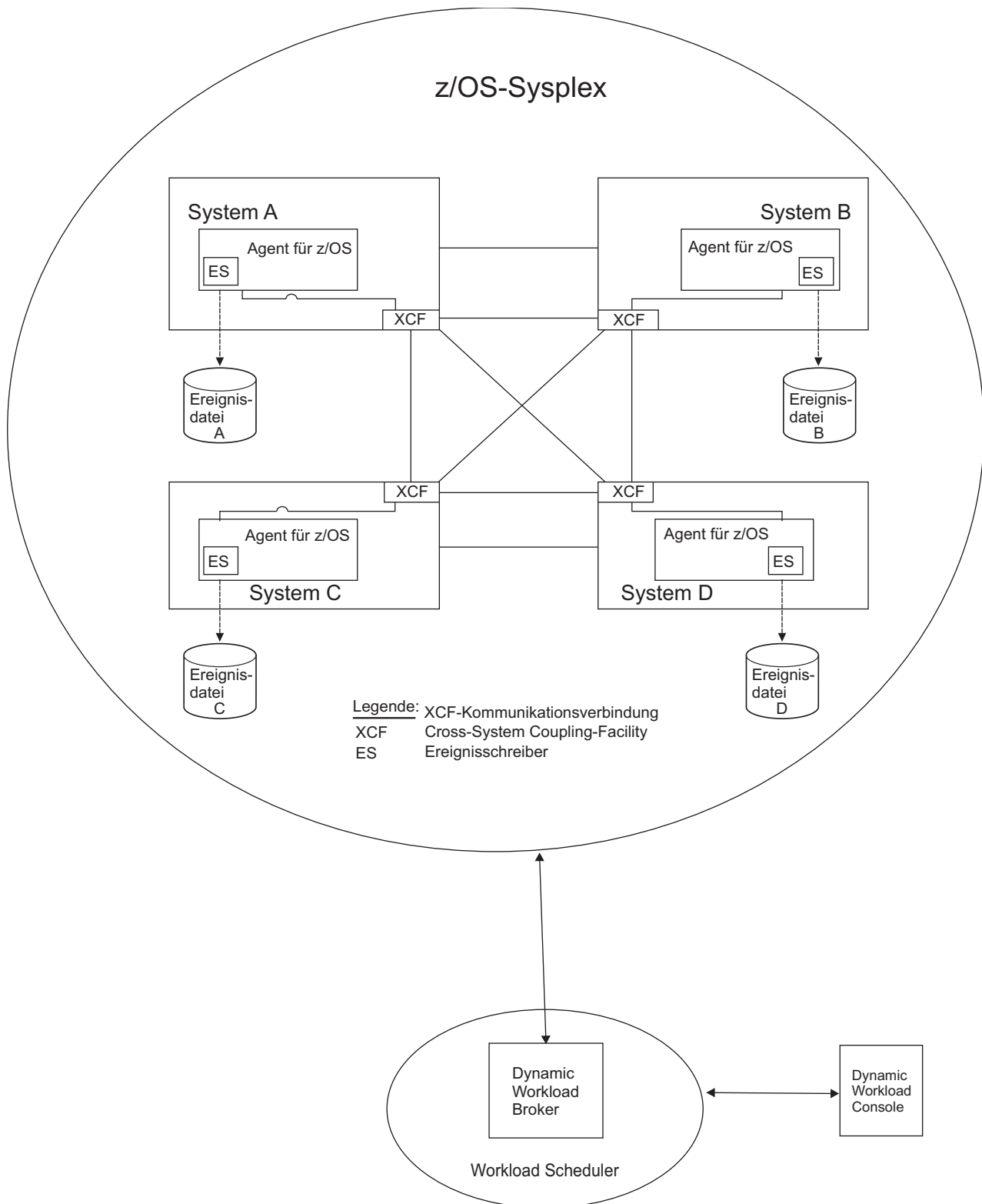


Abbildung 2. Agent für z/OS in einer SYSPLEX-Konfiguration

Die Weiterleitung eines Jobs an ein anderes System über NJE (Netzjobeintrag) wird mit dem Agenten für z/OS nicht unterstützt, weil die mögliche Abweichung seiner Job-ID und seiner Datums-/Zeitangabe im JES-Leser die Zuverlässigkeit der Überwachung beeinträchtigt.

Kapitel 3. Verwendung

Computer- und Workstationname des Agenten

Wenn der Agent für z/OS zum ersten Mal eine Verbindung zu IBM Workload Scheduler herstellt, wird ihm automatisch ein Computernamen zugeordnet.

Wenn der Agent für z/OS zum ersten Mal eine Verbindung zu der Komponente Dynamic Workload Broker von IBM Workload Scheduler herstellt, wird ihm ein Computernamen zugeordnet, mit dem die Workstation in IBM Workload Scheduler identifiziert wird.

Der zugeordnete Computernamen lautet:

subsystemname_systemname

Dabei gilt Folgendes:

subsystemname

Dies ist der Name der gestarteten Task von z/OS, die den Agenten startet.

systemname

Dies ist der Name des z/OS-Systems.

Der Subsystemnamen und der Systemnamen werden durch ein Unterstreichungszeichen () verbunden.

Beispiel: TDA1_ZOS10B1 ist der Computernamen eines Agenten, der von der gestarteten Task TDA1 im System ZOS10B1 gestartet wird.

Enthält der System- oder Subsystemnamen eines der Sonderzeichen, die durch X'5B', X'7B' oder X'7C' dargestellt werden (dies sind die Zeichen \$, # bzw. @ in der US-amerikanischen Codepage EBCDIC 037; sie werden jedoch in anderen EBCDIC-Codepages möglicherweise als andere Sonderzeichen angezeigt), werden die Sonderzeichen durch Unterstreichungszeichen () ersetzt, wenn der Computernamen zusammengesetzt wird. Beispiel: Lautet der Subsystemnamen ZAG5 und der Name des z/OS-Systems ZSY\$1 (in der Codepage IBM 037), ist der Computernamen, der diesem Agenten in IBM Workload Scheduler zugeordnet wird, ZAG5_ZSY_1.

Der Name wird in den Datenbanken von IBM Workload Scheduler und Dynamic Workload Broker gespeichert. Er wird verwendet, um einen Agenten für z/OS in den folgenden Schnittstellen zu identifizieren:

- Als Workstationnamen in den Composer- und Conman-Befehlszeilen und in den Dynamic Workload Console-Fenstern
- Als Computernamen in den Befehlszeilen und grafischen Benutzerschnittstellen von Dynamic Workload Broker

Dieser Name ist der sichtbare Teil eines längeren Identifikationskennsatzes, der dem Agenten bei der Installation zugeordnet wird. Eine interne ID wird verwaltet, um den Agenten im z/OS-System und in IBM Workload Scheduler zu verfolgen.

Den Workstationnamen können Sie mit dem Befehl `composer rename` oder mit Dynamic Workload Console ändern. Anschließend können Sie den neuen Namen in allen IBM Workload Scheduler-Schnittstellen verwenden, um den Agenten anzuge-

ben. Dies gilt nicht für die Dynamic Workload Broker-Schnittstellen, in denen stattdessen der Computernamen verwendet wird.

Agenten für z/OS auflisten

In Dynamic Workload Console und im Composer werden die Agenten für z/OS als Workstations des Typs Agent aufgelistet.

Führen Sie die normalen Prozesse zum Auflisten der Agenten aus:

- Wählen Sie in Dynamic Workload Console die Optionen **Terminierungsumgebung**→**Design**→**Workstations auflisten** aus.
- Im Composer führen Sie den Befehl `composer list ws` aus.

Wenn Sie feststellen wollen, welche Ihrer Workstations ein Agent für z/OS ist, oder wenn Sie die Liste filtern wollen, sehen Sie sich den Betriebssystemtyp an. Dieser ist für diese Agenten wie folgt aufgelistet:

- z/OS in Dynamic Workload Console
- Z im Composer

Im Composer müssen Sie Folgendes festlegen, wenn die Spalte OS Type in der Ausgabe des Befehls **list** sichtbar sein soll:

```
MAESTROCOLUMNNS=120
```

Dies muss vor der Ausführung des Befehls erfolgen.

Wenn Sie beispielsweise den Befehl

```
composer list ws=@
```

in dieser Domäne ausführen, zeigt die Liste der Workstations, dass die Workstations MAT229 und MAT229_1 Agenten für z/OS sind.

WorkstationName	Type	Domain	OSType	Ignored	Updated By	Updated On	Locked By	Locked On
RAL15062	manager	MASTERDM	W		masterad	09/28/2010	-	-
RAL15062_1	agent	-	W		ResourceAdvisorA	10/04/2010	-	-
RAL15062_DWB	broker	MASTERDM	0		masterad	09/28/2010	-	-
MAT229	agent	-	Z		ResourceAdvisorA	10/01/2010	-	-
MAT229_1	agent	-	Z		ResourceAdvisorA	10/01/2010	-	-
AWSBIA291I Total objects: 5								

Jobs definieren

Ebenso wie alle anderen IBM Workload Scheduler-Jobs können Sie Jobs für einen Agenten für z/OS entweder über die Composer-Befehlszeile oder über Dynamic Workload Console definieren.

Da alle Jobs des Agenten für z/OS über Dynamic Workload Broker übergeben werden, wird der Taskabschnitt ihrer Jobdefinitionen in der JSDL-XML-Sprache angegeben. Wenn Sie den Job über Dynamic Workload Console definieren, werden die Informationen, die Sie in den Eingabeanzeigen angeben, automatisch in JSDL umgewandelt.

Aufgrund der folgenden Merkmale gilt ein Job als Job eines Agenten für z/OS:

- Die ausführende Workstation ist ein Agent für z/OS.
- In der JSDL-Definition in Composer hat das Schlüsselwort `application` den Namen `jc1`. In Dynamic Workload Console lautet der Jobdefinitionstyp `z/OS`.

- Die Jobdefinition in JSDL enthält entweder die JCL, die von JES*n* ausgeführt werden soll, oder sie verweist auf die Position der JCL (Dateiname und Membername) im z/OS-System. Im ersten Fall wird die JCL als nach Definition übergeben bezeichnet, im zweiten als nach Verweis übergeben.

Während die JCLs, die nach Definition angegeben werden, Teil des Jobs des Agenten für z/OS sind und als solche in der IBM Workload Scheduler-Datenbank gespeichert werden, sind die JCLs, die nach Verweis angegeben werden, in Dateien des z/OS-Systems gespeichert, auf dem sich der Agent befindet, und werden bei der Übergabe mithilfe der in den entsprechenden Jobs angegebenen Koordinaten abgerufen.

Jobs in Dynamic Workload Console definieren

Die einfachste Möglichkeit, einen Job des Agenten für z/OS zu definieren, ist die Verwendung von Dynamic Workload Console.

Gehen Sie wie folgt vor, um eine Jobdefinition zu erstellen:

1. Klicken Sie in der Navigationsleiste auf **Verwaltung>Auslastungsentwurf>Auslastungsdefinitionen verwalten**.
2. Wählen Sie einen Enginamen aus, geben Sie Ihre Berechtigungsnachweise ein, falls erforderlich, und klicken Sie auf **Los**.
3. Klicken Sie in der Symbolleiste der Arbeitsliste im daraufhin angezeigten Pop-up-Fenster auf **Neu>Jobdefinition>Nativ>z/OS**.
Das Eigenschaftfenster für den Job wird geöffnet.
4. Au der Registerkarte 'Allgemein':
 - a. Geben Sie Folgendes ein:
 - Den Namen des Jobs.
 - Den Namen der Workstation des Agenten für z/OS.
 - Wählen Sie das Kontrollkästchen **Variablenuflösung zur Laufzeit** aus, wenn die JCL (nach Verweis oder nach Definition) Variablen enthält, die durch den Agenten aufgelöst werden müssen, bevor die JCL an JES weitergegeben wird.
 - Geben Sie optional Ausgabebedingungen an. Definieren Sie, welche Rückgabecodes den Job als erfolgreich abgeschlossen markieren, und andere Ausgabebedingungen, die nicht zum Status SUCC führen, aber festlegen, welcher Nachfolgejob als nächster ausgeführt werden soll.

Erfolgsbedingungen für die Ausgabe

Eine Bedingung, die angibt, dass der vorangegangene Job erfolgreich abgeschlossen wurde, wenn sie erfüllt ist. Der Jobstatus wird auf SUCC gesetzt. Erfolgsbedingungen für die Ausgabe können als Rückgabecodes, Jobstatus, Ausgabevariablen oder auf der Basis des Inhalts von Jobprotokollen angegeben werden.

Bedingungsname

Geben Sie einen Namen an, der die Erfolgsbedingung angibt, die von dem vorangegangenen Job erfüllt sein muss, bevor ein nachfolgender Job ausgeführt werden kann.

Bedingungswert

Geben Sie den Wert der Bedingung an, der ein erfolgreiches Ergebnis für den vorangegangenen Job bezeichnet.

Beispiele für eine Erfolgsbedingung für die Ausgabe: **Bedingungsname** STATUS_OK und **Bedingungswert** RC=0

Andere Bedingungen

Ausgabebedingungen, die nicht zum Status SUCC führen und festlegen, ob ein nachfolgender Job ausgeführt wird. Bedingungen können als Rückgabecodes, Jobstatus, Ausgabevariablen oder auf der Basis des Inhalts von Jobprotokollen angegeben werden.

Bedingungsname

Geben Sie einen Namen an, der die Bedingung angibt, die von dem vorangegangenen Job erfüllt sein muss, bevor ein nachfolgender Job ausgeführt werden kann.

Bedingungswert

Geben Sie den Wert der Bedingung an, die von dem vorangegangenen Job erfüllt sein muss, bevor ein nachfolgender Job ausgeführt werden kann.

Beispiel: Sie möchten eine Bedingung erstellen, die bezeichnet, dass der vorangegangene Job mit Fehlern beendet wurde. Sie können Ihre Ausgabebedingung wie folgt definieren: **Bedingungsname** STATUS_ERR1 und **Bedingungswert** RC=2

Das Format von **Bedingungswert** lautet sowohl für Erfolgsbedingungen für die Ausgabe als auch für andere Bedingungen wie folgt: (RC <operator> <operand>). Hierbei gilt Folgendes:

RC Das Schlüsselwort der Anweisung.

Operator

Der Vergleichsoperator. Zulässige Operatoren sind Vergleichsoperatoren (=, != oder <>, >, >=, <, <=), die mit logischen Operatoren (AND, OR, NOT) kombiniert werden können.

Operand

Eine beliebige ganze Zahl aus dem Bereich von -2147483647 bis 2147483647.

Sie können beispielsweise die folgenden Ausdrücke eingeben:

Erfolgsbedingungen für die Ausgabe

- (RC<=3) markiert einen Job als erfolgreich, wenn der Job mit einem Rückgabecode kleiner-gleich 3 beendet wird.
- NOT ((RC=0) AND (RC=1)) markiert einen Job als erfolgreich, wenn der Job mit einem anderen Rückgabecode als 0 und 1 beendet wird.
- (RC=2) OR (RC=4) markiert einen Job als erfolgreich, wenn der Job mit einem Rückgabecode gleich 2 oder gleich 4 beendet wird.
- (RC<7) AND (RC!= 5) markiert einen Job als erfolgreich, wenn der Job mit einem Rückgabecode kleiner als 7 und ungleich 5 beendet wird.
-

Andere Bedingungen

(RC=1) für eine Bedingung mit dem Namen STATUS_ERR.

(RC=4 OR RC=9) für eine Bedingung mit dem Namen FIRST_PATH.

(RC <>5) OR (RC > 2) für eine Bedingung mit dem Namen SECOND_FLOW.

Sie können im Feld **Bedingungswert** sowohl für erfolgreiche Bedingungen als auch für andere Ausgabebedingungen die Ausgabebedingung mithilfe von anderen Variablen als dem Rückgabecode angeben. Sie können z. B. wie folgt drei unterschiedliche Ausgabebedingungen angeben:

- **Bedingungsname:** STATUS_ERR **Bedingungswert:** RC=0
- **Bedingungsname:** STATUS_ERR1 **Bedingungswert:** RC=\${varname}
- **Bedingungsname:** STATUS_ERR2 **Bedingungswert:** RC=\${LOG.CONTENT}
- Sie können eine Erfolgsbedingung oder eine andere Ausgabebedingung für den Job festlegen, indem Sie die Jobausgabe analysieren. Zur Analyse der Jobausgabe müssen Sie die Variable `this.stdlist` überprüfen.

Geben Sie z. B. den folgenden Ausdruck ein:

```
contains(${this.stdlist},"error")
```

falls Sie einen Job als nicht erfolgreich markieren wollen, wenn das Wort "error" in der Jobausgabe enthalten ist.

- Speziell für einen Dateiübertragungsjob können Sie eine Erfolgsbedingung oder Erfolglosigkeitsbedingung für den Job festlegen, indem Sie die Jobeigenschaften analysieren.

Geben Sie z. B. den folgenden Ausdruck ein:

```
${this.File.1.Size}>0
```

wenn Sie einen Dateiübertragungsjob als erfolgreich angeben wollen, falls die Größe der übertragenen Datei größer als null ist.

- Speziell für einen Dateiübertragungsjob können Sie eine Erfolgsbedingung oder Erfolglosigkeitsbedingung für den Job festlegen, indem Sie die Jobeigenschaften oder die Jobausgabe eines anderen Jobs im selben Jobstrom analysieren.

Geben Sie z. B. den folgenden Ausdruck ein:

```
${this.NumberOfTransferredFiles}=  
  ${job.DOWNLOAD.NumberOfTransferredFiles}
```

wenn Sie einen Dateiübertragungsjob als erfolgreich angeben wollen, falls die Anzahl hochgeladener Dateien im Job gleich der Anzahl heruntergeladener Dateien in einem anderen Job mit dem Namen DOWNLOAD im selben Jobstrom ist.

- Alle Xpath-Funktionen und -Ausdrücke (Xpath - XML Path Language) werden für die obigen Bedingungen im Feld **Bedingungswert** unterstützt:
 - Zeichenfolgevergleiche (contains, starts-with, matches usw.)
 - Bearbeitung von Zeichenfolgen (concat, substring, uppercase usw.)
 - Numerischer Vergleich (=, !=, > usw.)
 - Funktionen für numerische Werte (abs, floor, round usw.)
 - Operatoren für numerische Werte (add, sum, div usw.)
 - Boolesche Operatoren

Die folgende Abbildung zeigt die Seite 'Allgemein' mit der Definition eines Jobs vom Typ z/OS.

Properties - z/OS (9.3.0.01) - NC050024_1#JCLJOBREF

General Affinity Recovery Options z/OS

* Name: JCLJOBREF * Workstation: NC050024_1

Description:

Variable resolution at runtime

Successful Output Conditions (force the job status to successful):

Condition Name	Condition Value
+	

Other Output Conditions (do not change the job status):

Condition Name	Condition Value
+	

Abbildung 3. Seite 'Allgemein' mit einer neuen Definition für den Agenten für z/OS

5. Geben Sie optional weitere Informationen auf den Seiten 'Affinität' und 'Wiederanlaufoptionen' ein.

JCL nach Verweis definieren

Wenn Sie einen Job definieren, der nur die Position der JCL, die übergeben werden soll, referenzieren muss, wählen Sie auf der Seite z/OS das Optionsfeld **nach Verweis** aus. Geben Sie anschließend die folgenden Informationen ein:

- Den Namen der Datei, in der die JCL gespeichert ist. Dieser Name kann bis zu 44 Zeichen lang sein und ist optional. Wenn der Dateiname nicht angegeben wurde, sucht der Agent für z/OS in der Dateiverkettungsbibliothek, die während der Installation für den Agenten deklariert wurde, nach dem Membernamen.
- Den Namen des JCL-Members in der Datei. Dieser Name kann bis zu 8 Zeichen lang sein und ist erforderlich.

Die folgende Abbildung zeigt die Seite 'z/OS', auf der die Option **nach Verweis** ausgewählt wurde:

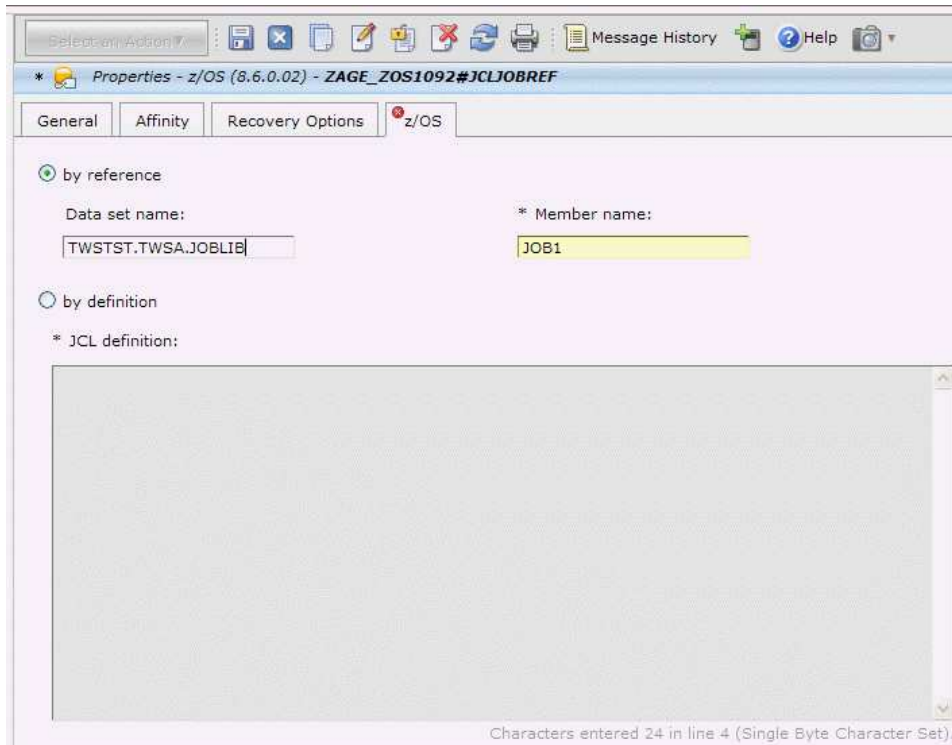


Abbildung 4. Nach Verweis angegebene JCL

JCL nach Definition definieren

Wenn die gesamte JCL der Jobdefinition hinzugefügt werden soll, wählen Sie auf der Seite 'z/OS' das Optionsfeld **nach Definition** aus und schreiben die JCL-Anweisung in das Feld **JCL-Definition**.

Die folgende Abbildung zeigt die Seite 'z/OS', auf der die Option **nach Definition** ausgewählt ist:

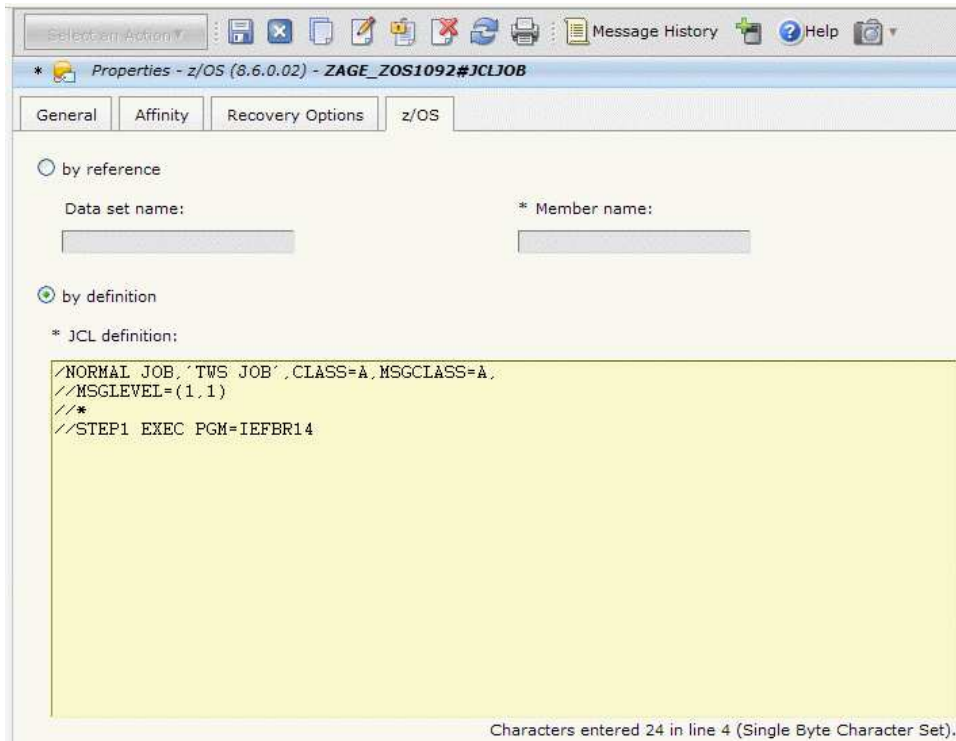


Abbildung 5. Nach Definition angegebene JCL

Definition in Composer

Sie können auch die Composer-Befehlszeile zum Definieren von Jobs für den Agenten für z/OS verwenden.

Das JSDL-Tagging ändert sich abhängig davon, ob die JCL im Job, die für JES vorgesehen ist, nach Verweis oder nach Definition definiert wird.

Jobs definieren, die auf die JCL-Position in z/OS verweisen

Das folgende Beispiel zeigt die Definition eines Jobs mit dem Namen JCLJOBREF.

Die Workstation ist ein Agent für z/OS mit dem Namen ZAGE_ZOS1092.

Die JSDL-Codierung verweist auf die JCL-Position im z/OS-System.

```
ZAGE_ZOS1092#JCLJOBREF
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jmgr:submitJobFromJSDL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:jmgr="http://www.abc.com/xmlns/prod/scheduling/1.0/JobManager"
  xmlns:jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdl"
  xmlns:jsdljcl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdljcl"
  xmlns:sdo="http://www.eclipse.org/emf/2003/SDO">
  <jmgr:JobDefinitionDocument>
    <jsdl:jobDefinition name="JCL">
      <jsdl:application name="jcl">
        <jsdljcl:jcl xsi:type="sdo:EDataObjectAnyType">
          <jsdljcl:JCLParameters xsi:type="sdo:EDataObjectAnyType">
            <jsdljcl:jcl xsi:type="sdo:EDataObjectAnyType">
              <jsdljcl:byRefOrByDef xsi:type="sdo:EDataObjectAnyType">
                <jsdljcl:byReference xsi:type="sdo:EDataObjectAnyType">
                  <jsdljcl:dataset xsi:type="sdo:EDataObjectAnyType">TWSTST.TWSA.JOBLIB</jsdljcl:dataset>
                  <jsdljcl:member xsi:type="sdo:EDataObjectAnyType">JOB1</jsdljcl:member>
                </jsdljcl:byReference>
              </jsdljcl:byRefOrByDef>
            </jsdljcl:jcl>
          </jsdljcl:JCLParameters>
        </jsdljcl:jcl>
      </jsdl:application>
    </jsdl:jobDefinition>
  </jmgr:JobDefinitionDocument>
</submitJobFromJSDL>
```

```

</jsdljcl:jcl>
</jsdljcl:JCLParameters>
<jsdljcl:JOBParameters xsi:type="sdo:EDataObjectAnyType">
  <jsdljcl:jobStreamName xsi:type="sdo:EDataObjectAnyType">JOBS</jsdljcl:jobStreamName>
  <jsdljcl:inputArrival xsi:type="sdo:EDataObjectAnyType">201206131200
</jsdljcl:inputArrival>
</jsdljcl:JOBParameters>
</jsdljcl:jcl>
</jsdl:application>
<jsdl:resources/>
</jsdl:jobDefinition>
</jmgr:JobDefinitionDocument>
<jmgr:Alias>ZA86_ZOS1354#JOBS.PROVA.JNUM-622656411</jmgr:Alias>
<jmgr:JobId>5e2efa42-1dab-31eb-a8f1-6aaa413a4cec</jmgr:JobId>
<jmgr:ClientNotifyURI>https://ts6087.enervt.com:31116/JobManagerRESTWeb/JobScheduler/job
</jmgr:ClientNotifyURI>
<jmgr:ClientNotifyURI>https://ts6087.enervt.com:31116/JobManagerRESTWeb/JobScheduler/job
</jmgr:ClientNotifyURI>
</jmgr:submitJobFromJSDL>

```

Beachten Sie die folgenden Schlüsselwörter:

<jsdl:application name="jcl">

Gibt an, dass der Job ein Job des Agenten für z/OS ist.

<jsdljcl:byReference>

Gibt an, dass sich die JCL, die von JES ausgeführt wird, im z/OS-System befindet und nur die Koordinaten ihrer Position hier angegeben werden.

<jsdljcl:dataset xsi:type="sdo:EDataObjectAnyType">...</jsdljcl:dataset>

Gibt den Namen der Datei an, in der die JCL gespeichert ist. Dieser Name kann bis zu 44 Zeichen lang sein und ist optional. Wenn der Dateiname nicht angegeben wurde, sucht der Agent für z/OS in der Dateiverkettungs-bibliothek, die während der Installation für den Agenten deklariert wurde, nach dem Membernamen.

Wichtig: Schreiben Sie das Schlüsselwort auch, wenn Sie keinen Dateinamen angeben. Das Element muss vorhanden sein, unabhängig davon, ob ein Wert angegeben wird oder nicht.

<jsdljcl:member xsi:type="sdo:EDataObjectAnyType">...</jsdljcl:member>

Gibt den Namen des JCL-Members in der Datei an. Dieser Name kann bis zu 8 Zeichen lang sein und ist erforderlich.

<jsdljcl:JOBParameters>

In diesem Abschnitt müssen Sie Folgendes angeben:

- Jobstromname
- Eingangszeit

Jobs definieren, die die JCL-Definition enthalten

Das folgende Beispiel zeigt die Definition eines Jobs mit dem Namen JCLJOB.

Die Workstation ist ein Agent für z/OS mit dem Namen ZAGE_ZOS1092.

Die JCL-Anweisung ist in die JSDL-Definition eingebettet.

```

ZAGE_ZOS1092#JCLJOB
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdl"
xmlns:jsdljcl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdljcl">
<jsdl:application name="jcl">
  <jsdljcl:jcl>
    <jsdljcl:JCLParameters>
      <jsdljcl:jcl>
        <jsdljcl:byRefOrByDef>

```

```

<jSDLjcl:byDefinition>
  <jSDLjcl:jclDefinition>
    /NORMAL JOB,'TWS JOB',CLASS=A,MSGCLASS=A,
    //MSGLEVEL=(1,1)
    /**
    //STEP1 EXEC PGM=IEFBRI4
  </jSDLjcl:jclDefinition>
</jSDLjcl:byDefinition>
</jSDLjcl:byRefOrByDef>
</jSDLjcl:jcl>
</jSDLjcl:JCLParameters>
<jSDLjcl:JOBParameters>
  <jSDLjcl:jobStreamName>${tws.jobstream.name}</jSDLjcl:jobStreamName>
  <jSDLjcl:inputArrival>${tws.job.ia}</jSDLjcl:inputArrival>
</jSDLjcl:JOBParameters>
</jSDLjcl:jcl>
</jSDL:application>
</jSDL:jobDefinition>
DESCRIPTION "Sample JCL Job Definition"

```

Beachten Sie die folgenden Schlüsselwörter:

<jSDL:application name="jcl">

Gibt an, dass der Job ein Job des Agenten für z/OS ist.

<jSDLjcl:byDefinition>

Gibt an, dass die JCL, die von JES ausgeführt wird, innerhalb der JSDL definiert wird.

<jSDLjcl:jclDefinition>

Enthält die gesamte JCL-Definition.

<jSDLjcl:JOBParameters>

In diesem Abschnitt müssen Sie Folgendes angeben:

- Jobstromname
- Eingangszeit

In diesem speziellen Beispiel werden die tatsächlichen Werte durch IBM Workload Scheduler-Variablen ersetzt. Diese werden von Dynamic Workload Broker bei der Übergabe aufgelöst.

Angeben, dass die JCL Variablen enthält, die bei der Ausführung aufgelöst werden müssen

Wenn die JCL Variablen enthält, die bei der Ausführung durch den Agenten aufgelöst werden, muss dies im Abschnitt `jobDefinition` der JSDL-Definition deklariert werden, und zwar unabhängig davon, ob sich die JCL in der Jobdefinition befindet oder durch ihre Position in z/OS referenziert wird.

Das folgende Beispiel zeigt dieselbe Definition wie in „Jobs definieren, die die JCL-Definition enthalten“ auf Seite 59, jedoch mit dem hinzugefügten Schlüsselwort, das angibt, dass Variablen vorhanden sind, die bei der Ausführung aufgelöst werden:

```

ZAGE_ZOS1092#JCLJOB
TASK
<?xml version="1.0" encoding="UTF-8"?>
<jSDL:jobDefinition xmlns:jSDL="http://www.abc.com/xmlns/prod/scheduling/1.0/jSDL"
xmlns:jSDLjcl="http://www.abc.com/xmlns/prod/scheduling/1.0/jSDLjcl"
XMLSchema:text="resolveVariableTable" name="jcl">
<jSDL:application name="jcl">
  <jSDLjcl:jcl>
    <jSDLjcl:JCLParameters>
  </jSDLjcl:jcl>
  <jSDLjcl:byRefOrByDef>
  <jSDLjcl:byDefinition>
    <jSDLjcl:jclDefinition>
      /NORMAL JOB,'TWS JOB',CLASS=A,MSGCLASS=A,
      //MSGLEVEL=(1,1)

```



```

    /**
    //STEP1 EXEC PGM=&MODULE
    </jsdljcl:jclDefinition>
    </jsdljcl:byDefinition>
    </jsdljcl:byRefOrByDef>
    </jsdljcl:jcl>
    </jsdljcl:JCLParameters>
    <jsdljcl:JOBParameters>
    <jsdljcl:jobStreamName>${tws.jobstream.name}</jsdljcl:jobStreamName>
    <jsdljcl:inputArrival>${tws.job.ia}</jsdljcl:inputArrival>
    </jsdljcl:JOBParameters>
    </jsdljcl:jcl>
    </jsdl:application>
    </jsdl:jobDefinition>
    DESCRIPTION "Sample JCL Job Definition"

```

Dabei gilt Folgendes:

XMLSchema:text="resolveVariableTable"

Gibt an, dass in der JCL Variablen vorhanden sind, die bei der Ausführung aufgelöst werden müssen. Wenn Sie den Job an den Agenten für z/OS übergeben, wird die Definition zusammen mit der Tabelle, die die Variable und den entsprechenden Wert enthält, an den Agenten übertragen. Dort wird sie verarbeitet, bevor sie an JES weitergegeben wird.

Achtung: Verwenden Sie dieses Schlüsselwort nicht unsachgemäß. Wenn keine Übereinstimmung zwischen dem Schlüsselwort und dem Vorhandensein bzw. Nichtvorhandensein der Variablen besteht, schlägt der Job fehl.

JCL definieren

Vorgehensweise beim Definieren der JCL.

Wenn Sie die JCLs in Dynamic Workload Console oder in der Composer-Befehlszeile definieren, ist es wichtig, dass Sie Folgendes beachten:

- In der JOB-Karte darf der Name des Programmierers maximal 19, nicht 20 Zeichen lang sein, da das letzte Zeichen (HEX '30x') für die interne Verwendung reserviert ist. Wenn der Agent einen längeren Namen des Programmierers feststellt, wird der Job nicht verarbeitet und eine Fehlernachricht wird ausgegeben.
- Der Name der JCL muss den folgenden Syntaxregeln entsprechen. Der Name muss die folgenden Eigenschaften aufweisen:
 - Er muss in Spalte 3 beginnen.
 - Er muss aus 1 bis 8 alphanumerischen Zeichen (Großbuchstaben von A bis Z, Zahlen von 0 bis 9) oder nationalen Sonderzeichen (\$, #, @) bestehen.

Anmerkung: Das System erkennt die folgenden hexadezimalen Darstellungen der US-amerikanischen nationalen Sonderzeichen im EBCDIC-Format:

- \$ (Dollarzeichen) als X'5B'
- # (Nummernzeichen) als X'7B'
- @ (kommerzielles A) als X'7C'

In Ländern außerhalb der USA können die US-amerikanischen nationalen Sonderzeichen, die über Terminaltastaturen eingegeben werden, eine andere hexadezimale Darstellung generieren und einen Fehler verursachen.

Beispiel: In einigen Ländern generiert die Eingabe des Zeichens \$ X'4A'. Dies bedeutet, dass Sie je nach der Codepage, die mit dem Parameter CODEPAGE der Initialisierungsanweisung TWSOPTS angegeben ist, diejenigen Zeichen verwenden müssen, die HEX X'7C', X'5B' und X'7B' im EBCDIC-Format ent-

sprechen. Beispiel: Ist IBM_280 in TWSOPTS als Systemcodepage angegeben, können Sie im Jobnamen das Pfundzeichen (£) verwenden, das als Hex X'7B' codiert wird.

- Er muss mit einem Buchstaben oder einem nationalen Sonderzeichen, jedoch nicht mit einer Zahl beginnen.
- Er muss von mindestens einem Leerzeichen gefolgt sein.

HFS- oder ZFS-Dateiauslösung

APAR PI09318 stellt Ihnen ein Dienstprogramm zur Dateiüberwachung bereit. Nach der Installation des APARs finden Sie dieses Dienstprogramm als Lademodul EELFLWAT in der Bibliothek SEELMD0. Mithilfe dieses Programms können Sie nach Dateisystemänderungen von HFS- oder ZFS-Dateien und -Verzeichnissen suchen, wenn Sie beispielsweise vor dem Ausführen eines Jobs, der eine Datei verarbeitet, sicherstellen wollen, dass diese Datei vorhanden ist. Durch das Definieren eines Jobs, der dieses Dienstprogramm ausführt, können Sie eine Dateiabhängigkeit implementieren, d. h. eine Beziehung zwischen einer Datei und einer Operation, in der eine bestimmte Aktivität für die Datei über den Operationsstart entscheidet. Beispielsweise können Sie den Job, der EELFLWAT ausführt, als vorangegangenen Job für die von der Datei abhängige Operation definieren.

Der Benutzer, der EELFLWAT ausführt, muss über Ausführungszugriff (x) für den Verzeichnispfad verfügen, der zu überwachende Dateien enthält.

Die Bibliothek SEELSAMP enthält das Member EELFLWAS als Beispiel-JCL zum Aufrufen des Dienstprogramms zur Dateiüberwachung.

Syntax

Wie im Member EELFLWAS der Bibliothek SEELSAMP gezeigt, wird das folgende Format für den Parameter der Anweisung EXEC unterstützt, in der PGM=EELFLWAT angegeben ist:

```
//FLWATCH EXEC PGM=EQQLWAT,PARM='ENVAR()/-c bedingungswert -dea endtermin  
// -fi dateipfad -i intervall -r rc -t tracestufe'
```

Beachten Sie Folgendes:

- Es wird kein Fortsetzungszeichen verwendet. Wenn Sie das Feld PARM fortsetzen wollen, unterbrechen Sie es in Spalte 72 und fahren Sie in Spalte 16 der nächsten Karte fort.
- Trennen Sie Argumente und Werte durch ein Leerzeichen.
- Übergeben Sie Umgebungsvariablen mithilfe von ENVAR() an EELFLWAT.
- EELFLWAT gibt Nachrichten und Traceinformationen (bei Bedarf) im Protokoll für den Job zurück, mit dem das Dienstprogramm ausgeführt wird. Wenn Sie die Abruffunktion für Jobprotokolle verwenden, müssen Sie das Feld **Ben.-System** in den Bereinigungsoptionen auf Operationsebene auf J setzen, und zwar in der Datenbank und im aktuellen Plan.
- Die Argumente sind nicht positionsgebunden.
- Sie können ein abgekürztes Format für alle Argumente verwenden. Im Allgemeinen können Sie die Argumente bis zu einer beliebigen Position nach dem ersten Zeichen abschneiden. Eine Ausnahme ist das Argument `deadline`, für das mindestens drei Zeichen erforderlich sind.

Argumente

-condition | -c

Die zu prüfende Bedingung. Gültige Werte sind:

wcr | waitCreated

Wartet, bis die Datei vorhanden ist. Wenn die Datei bereits vorhanden ist, wird **filewatch** sofort beendet. Wenn mit dem Argument **-filename** ein Verzeichnis angegeben wird, wartet der Prozess, bis das Verzeichnis vorhanden ist und eine neue Datei enthält.

wmr | waitModificationRunning

Wartet, bis sich die Dateigröße oder die Änderungszeit ändert. Wenn mit dem Argument **-filename** ein Verzeichnis angegeben wird, wartet der Prozess, bis sich beim Erstellen, Modifizieren oder Löschen einer Datei die Größe der Datei oder die Änderungszeit früherer Dateien ändert.

wmc | waitModificationCompleted

Prüft, ob das Ändern der Dateigröße oder der Änderungszeit gestoppt hat, d. h., dass **filewatch** auf drei Suchintervalle ohne Änderungen wartet. Wenn mit dem Argument **-filename** ein Verzeichnis angegeben wird, überprüft der Prozess die Änderung der Größe oder der Änderungszeit von früheren Dateien, z. B. ob sich die Anzahl Dateien im Verzeichnis und die Änderungszeit von früheren Dateien in drei Suchintervallen nicht ändert.

wmrc | waitModificationRunningCompleted

Wartet, bis sich die Dateigröße oder die Änderungszeit ändert und diese Änderung stoppt, d. h., dass **filewatch** nach der ersten Änderung auf drei Suchintervalle ohne weitere Änderungen wartet. Wenn mit dem Argument **-filename** ein Verzeichnis angegeben wird, überprüft der Prozess die Änderung der Größe oder der Änderungszeit von früheren Dateien, z. B. ob sich die Anzahl Dateien im Verzeichnis und die Änderungszeit von früheren Dateien in drei Suchintervallen nicht ändert.

wdl | waitDelete

Stoppt die Ausführung, wenn die Datei gelöscht wird. Wenn mit dem Argument **-filename** ein Verzeichnis angegeben wird, wartet der Prozess, bis eine Datei aus dem Verzeichnis gelöscht wird.

-deadline | -dea

Der Zeitraum bis zum Endtermin, ausgedrückt in Sekunden. Gültige Formate sind:

- Eine ganze Zahl zwischen 0 und 31.536.000 (der höhere Wert entspricht einem Jahr). Geben Sie 0 an, wenn **filewatch** in einer Endlosschleife ausgeführt werden soll.
- *hh:mm:ss* im Bereich zwischen 00:00:01 und 24:00:00, um eine Zeit desselben Tages auszuwählen, an dem **filewatch** gestartet wurde.

Diese Angabe entspricht dem GMT-Wert (Greenwich Mean Time, mittlere Greenwich-Zeit). Zur Angabe eines Werts, der sich vom GMT-Wert unterscheidet, verwenden Sie den Parameter **ENVAR** wie in „Beispiel“ auf Seite 64 gezeigt. Detaillierte Informationen zur Einstellung der Umgebungsvariablen **TZ** finden Sie im Handbuch *UNIX System Services Command Reference*.

-filename | -fi

Der zu verarbeitende Dateipfad. Sie können Leerzeichen oder Sonderzei-

chen einbetten, indem Sie Anführungszeichen verwenden. Platzhalterzeichen werden nicht unterstützt. Wenn Sie mehrere Dateien in den Überwachungsprozess einschließen wollen, können Sie die Dateien in einem Verzeichnis speichern und dieses Verzeichnis über einen Dateipfad angeben.

-interval | -i

Das Dateisuchintervall, ausgedrückt in Sekunden. Geben Sie eine ganze Zahl im folgenden Bereich an:

- 5 – 3600, wenn **wcr** oder **wdl** als Bedingungswert angegeben wird.
- Andernfalls 30 – 3600.

Der Standardwert ist 60.

-returncode | -rc

Der Rückgabecode des Exits, wenn die Datei bis zum Endtermin nicht gefunden wird. Geben Sie eine ganze Zahl zwischen 0 und 255 an. Der Wert des Rückgabecodes wird ignoriert, wenn Sie 0 als Endterminwert angeben. Der Standardwert ist 4.

-trace | -t

Tracestufe für interne Protokollierung und Traces. Mögliche Werte:

- 0 Ausschließlich Fehlernachrichten empfangen.
- 1 Gibt die Stufe *Fine* an, um die wichtigsten Nachrichten mit dem geringsten Umfang zu empfangen.
- 2 Gibt die Stufe *Finer* an, um Eingangstraces und Exit-Traces zu aktivieren.
- 3 Gibt die Stufe *Finest* an, um die detaillierteste Traceausgabe zu empfangen.

Der Standardwert ist 0.

Sie finden die Traceausgabe im Protokoll für den Job, der **filewatch** ausführt.

Beispiel

```
//FLWATCH EXEC PGM=EELFLWAT,PARM='ENVAR(TZ=GMT-1CET)'/-co wcr -dead 30  
// -fi /u/falsi/prova -int 30 -t 2'
```

In diesem Beispiel wird die Zeitzone mithilfe von `ENVAR(GMT-1CET)` in Mitteleuropäische Zeit (CET - Central European Time) geändert. GMT-1 ist der Zeitonenwert und CET ist die ausgewählte Sommerzeit.

Jobs übergeben

Alle Jobs des Agenten für z/OS können entweder Teil eines Jobstroms sein und in einem Plan übergeben werden oder jederzeit mit den Befehlen `conman submit` oder den entsprechenden Dynamic Workload Console-Anzeigen übergeben werden.

Übergeben Sie Jobs des Agenten für z/OS ebenso wie alle anderen IBM Workload Scheduler-Jobs.

Wenn Sie einen Job des Typs JCL über einen Produktionsplan oder einzeln über eine `conman`-Befehlszeile bzw. die Fenster **Übergeben** von Dynamic Workload Console übergeben, wird der Job von Dynamic Workload Broker verarbeitet und an den Agenten für z/OS gesendet, der in der Jobdefinition angegeben ist.

Der Agent empfängt die Jobübergabeanforderungen. Die Jobübergabeanforderungen enthalten entweder den Hauptteil der JCL, der zur Ausführung an JES*n* weitergegeben werden soll, oder einen Verweis auf ein Member einer partitionierten Datei, die die JCL enthält. Wenn der Verweis nur das Member, jedoch nicht die Datei nennt, sucht der Agent den Membernamen in der Dateiverkettungsbibliothek, die für den Agenten bei der Installation deklariert wurde (Standardwert: EELJBLIB).

Wenn in der JCL die Variablensubstitution angefordert wird, werden die Variablen tabellen, die die in der JCL verwendeten Variablen (und die entsprechenden Werte) enthalten, zusammen mit der Übergabeanforderung an den Agenten gesendet.

Bei Empfang einer Jobübergabeanforderung speichert die HT-Task den Job kurz in der HTREF-Servicedatenbank, die zum Agenten für z/OS gehört. Wenn die Anforderung eine JCL nach Verweis oder eine Variablensubstitution beinhaltet, wird die referenzierte JCL innerhalb von HTREF abgerufen bzw. die Variablen innerhalb von HTREF aufgelöst. Ein Failovermechanismus für den Agenten verfolgt die Jobs, die nicht verarbeitet werden, wenn der Agent aus irgendeinem Grund die Verbindung verliert oder fehlschlägt, und nimmt den Übergabethread wieder auf, sobald die Kommunikation zwischen IBM Workload Scheduler und dem Agenten erneut gestartet wurde. Details zu diesem Prozess finden Sie in „Informationen zu Resynchronisationsnachrichten“ auf Seite 107.

Abb. 6 auf Seite 66 zeigt die Route, die ein Job nimmt, wenn er beim Agenten für z/OS ankommt und zur Verarbeitung an JES weitergegeben wird.

Von Dynamic Workload Broker übergebener Job

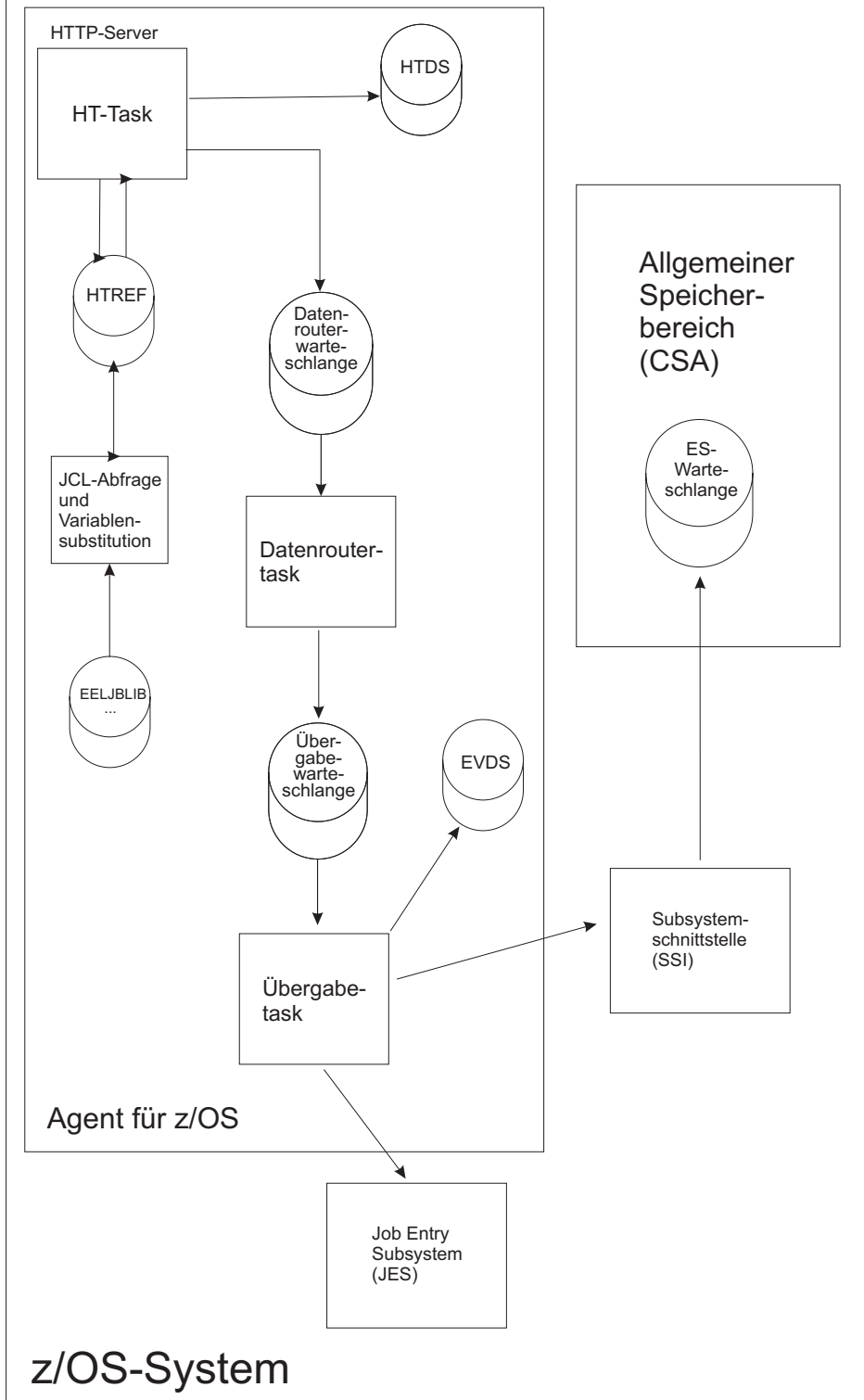


Abbildung 6. Route eines Jobs innerhalb des Agenten für z/OS

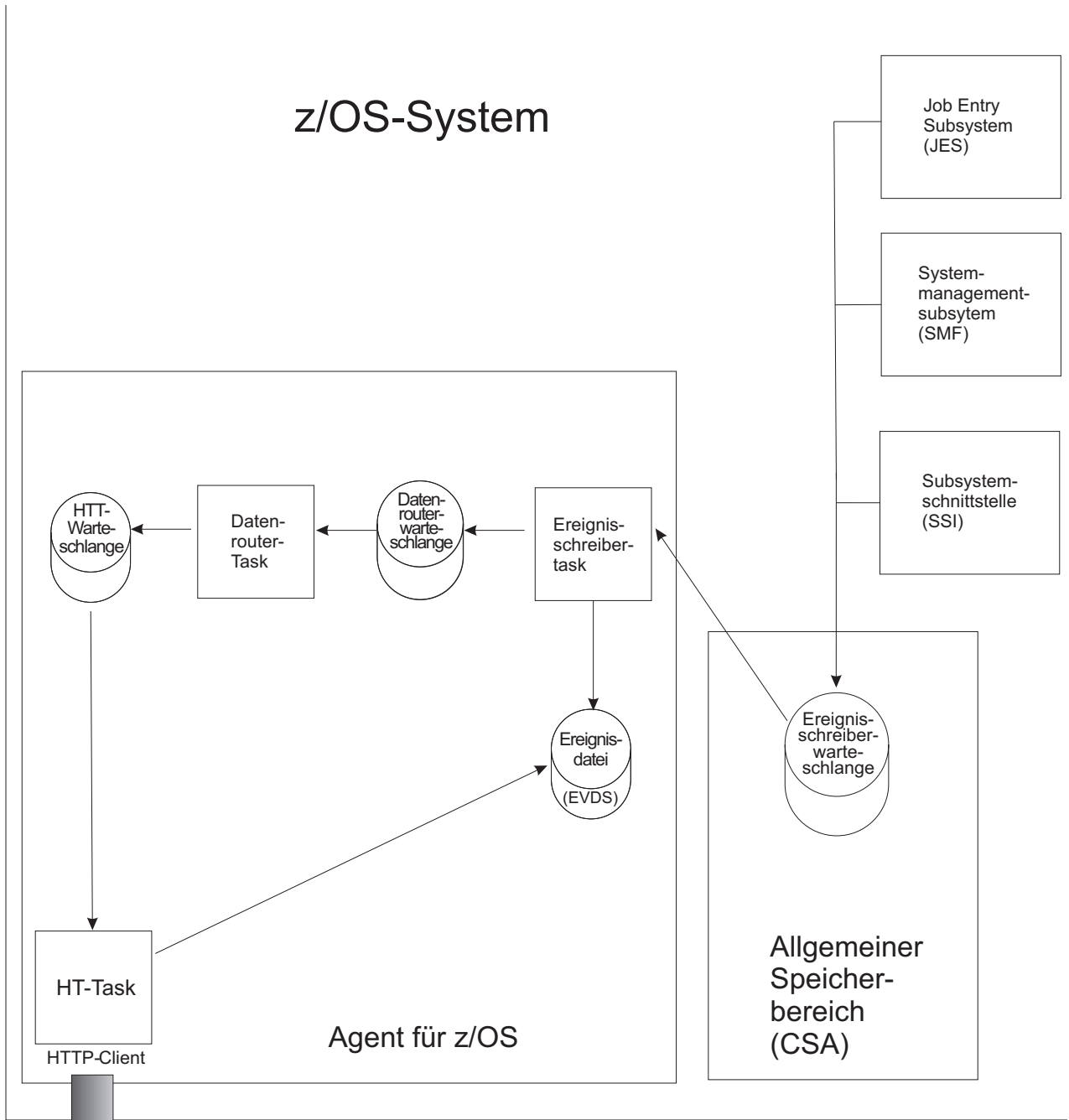
Wenn die JCL bei der Übergabetask ankommt, wird der Job über die Datei EELBRDS an JES übergeben. Die Datei EELBRDS wird verwendet, um einen internen JES-Leser zuzuordnen.

Beachten Sie, dass die JCLs, die von IBM Workload Scheduler stammen, auf Platte (HTDS) gespeichert werden. Auf diese Weise ist sichergestellt, dass die Jobs bei einer Unterbrechung des Agenten nicht verloren gehen, sondern an JES gesendet werden, wenn der Agent fortgesetzt wird.

Nachdem die JCL der Datei EELBRDS hinzugefügt wurde, wird die Ereignisdatei (EVDS) mit dem neuen Jobstatus aktualisiert. Eine Aktualisierung wird außerdem an Dynamic Workload Broker über die z/OS-Subsystemschnittstelle (SSI - Subsystem Interface) gesendet.

Nachdem ein Job verarbeitet wurde, wird sein Status als Ereignis an IBM Workload Scheduler zurückgesendet.

Abb. 7 auf Seite 68 zeigt die Route, die Ereignisse in Bezug auf den Status übergebener Jobs nehmen, wenn sie von JES zurückgegeben, vom Agenten für z/OS verarbeitet und an IBM Workload Scheduler zurückgesendet werden.



Ereignisse werden zurückgesendet

Abbildung 7. Route eines Statusereignisses innerhalb des Agenten für z/OS, wenn es von JES auf dem Weg zu IBM Workload Scheduler zurückgegeben wird

Der Agent verwendet die normalen Überwachungsverfahren auf der Basis von SMF- und JES-Exits, um den Status aller Jobs im System zu überwachen und um die Jobs an Dynamic Workload Broker zurückzusenden.

Beachten Sie, dass die Ereignisse in der Ereignisschreiberwarteschlange im allgemeinen Speicherbereich (CSA - Common Storage Area) und außerdem in der Ereignisdatei (EVDS) auf Platte gespeichert werden. Auf diese Weise ist sichergestellt, dass sie bei einer Unterbrechung des Agenten nicht verloren gehen, sondern an IBM Workload Scheduler gesendet werden, wenn der Agent fortgesetzt wird.

Zurück in IBM Workload Scheduler empfängt die Ereignisfilterkomponente von Dynamic Workload Broker alle Ereignisse, die der Agent für z/OS sendet, und gleicht sie mit einer Jobtabelle ab. Liegt eine Übereinstimmung vor, wird das Ereignis verarbeitet; andernfalls wird es gelöscht.

Der Ausgabemanager verarbeitet Jobprotokollanforderungen von IBM Workload Scheduler, indem er die Jobausgabe aus dem JES-Spoolprogramm abrufen (falls noch vorhanden).

Wird eine Jobübergabeanforderung von einem Agenten für z/OS in einer Sysplex-Umgebung empfangen, führt der Agent den Job aus. Der Job kann jedoch von JES an einen beliebigen anderen Knoten des Sysplex weitergeleitet werden, wo ein anderer Agent den Job überwacht und das Ereignis an IBM Workload Scheduler zurücksendet. Dort wird geprüft, ob das Ereignis zu einem der übergebenen Jobs gehört, und sein Status geändert, wenn dies der Fall ist.

Die Verwendung von NJE (Netzjobeintrag) wird mit dem Agenten für z/OS nicht unterstützt, da dies zu einer fehlerhaften Überwachung des Jobstatus führen kann.

Variablen in Jobs verwenden

Sie können Variablen in Ihre Jobdefinition einschließen.

Die Variablen werden zur Übergabezeit aufgelöst. Je nach dem Ort, wo sie aufgelöst werden, können sie in zwei Typen eingeteilt werden:

- Variablen, die von IBM Workload Scheduler auf der Dynamic Workload Broker-Ebene aufgelöst werden, bevor der Job an den Agenten für z/OS übergeben wird. Sie müssen im JSDL-Teil der Jobdefinition enthalten sein.

Sie können auch Eigenschaftsvariablen oder die Jobausgabe zwischen zwei Jobs in derselben Jobstrominstanz übergeben. Weitere Informationen zu den Eigenschaften, die Sie als Variablen verwenden können, oder zur Übergabe der Jobausgabe finden Sie in „Variablen zwischen Jobs in derselben Jobstrominstanz übergeben“ auf Seite 71.

- Variablen, die vom Agenten für z/OS aufgelöst werden, bevor die JCL an JES übergeben wird. Sie müssen in der JCL eingebettet vorhanden sein oder in der Jobdefinition referenziert werden.

Variablen, die von IBM Workload Scheduler aufgelöst werden

Den Variablen werden ihre Werte durch Dynamic Workload Broker bei der Jobübergabe zugewiesen, bevor der Job an den Agenten für z/OS weitergegeben wird.

Die folgenden Variablen werden unterstützt:

Tabelle 12. Unterstützte Variablen in JSDL-Definitionen

Variablenname	Beschreibung
tws.host.workstation	Name der Host-Workstation
tws.job.date	Datum des übergebenen Jobs
tws.job.fqname	Vollständig qualifizierter Name des Jobs (UNISON_JOB)
tws.job.ia	Eingangszeit des Jobs
tws.job.interactive	Der Job ist interaktiv. Gültige Werte sind true oder false. Nur für abwärtskompatible Jobs anwendbar.
tws.job.logon	Berechtigungsnachweise des Benutzers, der den Job ausführt (LOGIN). Nur für abwärtskompatible Jobs anwendbar.
tws.job.name	Name des übergebenen Jobs
tws.job.num	Nummer des übergebenen Jobs
tws.job.priority	Priorität des übergebenen Jobs
tws.job.promoted	Job ist weitergeleitet. Gültige Werte sind YES oder No. Weitere Informationen zur Weiterleitung dynamischer Jobs finden Sie im Abschnitt zur Weiterleitung von Jobs, die in dynamischen Pools terminiert sind, im Handbuch <i>Dynamische Workloadterminierung</i> .
tws.job.recnum	Datensatznummer des Jobs
tws.job.resourcesForPromoted	Menge der erforderlichen logischen Ressourcen, die einem weitergeleiteten Job in einem dynamischen Pool zugeordnet sind. Gültige Werte sind 1, wenn der Job weitergeleitet ist, oder 10, wenn der Job nicht weitergeleitet ist. Weitere Informationen zur Weiterleitung dynamischer Jobs finden Sie im Abschnitt zur Weiterleitung von Jobs, die in dynamischen Pools terminiert sind, im Handbuch <i>Dynamische Workloadterminierung</i> .
tws.job.workstation	Name der Workstation, auf der der Job definiert ist
tws.jobstream.id	ID des Jobstroms, der den Job enthält (UNISON_SCHED)
tws.jobstream.name	Name des Jobstroms, der den Job enthält (UNISON_SCHED)
tws.jobstream.workstation	Name der Workstation, auf der der Jobstrom definiert ist, der den Job enthält
tws.master.workstation	Name des Masterdomänenmanagers (UNISON_MASTER)
tws.plan.date	Startdatum des Produktionsplans (UNISON_SCHED_DATE)
tws.plan.date.epoch	Startdatum des Produktionsplans im Epochenformat (UNISON_SCHED_EPOCH)
tws.plan.runnumber	Ausführungsnummer des Produktionsplans (UNISON_RUN)

Wenn Sie diese Variablen in die Jobdefinition im Composer einschließen, fügen Sie ein Dollarzeichen (\$) vor der Variablen ein und schließen Sie die Variable in geschweifte Klammern ein; beispielsweise `${twsw.master.workstation}`.

Variablen zwischen Jobs in derselben Jobstrominstanz übergeben

In vielen Szenarios kann die Jobausgabe oder eine Jobeigenschaft des ersten Jobs in einem Jobstrom die Eingabe für die Ausführung der nachfolgenden Jobs in derselben Jobstrominstanz sein. Dies gilt auch für den Jobstrom JOBS.

Im Szenario befinden sich *JobA* und *JobB* in derselben Jobstrominstanz und *JobA* übergibt während der Ausführung einige Variablenwerte an *JobB*. Sie können die folgenden Variablen von *JobA* an *JobB* übergeben:

- *JobA* exportiert einige Eigenschaften und *JobB* referenziert diese Eigenschaften in seiner Definition als Variablen in einem vordefinierten Format. Während der Ausführung werden die Variablen von *JobB* automatisch aufgelöst. Die Jobeigenschaften, die Sie exportieren können, hängen von dem Jobtyp ab, den Sie definieren. Siehe „Jobeigenschaften aus einem Job an anderen Job in derselben Jobstrominstanz übergeben“.
- *JobA* exportiert seinen Standardausgabewert und *JobB* referenziert diese Standardausgabe als Eigenschaft in der Definition von *JobB* als Variable. Während der Ausführung wird die Variable von *JobB* automatisch aufgelöst. Siehe „Jobstandardausgabe aus einem Job an anderen Job in derselben Jobstrominstanz übergeben“ auf Seite 73.
- *JobA* exportiert seinen Standardausgabewert und *JobB* referenziert diese Standardausgabe als seinen Standardeingabewert. Diese Option ist nur für ausführbare Jobs gültig. Siehe „Jobstandardausgabe aus einem Job als Standardeingabe an anderen Job in derselben Jobstrominstanz übergeben“ auf Seite 74.

Anmerkung: Der Jobstrom USERJOBS, der von IBM Workload Scheduler-Prozessen erstellt wird, unterstützt nicht die Übergabe von Variablen zwischen Jobs, die zu ihm gehören.

Jobeigenschaften aus einem Job an anderen Job in derselben Jobstrominstanz übergeben:

Sie können einige Jobeigenschaften von einem Job im IBM Workload Scheduler-Agenten für z/OS in einen anderen Job in derselben Jobstrominstanz exportieren.

Sie können einige Jobeigenschaften von einem Job im IBM Workload Scheduler-Agenten für z/OS in einen anderen Job in derselben Jobstrominstanz exportieren. Wollen Sie eine Jobeigenschaft innerhalb der Definition eines anderen Jobs hinzufügen, die bei der Ausführung lokal auf dem Agenten aufgelöst wird, verwenden Sie die folgende Syntax:

```
${job:<jobname>.<eigenschaftsname>}
```

Dabei ist *<jobname>* der Namenswert oder Aliasnamenswert des Jobs, aus dem die Eigenschaftswerte exportiert werden sollen, und *<eigenschaftsname>* ist die Eigenschaft, auf die Sie sich beziehen. Bei dem Wert für *<eigenschaftsname>* muss die Groß-/Kleinschreibung nicht beachtet werden.

In Tabelle 13 auf Seite 72 wird die Liste der Eigenschaften gezeigt, die Sie von einem dynamischen Job im IBM Workload Scheduler-Agenten für z/OS an einen anderen Job übergeben können, und es wird die Zuordnung zwischen den Eigen-

schaften Extra information (Zusätzliche Informationen) des Jobs und den Eigenschaften angeben, die Sie verwenden können.

Tabelle 13. Eigenschaften für dynamische Jobs im IBM Workload Scheduler-Agenten für z/OS

Eigenschaften für dynamische Jobs im IBM Workload Scheduler-Agenten für z/OS, die in der Definition eines anderen Jobs übergeben werden können	Dynamischer Job im IBM Workload Scheduler-Agenten für z/OS, Eigenschaften Extra information (Zusätzliche Informationen)
<code>\${job:<jobname>.zAgentJESId}</code>	JES-ID
<code>\${job:<jobname>.zAgentJobName}</code>	Jobname
<code>\${job:<jobname>.zAgentStartReaderTime}</code>	Startzeit des Eingabeprogramms

Beispiel

In dem folgenden Beispiel wird gezeigt, wie Variablen von einem im IBM Workload Scheduler-Agenten für z/OS ausgeführten Job an einen anderen ausführbaren Job übergeben werden können, der in derselben Jobstrominstanz ausgeführt wird. Der Jobstrom WIN92MAS#JS_PROP enthält die Jobs ZSPD_ZOS1274#DDRIVEN_JOB (mit dem Aliasnamen JOBA) und NC112016#JOB. Der ausführbare Job NC112016#JOB referenzier die folgenden Eigenschaften von JOBA, der im IBM Workload Scheduler-Agenten für z/OS definiert ist:

- zAgentJESId
- zAgentJobName
- JOBA.zAgentStartReaderTime

Datenbankdefinitionen:

```
SCHEDULE WIN92MAS#JS_PROP
:
ZSPD_ZOS1274#DDRIVEN_JOB AS JOBA
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/
jsdl" xmlns:
jsdljcl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdljcl" name="JCL">
  <jsdl:application name="jcl">
    <jsdljcl:jcl>
      <jsdljcl:JCLParameters>
        <jsdljcl:jcl>
          <jsdljcl:byRefOrByDef>
            <jsdljcl:byDefinition>
              <jsdljcl:jclDefinition>//JOB JOB
//S1 EXEC PGM=IEFBR14</jsdljcl:jclDefinition>
            </jsdljcl:byDefinition>
          </jsdljcl:byRefOrByDef>
        </jsdljcl:jcl>
      </jsdljcl:JCLParameters>
      <jsdljcl:JOBParameters>
        <jsdljcl:jobStreamName>${tws.jobstream.name}</jsdljcl:jobStreamName>
        <jsdljcl:inputArrival>${tws.job.ia}</jsdljcl:inputArrival>
      </jsdljcl:JOBParameters>
    </jsdljcl:jcl>
  </jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Added by composer for job stream: WIN92MAS#JS_PROP."
RECOVERY STOP
```

NC112016#JOB

```

TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:XMLSchema="http://www.w3.org/2001/
XMLSchema" xmlns:
jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdl" xmlns:
jsdle="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdle" XMLSchema:
text="resolveVariableTable" name="executable">
  <jsd1:application name="executable">
    <jsdle:executable interactive="false">
      <jsdle:script>
echo JES Id = ${job:JOBA.zAgentJESId}
echo Job Name = ${job:JOBA.zAgentJobName}
echo Start Reader Time = ${job:JOBA.zAgentStartReaderTime}
</jdsle:script>
      </jdsle:executable>
    </jds1:application>
  </jds1:jobDefinition>
DESCRIPTION "Added by composer for job stream: WIN92MAS#JS_PROP."
RECOVERY STOP
FOLLOWS JOBA

```

Jobstandardausgabe aus einem Job an anderen Job in derselben Jobstrominstanz übergeben:

Sie können die Jobstandardausgabe aus einem dynamischen Job an einen anderen Job in derselben Jobstrominstanz exportieren.

Wollen Sie eine Jobstandardausgabe innerhalb der Definition eines anderen Jobs hinzufügen, die bei der Ausführung lokal auf dem Agenten aufgelöst wird, verwenden Sie die folgende Syntax:

```
${job:<jobname>.stdlist}
```

Dabei ist *<jobname>* der Namenswert oder Aliasnamenswert des Jobs, aus dem die Jobstandardausgabe exportiert werden soll.

Beispiel

In dem folgenden Beispiel wird gezeigt, wie Variablen von einem im IBM Workload Scheduler-Agenten für z/OS ausgeführten Job an einen anderen ausführbaren Job übergeben werden können, der in derselben Jobstrominstanz ausgeführt wird. Der Jobstrom WIN92MAS#JS_PROP enthält die Jobs ZSPD_ZOS1274#DDRIVEN_JOB (mit dem Aliasnamen JOBA) und NC112019#JOB0D. Der ausführbare Job NC112019#JOB0D referenziert die Standardausgabe von JOBA. Datenbankdefinitionen:

```

SCHEDULE WIN92MAS#JS_PROP
:
ZSPD_ZOS1274#DDRIVEN_JOB AS JOBA
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsd1:jobDefinition xmlns:jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/
jsdl" xmlns:
jsdljcl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdljcl" name="JCL">
  <jsd1:application name="jcl">
    <jsd1jcl:jcl>
      <jsd1jcl:JCLParameters>
        <jsd1jcl:jcl>
          <jsd1jcl:byRefOrByDef>
            <jsd1jcl:byDefinition>
              <jsd1jcl:jclDefinition>//JOB JOB
//S1 EXEC PGM=IEFBR14</jds1jcl:jclDefinition>
            </jds1jcl:byDefinition>
          </jds1jcl:byRefOrByDef>
        </jds1jcl:jcl>
      </jds1jcl:JCLParameters>

```

```

    <jsdljcl:JOBParameters>
      <jsdljcl:jobStreamName>${tw.s.jobstream.name}</jsdljcl:jobStreamName>
      <jsdljcl:inputArrival>${tw.s.job.ia}</jsdljcl:inputArrival>
    </jsdljcl:JOBParameters>
  </jsdljcl:jcl>
</jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Added by composer for job stream: WIN92MAS#JS_PROP."
RECOVERY STOP

```

NC112019#JOB

```

TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/
jsdl" xmlns:
jsdle="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdle" name="executable">
  <jsdl:application name="executable">
    <jsdle:executable interactive="false">
      <jsdle:script>echo &quot;stdlist: ${job:JOBA.stdlist}&quot;</jsdle:script>
    </jsdle:executable>
  </jsdl:application>
</jsdl:jobDefinition>
DESCRIPTION "Added by composer for job stream: WIN92MAS#JS_PROP."
RECOVERY STOP
FOLLOWS JOBA
END

```

Jobstandardausgabe aus einem Job als Standardeingabe an anderen Job in derselben Jobstrominstanz übergeben:

Sie können die Jobstandardausgabe aus einem dynamischen Job als Standardeingabe in einen anderen Job in derselben Jobstrominstanz exportieren.

Wollen Sie eine Jobstandardausgabe innerhalb der Definition eines anderen Jobs hinzufügen, die bei der Ausführung lokal auf dem Agenten aufgelöst wird, verwenden Sie die folgende Syntax:

```

${job:<jobname>.stduri}

```

Dabei ist *<jobname>* der Namenswert oder Aliasnamenswert des Jobs, aus dem die Jobstandardausgabe exportiert werden soll.

Beispiel

In dem folgenden Beispiel wird gezeigt, wie Variablen von einem im IBM Workload Scheduler-Agenten für z/OS ausgeführten Job an einen anderen ausführbaren Job übergeben werden können, der in derselben Jobstrominstanz ausgeführt wird. Der Jobstrom WIN92MAS#JS_PROP enthält die Jobs ZSPD_ZOS1274#DDRIVEN_JOB (mit dem Aliasnamen JOBA) und NC112019#JOB. Der ausführbare Job NC112019#JOB referenziert die Standardausgabe von JOBA. Datenbankdefinitionen:

```

SCHEDULE WIN92MAS#JS_PROP
:
ZSPD_ZOS1274#DDRIVEN_JOB AS JOBA
TASK
  <?xml version="1.0" encoding="UTF-8"?>
<jsdl:jobDefinition xmlns:jsdl="http://www.abc.com/xmlns/prod/scheduling/1.0/
jsdl" xmlns:
jsdljcl="http://www.abc.com/xmlns/prod/scheduling/1.0/jsdljcl" name="JCL">
  <jsdl:application name="jcl">
    <jsdljcl:jcl>
      <jsdljcl:JCLParameters>
        <jsdljcl:jcl>
          <jsdljcl:byRefOrByDef>
            <jsdljcl:byDefinition>

```

```

        <jSDLjcl:jclDefinition>//JOB JOB
//S1 EXEC PGM=IEFBR14</jSDLjcl:jclDefinition>
        </jSDLjcl:byDefinition>
        </jSDLjcl:byRefOrByDef>
        </jSDLjcl:jcl>
        </jSDLjcl:JCLParameters>
        <jSDLjcl:JOBParameters>
        <jSDLjcl:jobStreamName>${tws.jobstream.name}</jSDLjcl:jobStreamName>
        <jSDLjcl:inputArrival>${tws.job.ia}</jSDLjcl:inputArrival>
        </jSDLjcl:JOBParameters>
        </jSDLjcl:jcl>
        </jSDL:application>
</jSDL:jobDefinition>
DESCRIPTION "Added by composer for job stream: WIN92MAS#JS_PROP."
RECOVERY STOP

```

NC112019#JOB

```

TASK
        <?xml version="1.0" encoding="UTF-8"?>
<jSDL:jobDefinition xmlns:XMLSchema="http://www.w3.org/2001/XMLSchema" xmlns:
jSDL="http://www.abc.com/xmlns/prod/scheduling/1.0/jSDL" xmlns:
jSDLe="http://www.abc.com/xmlns/prod/scheduling/1.0/
jSDLe" XMLSchema:text="resolveVariableTable"
                                name="executable">
        <jSDL:application name="executable">
        <jSDLe:executable input="{job:JOBA.stduri}" interactive="false" path="cat"/>
        </jSDL:application>
</jSDL:jobDefinition>
DESCRIPTION "Added by composer for job stream: WIN92MAS#JS_PROP."
RECOVERY STOP
FOLLOWS JOBA
END

```

Variablen, die vom Agenten für z/OS aufgelöst werden

Diese Variablen werden vom Agenten für z/OS in der JCL aufgelöst, bevor die JCL zur Ausführung an JES weitergegeben wird.

Die folgenden Typen von Variablen werden unterstützt:

- Benutzerdefinierte Variablen
- Vordefinierte Variablen
- JCL-Anpassungsanweisungen

Agenten zum Ausführen der Variablensubstitution konfigurieren

Zur Ausführung dieses Typs von Variablensubstitution müssen Sie einige Schlüsselwörter der Anweisung TWSOPTS konfigurieren.

Führen Sie eine der folgenden Aktionen aus, um diesen Typ von Variablensubstitution ausführen zu können:

- Setzen Sie das Schlüsselwort VARSUB der Anweisung TWSOPTS auf den Wert YES. Dies bedeutet, dass die Variablensubstitution vom Anfang der JCL für alle Jobs stattfindet, die zur Ausführung auf dem Agenten für z/OS definiert sind.
- Setzen Sie das Schlüsselwort VARSUB der Anweisung TWSOPTS auf den Wert SCAN und geben Sie die Anweisung // *%OPC SCAN in Ihrem Job an. Die Substitution im Job beginnt an der Position, an der die Anweisung SCAN gefunden wird.

Wenn Sie Fehler durch die Variablensubstitution umgehen wollen, definieren Sie das Schlüsselwort VARFAIL. Wenn Sie die Variablensubstitution auch auf Inlineprozeduren anwenden wollen, verwenden Sie die Anweisung VARPROC. Details hierzu finden Sie in den Informationen zur Konfigurationsanweisung TWSOPTS.

Variablen in JCL codieren

Das Codieren von Variablen in JCL folgt bestimmten Regeln.

Unabhängig davon, ob es sich um benutzerdefinierte oder vom Produkt bereitgestellte Variablen handelt, können Variablennamen bis zu acht alphanumerische Zeichen lang sein, wobei das erste ein alphabetisches Zeichen sein muss. Variablenwerte können bis zu 44 alphanumerische Zeichen enthalten.

Bei der Verwendung einer Variablen in einem Job stellen Sie der Variablen ein Et-Zeichen (&), ein Prozentzeichen (%) oder ein Fragezeichen (?) voran. Das Symbol, das der Variablen vorausgeht, bestimmt, wie die Variable aufgelöst wird.

Et-Zeichen (&)

Diese Variablen werden von links nach rechts innerhalb der Zeile durch Werte ersetzt. Et-Zeichen-Variablen entsprechen den Standardvariablen in z/OS JCL-Prozeduren und verhalten sich so. Weitere Informationen finden Sie in der *JCL-Referenz*.

Wenn auf eine &-Variable unmittelbar eine %-Variable folgt (d. h. ohne Abschlusszeichen dazwischen), wird eine Verbundvariable gebildet. Siehe Verbundvariablen. Eine Verbundvariable kann auch gebildet werden, wenn eine &-Variable unmittelbar auf eine ?-Variable folgt.

Eine Zeichenfolge, die mit && beginnt, wird nicht substituiert. Dies liegt daran, dass das doppelte Et-Zeichen in JCL normalerweise zur Bezeichnung einer temporären Datei verwendet wird. Alle Zeichenfolgen dieser Art bleiben von der Variablensubstitution ausgenommen.

Prozentzeichen (%)

Diese Variablen können verwendet werden, um einfache Variablen und Verbundvariablen zu bilden.

Einfache Variablen

Wenn der Variablen ein Prozentzeichen (%) vorausgeht und sie durch einen Punkt oder durch ein anderes Abschlusszeichen als % abgeschlossen wird, wird der Variablen ein Wert zugeordnet und die Substitution wird für diese Variable durchgeführt.

Verbundvariablen

Bei Verwendung der JCL-Substitution können Sie *Verbundvariablen* bilden. Eine Verbundvariable setzt sich aus einer Verkettung der folgenden Elemente zusammen:

- Eine Variable (beliebigen Typs) gefolgt von einer Prozentzeichenvariablen ohne dazwischen gesetzte Punkte oder andere Abschlussymbole
- Eine Fragezeichenvariable gefolgt von einer Et-Zeichen-Variablen ohne dazwischen gesetzte Punkte oder andere Abschlussymbole

Die Werte der Prozentzeichenvariablen, die eine Verbundvariable bilden, werden nicht direkt substituiert. Stattdessen werden diese Werte zur Bildung neuer Variablen verwendet, denen eigene Werte zugeordnet werden. Diese Variablen werden in einer Reihe von Durchläufen aufgelöst. Die einzelnen Variablen, die die Verbundvariable bilden, werden von rechts nach links aufgelöst.

Betrachten Sie zum Beispiel die folgende JCL-Zeile aus einem Job:

```
//STEPLIB DD DSN=MY.&DATA%SET,DISP=OLD
```


Nehmen Sie an, dass für SET der Wert LIB angegeben wurde. Nach dem ersten Durchlauf wird aus der Variablen DATA%SET die Variable DATALIB, weil die am weitesten rechts stehende Prozentzeichenvariable im ersten Durchlauf aufgelöst wird. Dieser erste Durchlauf hat jetzt zu einer neuen Variablen DATALIB geführt, die der Agent im nächsten Durchlauf durch diese JCL-Zeile versucht aufzulösen.

Verbundvariablen können aus einer ganzen Reihe von %-Variablen zusammengesetzt werden. Betrachten Sie das folgende Beispiel:

```
//DDNAME1 DD DSN=MY.%VAR1%VAR2%VAR3....DATA,DISP=OLD
```

Nehmen Sie an, dass VAR3 den Wert SIX und VAR2SIX den Wert JUNE hat. Im ersten Durchlauf durch diese JCL-Zeile wird die Variable %VAR1%VAR2%VAR3....DATA zu %VAR1%VAR2SIX....DATA. Im zweiten Durchlauf wird die Variable %VAR1%VAR2SIX. zu %VAR1JUNE....DATA. Der Wert, der %VAR1JUNE. zugeordnet ist, bestimmt den endgültigen Wert, der substituiert wird.

Bei jeder Substitution wird ein Punkt getilgt, wenn der Wert der Variablen eingesetzt wird. Sie müssen die richtige Anzahl von Punkten angeben, um sicherzustellen, dass die Substitution ordnungsgemäß durchgeführt wird. Im vorigen Beispiel war ein zusätzlicher Punkt erforderlich, um den Anfang des Dateiqualifikationsmerkmals der zweiten Ebene zu bezeichnen.

Im nächsten Beispiel benötigen Sie nur eine Klammer, um die Verbundvariable zu vervollständigen. Dies liegt daran, dass die Klammer bei der Substitution nicht gelöscht wird.

```
//DDNAME1 DD DSN=MY.%VAR1%VAR2%VAR3(MEMBER),DISP=OLD
```

Fragezeichen (?)

Fragezeichenvariablen sind *positionsgebunden*; d. h., dass Sie angeben können, in welcher Spalte der Zeile der Variablenwert beginnen soll, wenn die Variable substituiert wird. Die Position, an der der Wert platziert wird, wird in dem Job angegeben, in dem die Variable verwendet wird. Beispiel:

```
?VAR1.
```

Diese Angabe bewirkt, dass der Wert von VAR1 in der Zeile und Spalte platziert wird, wo sich die Variable befindet.

```
?nnVAR1.
```

Diese Angabe bewirkt, dass der Wert von VAR1 in der Zeile, in der sich die Variable befindet, platziert wird und an der Spalte mit der Nummer beginnt, die durch *nn* angegeben wird.

In einer JCL-Zeile können mehrere ?-Variablen vorhanden sein. Die Positionen der Variablen selbst haben keinen Einfluss auf die Positionen der Variablenwerte. Diese Positionen werden durch die Spaltennummer bestimmt, die für die Variable angegeben wird. Beispiel:

```
//SYSIN DD *
.....1.....2.....3.....4.....5.....6.....7..
      ?20VAR1.?9VAR2.
```

Dabei hat VAR1 den Wert APRIL und VAR2 den Wert MAY. (Die Skalenzeile wurde nur zu Demonstrationszwecken eingefügt.) Das Ergebnis nach der Variablensubstitution sähe wie folgt aus:

```
//SYSIN DD *
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
          MAY          APRIL
```

Der Wert von ?-Variablen wird auf die gleiche Weise berechnet wie bei &- und %-Variablen und in der gleichen Reihenfolge. Jedoch werden ?-Variablen erst substituiert, nachdem alle Prozentzeichen- und Et-Zeichen-Variablen substituiert wurden. Dies liegt daran, dass der Wert der ?-Variablen nur in Bereichen einer Zeile platziert werden kann, die leer sind. Der Agent kann erst bestimmen, welche Bereiche einer Zeile leer sind, wenn die Substitution der Et-Zeichen- und Prozentzeichenvariablen stattgefunden hat.

Tabellenvariablen können sich nicht überlappen. Das heißt, die Werte von zwei verschiedenen Variablen können nicht so definiert werden, dass sie denselben Bereich einer Zeile belegen. Der Platz, den die Variablen ursprünglich selbst belegen, wird bei der Substitution ignoriert. Beispiel:

```
//SYSIN DD *
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
          ?20VAR1.?21VAR2.
/*
```

Dabei hat VAR1 den Wert APRIL und VAR2 den Wert MAY. Die Substitution wäre ungültig, weil beide Variablen versuchen, die Spalten 21, 22 und 23 zu verwenden.

Der Agent ändert den Bereich, der von der Variablen belegt wird, in Leerzeichen, wenn er nicht durch den substituierten Wert belegt wird. Beispiel:

```
//SYSIN DD *
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
THIS IS?40VAR1. THE STANDARD DATA.          IS A WET MONTH.
```

VAR1 hat den Wert APRIL. Nach der Substitution sieht die Zeile wie folgt aus:

```
//SYSIN DD *
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
THIS IS          THE STANDARD DATA.          APRIL IS A WET MONTH.
```

Der Agent hat den von der Variablen ursprünglich belegten Bereich in Leerzeichen geändert. Die anderen Daten in der Zeile werden nicht verschoben.

Anmerkung: Vordefinierte Variablen haben keine implizierte Position. Wenn diese Variablen als Tabellenvariablen angegeben sind, müssen Sie die Spaltennummer mit angeben. Beispiel: ?0ADID wird nicht akzeptiert, während ?200ADID gültig ist: Die Anwendungs-ID wird ab Spalte 20 eingesetzt.

Sie können Variablen wiederholt innerhalb des Jobs mit unterschiedlichen Präfixsymbolen verwenden.

Einer Et-Zeichen- oder Prozentzeichenvariablen kann ein Wert zugewiesen werden, der selbst eine Variable ist.

Ein Punkt bezeichnet das Ende eines Variablennamens. Zur Wahrung der Kompatibilität mit der Variablensubstitution in z/OS JCL-Prozeduren geht der Agent für

z/OS davon aus, dass ein Variablenname zu Ende ist (auch wenn der abschließende Punkt fehlt), wenn der Variablen eines der in der folgenden Tabelle aufgeführten Symbole folgt.

Tabelle 14. Symbole, die das Ende von Variablen markieren

Symbol	Beschreibung
,	Komma
/	Schrägstrich
'	Einfaches Anführungszeichen
(Linke Klammer
)	Rechte Klammer
*	Stern
+	Pluszeichen
-	Gedankenstrich
=	Gleichheitszeichen
	Leerzeichen ()

Beispiel: LIBRARY hat für die folgende Anweisung den Wert LINKLIB:

```
//STEPLIB DD DSN=MY.&LIBRARY.(HDEAQ03),DISP=SHR
```

Oder für die folgende Anweisung (ohne abschließenden Punkt):

```
//STEPLIB DD DSN=MY.&LIBRARY(HDEAQ03),DISP=SHR
```

In diesem Fall sieht die JCL-Zeile nach der Substitution wie folgt aus:

```
//STEPLIB DD DSN=MY.LINKLIB(HDEAQ03),DISP=SHR
```

Das Produkt geht davon aus, dass die Variable LIBRARY endet, wenn es die linke Klammer '(' erkennt.

Der abschließende Punkt wird bei der Substitution einer Variablen gelöscht. Andere Abschlussymbole verbleiben an ihrer Position.

Benutzerdefinierte Variablen

Sie können Ihre eigenen definierten Variablen in den JCLs hinzufügen.

Die Namen benutzerdefinierter Variablen können bis zu acht alphanumerische Zeichen enthalten, wobei das erste Zeichen alphabetisch sein muss. Ein längerer Name wird als nicht gültig betrachtet und die Variable wird nicht verarbeitet. In diesem Fall wird ein Fehler in EELMLOG aufgezeichnet.

Variablenwerte können bis zu 44 alphanumerische Zeichen enthalten. Längere Werte werden auf 44 Zeichen abgeschnitten.

Zum Definieren dieser Variablen verwenden Sie die Variablentabellendefinition in Dynamic Workload Console. Gehen Sie wie folgt vor, um eine Variablentabelle zu definieren und Variablen hinzuzufügen:

1. Klicken Sie in der Navigationsleiste auf **Verwaltung>Auslastungsentwurf>Auslastungsdefinitionen verwalten**.
2. Wählen Sie die IBM Workload Scheduler-Engine aus, wenn Sie dazu aufgefordert werden, und geben Sie eine gültige Benutzer-ID und ein Kennwort an, wenn dies erforderlich ist.

3. Wählen Sie **Neu** oder **Suchen** und anschließend **Variablen-tabelle** aus.

Alternativ können Sie den Befehl **composer variable** verwenden.

Variablen-tabellen können auf der Ebene von Laufzyklen, Jobströmen und Workstations zugewiesen werden.

Die Variablen (und die Werte), die in einer JCL enthalten sind, werden bei der Übergabe in einer einzigen Tabelle an den Agenten für z/OS weitergegeben. Wenn der Scheduler feststellt, dass derselbe Variablenname in mehr als einer der Tabellen vorhanden ist, die der JCL zugeordnet sind, wählt er den ersten Wert in der folgenden Reihenfolge aus:

1. Laufzyklus
2. Jobstrom
3. Workstation

Anmerkung: Der Agent verarbeitet keine Variablen, die in der globalen Variablen-tabelle (Standardvariablen-tabelle) definiert sind.

Vordefinierte Variablen

Mit IBM Workload Scheduler werden vordefinierte Variablen für die Verwendung mit dem Agenten für z/OS bereitgestellt.

Der Agent versucht nie, Variablen-Definitionen für diese Variablen aus einer Variablen-tabelle zu lesen. Die folgenden Typen von Variablen sind verfügbar:

- Jobstrombezogene Variablen
- Jobbezogene Variablen
- Datumsbezogene Variablen
- Variablen mit dynamischem Format
- Temporäre Variablen

Jobstrombezogene Variablen:

Diese Variablen beziehen sich auf Informationen zur Jobstrominstanz.

In der folgenden Tabelle sind vordefinierte jobstrombezogene Variablen aufgeführt:

Tabelle 15. Vordefinierte jobstrombezogene Variablen

Variablenname	Länge (in Byte)	Beschreibung
OADID	16	Jobstromname.
ODAY	1	Wochentag des Eingangs der Jobstrominstanz (1-7); mit 1 = Montag, 7 = Sonntag.
ODD	2	Tag des Monats des Eingangs der Jobstrominstanz im DD-Format.
ODDD	3	Tag des Jahres des Eingangs der Jobstrominstanz im DDD-Format.
ODMY1	6	Eingangsdatum der Jobstrominstanz im DDMMYY-Format.
ODMY2	8	Eingangsdatum der Jobstrominstanz im DD/MM/YY-Format.
OHH	2	Stunde des Eingangs der Jobstrominstanz im HH-Format.

Tabelle 15. Vordefinierte jobstrombezogene Variablen (Forts.)

Variablenname	Länge (in Byte)	Beschreibung
OHHMM	4	Stunde und Minute des Eingangs der Jobstrominstanz im HHMM-Format.
OMM	2	Monat des Eingangs der Jobstrominstanz im MM-Format.
OMMY	4	Monat und Jahr des Eingangs der Jobstrominstanz im MMY-Format.
OWW	2	Woche des Jahres des Eingangs der Jobstrominstanz im WW-Format.
OWWD	3	Woche und Tag in der Woche des Eingangs der Jobstrominstanz im WWD-Format. Dabei ist WW die Nummer der Woche im Jahr und D der Tag in der Woche.
OWWLAST	1	Ein Wert (Y - yes oder N - no), der angibt, ob das Eingangsdatum der Jobstrominstanz in der letzten Woche des Monats liegt.
OWWMONTH	1	Ein Wert zwischen 1 und 6, der die Woche im Monat des Eingangs der Jobstrominstanz angibt, wobei jede neue Woche an einem Montag beginnt. Betrachten Sie zum Beispiel die folgenden Laufeingangsdatumsangaben für den Monat März im Jahr 1997: Datum OWWMONTH Samstag, der 1. 1 Montag, der 3. 2 Montag, der 31. 6
OYMD	8	Eingangsdatum der Jobstrominstanz im YYYYMMDD-Format.
OYM	6	Monat im Jahr des Eingangs der Jobstrominstanz im YYYYMM-Format.
OYMD1	6	Eingangsdatum der Jobstrominstanz im YMMDD-Format.
OYMD2	8	Eingangsdatum der Jobstrominstanz im YY/MM/DD-Format.
OYMD3	10	Eingangsdatum der Jobstrominstanz im YYYY/MM/DD-Format.
OYY	2	Jahr des Eingangs der Jobstrominstanz im YY-Format.
OYYDDD	5	Eingangsdatum der Jobstrominstanz als Julianisches Datum im YYDDD-Format.
OYYMM	4	Monat im Jahr des Eingangs der Jobstrominstanz im YYMM-Format.
OYYYY	4	Jahr des Eingangs der Jobstrominstanz im YYYY-Format. Beispiel: 1997.

Jobbezogene Variablen:

Diese Variablen beziehen sich auf Informationen zur Jobstanz.

In der folgenden Tabelle sind vordefinierte jobbezogene Variablen aufgeführt:

Tabelle 16. Vordefinierte jobbezogene Variablen

Variablenname	Länge (in Byte)	Beschreibung
OSSID	16	Subsystemname des Agenten für z/OS.

Datumsbezogene Variablen:

Diese Variablen beziehen sich auf das aktuelle Datum und die aktuelle Uhrzeit. Das heißt, die Uhrzeit und das Datum des Zeitpunkts, zu dem der Job übergeben wurde.

In der folgenden Tabelle sind vordefinierte datumsbezogene Variablen aufgeführt:

Tabelle 17. Vordefinierte datumsbezogene Variablen

Variablenname	Länge (in Byte)	Beschreibung
CDAY	1	Aktueller Tag der Woche; 1 = Montag, 7 = Sonntag.
CDD	2	Aktueller Tag des Monats im DD-Format.
CDDD	3	Nummer des Tags im aktuellen Jahr.
CDDMMYY	6	Aktuelles Datum im DDMMYY-Format.
CHH	2	Aktuelle Uhrzeit im HH-Format.
CHHMM	4	Aktuelle Stunde und Minute im HHMM-Format.
CHHMSS	6	Aktuelle Stunde, Minute und Sekunde im HHMMSS-Format.
CHHMSSX	8	Aktuelle Stunde, Minute, Sekunde und Hundertstelsekunden im HHMMSSXX-Format.
CMM	2	Aktueller Monat im MM-Format.
CMMYY	4	Aktueller Monat im Jahr im MMY-Format.
CWW	2	Nummer der Woche im aktuellen Jahr.
CWWD	3	Aktueller Tag in der Woche im WWD-Format. Dabei ist WW die Nummer der Woche im Jahr und D der Tag in der Woche.
CYMD	8	Aktuelles Datum im YYYYMMDD-Format.
CYY	2	Aktuelles Jahr im YY-Format.
CYYDDD	5	Aktuelles Julianisches Datum im YYDDD-Format.
CYYMM	4	Aktueller Monat im Jahr im YYMM-Format.
CYYMMDD	6	Aktuelles Datum im YYMMDD-Format.
CYYYY	4	Aktuelles Jahr im YYYY-Format. Beispiel: 1997.
CYYYYMM	6	Aktueller Monat im Jahr im YYYYMM-Format.

Variablen mit dynamischem Format:

Vordefinierte Variablen mit dynamischem Format sind zeit- und datumsbezogen.

Sie definieren das für diese Variablen erforderliche Format mit der Anweisung SETFORM. Wenn Sie zum Beispiel das Eingangsdatum des Laufs durch das Format MM:DD:YY substituieren wollen, definieren Sie die dynamische Variable OC-DATE wie folgt:

```
//*%OPC SETFORM ODATE=(MM:DD:YY)
```

Wenn Sie das Format einer Variablen mit dynamischem Format mit der Anweisung SETFORM definiert haben, können Sie später im Job ein anderes Format verwenden, indem Sie dieselbe Variable mit einer anderen Anweisung SETFORM umdefinieren.

In der folgenden Tabelle sind diese Variablen aufgeführt.

Tabelle 18. Vordefinierte Variablen mit dynamischem Format

Variablenname	Beschreibung
CDATE	Aktuelles Datum.
CTIME	Aktuelle Uhrzeit.
ODATE	Eingangsdatum der Jobstrominstanz.
OCTIME	Eingangszeit der Jobstrominstanz (Stunden und Minuten).
OPIADATE	Eingangsdatum der Jobinstanz (wenn leer, übernimmt diese Variable den Wert des Eingangsdatums der Jobstrominstanz).
OPIATIME	Eingangszeit der Jobinstanz (wenn leer, übernimmt diese Variable den Wert der Eingangszeit der Jobstrominstanz).

Temporäre Variablen:

Sie können temporäre Variablen mithilfe der Anweisung SETVAR erstellen.

Sie können temporäre Variablen mit einem der folgenden Elemente erstellen:

- Arithmetischer Ausdruck für die datums- oder zeitbezogenen Variablen
- Teilzeichenfolge einer anderen Variablen
- Ergebnis einer arithmetischen Addition oder Subtraktion
- Verkettete Zeichenfolgen oder Variablen, die auf einen alphanumerischen Wert gesetzt sind

Beispiel: Sie möchten auf den ersten Arbeitstag der nächsten Woche nach dem Eingangsdatum des Jobstroms im Format YY/MM/DD Bezug nehmen. Sie können dies tun, indem Sie mithilfe der Anweisung SETVAR eine temporäre Variable aus der bereitgestellten Variable OYMD2 erstellen. Der temporären Variablen wird der Wert (Datum) des ersten Arbeitstags nach dem Eingangsdatum des Jobstroms wie folgt zugewiesen:

Beispiel

```
//*%OPC SCAN  
//*%OPC SETVAR TVAR=(OYMD2+1WK)
```

Wenn das Eingangsdatum des Jobstroms 12/07/20 ist und der erste Arbeitstag der folgenden Woche Montag 12/07/27, wird der Variablen TVAR der Wert 12/07/27 zugewiesen. Sie können jetzt TVAR wie eine normale Variable in den übrigen Teilen des Jobs verwenden. Sie können der Variablen später im Job auch einen neuen Wert geben.

Details finden Sie in „Anweisung SETVAR“ auf Seite 87.

JCL-Anpassungsanweisungen

Der Agent für z/OS arbeitet mit besonderen Kommentaranweisungen, die als *Anweisungen* ('Directives') bezeichnet werden, um den Ein- und Ausschluss von Zeilen zu verwalten und Aspekte der Variablensubstitution zu steuern.

Dabei handelt es sich um die folgenden Anweisungen:

- SCAN
- SETFORM
- SETVAR
- BEGIN und END
- FETCH

Für die Anweisungen gilt die folgende allgemeine Syntax:

- Jede Anweisung muss in einer neuen 80-Byte-Zeile beginnen.
- Alle Anweisungen beginnen mit `//*%0PC` in den Spalten 1 bis 7, gefolgt von mindestens einem Leerzeichen.
- Anweisungsparameter können in beliebiger Reihenfolge codiert werden.
- Anweisungsparameter können mehrmals in derselben Anweisung vorkommen.
- Anweisungsparameter werden durch Kommas ohne eingebettete Leerzeichen in derselben Zeile getrennt.
- Wenn mehr als ein Parameterwert angegeben wird, sind Klammern erforderlich.
- Eine Anweisungsspezifikation darf nicht länger als 71 Zeichen sein. Sie kann in einer neuen Zeile fortgesetzt werden, wenn die Anweisung durch ein Komma nach einem vollständig oder teilweise angegebenen Parameter geteilt wird.
- Die Positionen 72 bis 80 werden ignoriert.
- Jede Fortsetzungszeile muss mit `//*%0PC` in den Spalten 1 bis 7, gefolgt von mindestens einem Leerzeichen, beginnen.
- Nach der erfolgreichen Ausführung einer Anweisung wird die Zeichenfolge `//*%0PC` in `//*>0PC` geändert.

Wenn eine Zeile mit `//*%0PC` beginnt und keine der bekannten Anweisungen gefunden wird, behandelt die Jobsstitutionsroutine des Agenten alle anderen Anweisungen, die sie findet, als "unbekannt" und führt keine Aktion aus.

Anmerkung: Wenn Sie `VARSUB(SCAN)` in der Anweisung `TWSOPTS` festlegen, muss die Anweisung `SCAN` in der JCL vorhanden sein, damit alle nachfolgenden Anweisungen verarbeitet werden.

Anweisung SCAN:

Zweck

Wenn das Schlüsselwort `VARSUB` der Anweisung `TWSOPTS` auf den Wert `SCAN` gesetzt wird, informiert diese Anweisung, wenn sie in der JCL gefunden wird, den Agenten für z/OS, dass die Variablensubstitution bei dieser Zeile beginnen soll. Dies gilt auch für die Verarbeitung der Anweisungen, das heißt, dass die Verarbeitung der Anweisungen bei der Zeile beginnt, in der `SCAN` angegeben ist.

Syntax

►► `//*%0PC SCAN` ◀◀

Hinweise

Die Anweisung SCAN wird nur berücksichtigt, wenn der Parameter VARSUB der Anweisung TWSOPTS auf den Wert SCAN gesetzt ist.

Ausgehend von der Annahme, dass VARSUB(SCAN) angegeben ist, wird im folgenden Beispiel die Variable MODULE nicht substituiert, weil sie sich vor der Anweisung SCAN befindet. Die Variable LIBRARY, die nach der Anweisung SCAN angeordnet ist, wird substituiert.

Beispiel

```
//OPSTATUS JOB (ACCOUNT),'Set completed',CLASS=A
//STEP1 EXEC PGM=&MODULE.
//%OPC SCAN
//STEPLIB DD DSN=TWS.LOAD.&LIBRARY.,DISP=SHR
//EELMLIB DD DSN=TWS.MESSAGE.LIBRARY,DISP=SHR
//EELMLOG DD SYSOUT=A
//SYSIN DD *
/*
```

Anweisung SETFORM:

Zweck

Diese Anweisung definiert das Format bereitgestellter Variablen mit dynamischem Format. Nach der Verarbeitung der Anweisung SETFORM durch den Agenten können Sie die Variable verwenden und arithmetische Berechnungen mit der Variablen durchführen. Sie können die Variable innerhalb des Jobs mehrmals umdefinieren, wenn dies erforderlich ist.

Syntax

```
►►—//%OPC SETFORM—name_der_dynamischen_variablen—=(—format—)—————►
```

Parameter

name_der_dynamischen_variablen=(*formatausdruck*)

Die dynamische Variable verwendet das im Formatausdruck definierte Format.

Hinweise

Der Name der dynamischen Variablen muss ein Name der vordefinierten dynamischen Variablen sein (siehe „Variablen mit dynamischem Format“ auf Seite 82).

Der Formatausdruck kann eine Kombination aus zeitbezogenen Schlüsselwörtern, datumsbezogenen Schlüsselwörtern und Begrenzern sein.

Die folgenden datumsbezogenen Schlüsselwörter sind verfügbar:

CC Stellt das Jahrhundert dar. Dies wird in Kombination mit YY verwendet, um das Format einer vollen Jahresangabe wie 2012 zu definieren.

YY Stellt die beiden letzten Ziffern im Jahr dar.

MM Stellt den Monat dar.

DDD Stellt den Tag im Jahr dar. Dieses Schlüsselwort wird vor dem Schlüssel-

wort DD substituiert: Die Zeichenfolge DDDDDD wird als zwei Schlüsselwörter DDD und nicht als drei Schlüsselwörter DD interpretiert.

DD Stellt den Tag im Monat dar.

Die folgenden zeitbezogenen Schlüsselwörter sind verfügbar:

HH Stellt die Stunde dar.

MM Stellt die Minuten dar.

Alle anderen Zeichen im Formatausdruck werden als Begrenzer interpretiert. Solche Begrenzer können alphabetische und numerische Zeichen sowie beliebige Symbole mit Ausnahme der Variablensubstitutionszeichen &, %, ?, = und den Klammern () sein.

Für die zeitbezogenen dynamischen Variablen OCTIME, OPIATIME und CTIME werden nur HH und MM erkannt. YY wird zum Beispiel nicht substituiert. MM wird durch den Minutenteil der Uhrzeit substituiert.

Für datumsbezogene dynamische Variablen werden nur CCYY, YY, MM, DD und DDD erkannt. CC ohne YY wird nicht erkannt. HH wird nicht substituiert. MM wird durch den Monatsteil des Datums substituiert.

Sie können mehr als einen Begrenzer zwischen Schlüsselwörtern verwenden.

Beispiel: MM//DD-- YY ist ein gültiger Formatausdruck.

Begrenzer sind optional. Das heißt, Sie können aufeinander folgende Schlüsselwörter ohne Begrenzer wie DDMMYY definieren.

Nehmen Sie für die folgenden Beispiele an, dass die Eingangszeit des Laufs der 31. Dezember 2012, 16:10 Uhr ist.

Beispiel

```
//*%OPC SCAN
//*%OPC SETFORM ODATE=(YY/MM/DDD)
```

Die resultierende Variable &ODATE wäre: 12/12/365

Für die Beispiele in Tabelle 19 wird dasselbe Eingangsdatum für den Lauf verwendet.

Tabelle 19. Substitutionsergebnisse unter Verwendung des dynamischen Formats

Variable mit dynamischem Format	Formatausdruck	Ergebnis
OCDATE	YY-MM-DDABC	12-12-31ABC
OCTIME	HH MM	16 10 (Beachten Sie, dass MM durch die Minuten bei Zeitvariablen und durch den Monat bei Datumsvariablen substituiert wird.)
OCDATE	DDDDD	36531. DDD ist der 365. Tag des Jahres und DD ist der Tag des Monats.
OCDATE	DDDD	365D. DDD ist der 365. Tag des Jahres. Für das letzte D wurde keine Entsprechung gefunden.

Tabelle 19. Substitutionsergebnisse unter Verwendung des dynamischen Formats (Forts.)

Variable mit dynamischem Format	Formatausdruck	Ergebnis
OCDATE	YYMMHHMMSS	1212HH12SS. Dies ist eine Datumsvariable, sodass HH nicht substituiert wird.

Anweisung SETVAR:

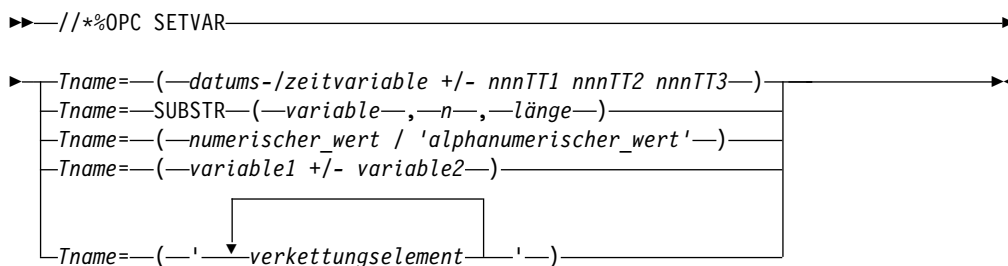
Zweck

Diese Anweisung erstellt eine temporäre Variable unter Verwendung eines der folgenden Elemente:

- Arithmetischer Ausdruck zusammen mit angegebenen Datums- oder Zeitvariablen
- Teilzeichenfolge einer anderen Variablen
- Ergebnis einer arithmetischen Addition oder Subtraktion
- Verkettete Zeichenfolgen oder Variablen, die auf einen alphanumerischen Wert gesetzt sind

Wenn der Agent für z/OS die Anweisung SETVAR verarbeitet hat, können Sie die temporäre Variable auf dieselbe Weise wie andere Variablen verwenden. Sie können eine temporäre Variable später im Job neu definieren.

Syntax



Parameter

Tname=(*datums-/zeitvariable +/- nnnTT1 nnnTT2 nnnTT3*)

Tname

Der Name der temporären Variablen, der mit dem Buchstaben T beginnt.

datums-/zeitvariable

Eine Angabe in einem der folgenden definierten Formate:

Datumsformate:

ODMY1, ODMY2, OYMD, OYMD1, OYMD2, OYMD3, CDDMMYY, CYMD, CYYMMDD, OCDATE, CDATE

Formate für Tag im Jahr:

ODDD, OYYDDD, OLYYDDD, CDDD, CYYDDD

Formate für Tag im Monat:

ODD, CDD

Formate für Tag in der Woche:

OWWD, CWWD

Formate für Wochentag:

ODAY, CDAY

Formate für Woche:

OWW, CWW

Formate für Monat:

OMM, OMMYY, OYM, OYYMM, CMM, CMMYY,
CYYMM, CYYYYMM

Formate für Jahr:

OYY, OYYYY, CYY, CYYYY

Formate für Zeit:

OHHMM, CHHMM, CHHMMSSX, CTIME, OCTIME

Formate für Stunde:

OHH, OLHH, CHH, CYYY

nnn Eine Zahl aus dem Bereich von 0 bis 999.

TT1 Der erste mögliche Typ. Sie können die folgenden Werte angeben:

Für datumsbezogene Variablen

CD Kalendertage.

WK Wochen. Wochen werden vor der Durchführung der Berechnung in Tage konvertiert.

MO Monate. Die Durchführung einer Berechnung am Monatsteil betrifft nur den Monat und möglicherweise das Jahr. Die Berechnung generiert immer gültige Ergebnisse, wobei tatsächliche Monatslängen und Schaltjahre berücksichtigt werden. Beispiel: Das Addieren eines Monats zum Wert 080131 ergibt 080229 (unter Berücksichtigung des Schaltjahrs).

YR Jahre.

Für zeitbezogene Variablen

HH Stunden.

MM Minuten.

SS Sekunden.

TT2 Der zweite mögliche Typ, gültig nur für zeitbezogene Variablen. Sie können die folgenden Werte angeben:

MM Minuten.

SS Sekunden.

TT3 Der dritte mögliche Typ, gültig nur für zeitbezogene Variablen. Sie können nur den Wert **SS** (Sekunden) angeben.

Sie können das Format *nnnTT1 nnnTT2 nnnTT3* nur für zeitbezogene Variablen verwenden, um Stunden, Minuten und Sekunden zu einem gegebenen Zeitwert hinzuzuaddieren oder davon zu subtrahieren. Geben Sie dieses Tripelformat nur an, wenn Sie Stunden, Minuten und Sekunden angeben wollen.

Die Verwendung von doppelten Typen, wie zum Beispiel in 6HH, 5MM, 7MM, ist nicht zulässig.

variable

Der Name der Quellenzeichenfolgenvariablen. Verwenden Sie einen vorhandenen Variablennamen, der ordnungsgemäß definiert und zugänglich ist. Sie können eine beliebige bereitgestellte oder benutzerdefinierte Variable angeben. Die Länge der Variablen, die ersetzt wird, ist auf die verbleibende JCL-Zeilenzahl begrenzt, die nicht von der Anweisung verwendet wird.

n Eine ganze Zahl im Bereich von 1 bis 60. Sie definiert die Anfangsposition

in *variable* der Teilzeichenfolgevariablen. Wenn dieser Wert die Länge von *variable* überschreitet, wird die resultierende Teilzeichenfolge mit Leerzeichen aufgefüllt.

länge Eine ganze Zahl im Bereich von 1 bis 60. Sie definiert die Länge der Teilzeichenfolgevariablen. Wenn dieser Wert die Länge von *variable* überschreitet, wird die resultierende Teilzeichenfolge mit Leerzeichen aufgefüllt.

numerischer_wert

Eine ganze Zahl im Bereich von 0 bis 99999, die den Wert der temporären Variablen definiert.

alphanumerischer_wert

Eine Zeichenfolge aus alphanumerischen Zeichen, die den Wert der temporären Variablen definiert. Schließen Sie die Zeichenfolge in einfache Anführungszeichen ein. Die Zeichenfolge kann bis zu 48 Zeichen lang sein.

Wenn die Länge der gesamten Anweisung in der JCL die Spalte 72 erreicht, wird eine Fehlernachricht ausgegeben.

variable1

Eine ganze Zahl im Bereich von 0 bis 99999, die den ersten Operanden einer arithmetischen Addition oder Subtraktion definiert, deren Ergebnis den Wert der temporären Variablen definiert. Der arithmetische Ausdruck darf keine Leerzeichen enthalten.

variable2

Eine ganze Zahl im Bereich von 0 bis 99999, die den zweiten Operanden einer arithmetischen Addition oder Subtraktion definiert, deren Ergebnis den Wert der temporären Variablen definiert. Der arithmetische Ausdruck darf keine Leerzeichen enthalten.

verkettungselement

Ein Element der folgenden Art:

- Eine Variable, die zuvor auf einen alphanumerischen Wert gesetzt wurde
- Eine Zeichenfolge aus alphanumerischen Zeichen

Schließen Sie die Elementliste in einfache Anführungszeichen ein. Das Ergebnis darf 48 Zeichen nicht überschreiten.

Hinweise

Die folgenden Beispiele demonstrieren, wie temporäre Variablen verwendet werden, die mit der Anweisung SETVAR erstellt wurden:

Beispiel 1 (mit einem arithmetischen Ausdruck)

Wenn das Eingangsdatum des Laufs 12/12/26 ist, wird der Ausdruck wie folgt substituiert:

```
TVAR=(360+4)  
TVAR=364
```

Wenn das Eingangsdatum des Laufs 12/12/30 ist, wird der Ausdruck wie folgt substituiert:

```
TVAR=(364+4)  
TVAR=003
```

Beispiel 2 (mit einer Variablen dynamischen Formats)

```

//%OPC SCAN
//%OPC SETFORM CDATE=(ACCURATE DATE CCYY MM DD)
//%OPC SETVAR TDATE=(CDATE + 1CD)

```

Wenn das Eingangsdatum des Laufs 12/12/26 ist, werden die Ausdrücke wie folgt substituiert:

```

CDATE = 'ACCURATE DATE 2012 12 26'
TDATE = 'ACCURATE DATE 2012 12 27'

```

Wenn der Ausdruck bereitgestellte Variablen mit dynamischem Format enthält, die den ersten oder letzten Tag im Monat oder im Jahr der Eingangszeit der Jobstrominstanz beinhalten, muss das berechnete Datum innerhalb des Bereichs von vier Jahren vor und sieben Jahren nach dem aktuellen Jahr liegen. Wenn das aktuelle Jahr 2012 ist, muss das resultierende Datum der temporären Variablen nach dem 31.12.2007 (07/12/31) und vor dem 01.01.2020 (20/01/01) liegen.

Beispiel 3 (Verwendung von SUBSTR)

- Verwendung einer Variablen, die in einer JCL-Variablentabelle definiert ist:

```

//%OPC SETVAR TVAR=SUBSTR(&VAR1,2,4)

```

VAR1 ist eine Variable, die in einer JCL-Variablentabelle definiert ist.

TVAR ist eine Teilzeichenfolge des Werts von VAR1, die an Position 2 beginnt und 4 Zeichen lang ist.

- Verwendung einer vordefinierten Variablen:

```

//%OPC SETFORM OCDATE=(YYMMDD)
//%OPC SETVAR TVAR1=('&OCDATE')
//%OPC SETVAR TVAR2=SUBSTR(&TVAR1,3,2)
//%OPC SETVAR TVAR3=(OCDATE + 1MO)
//%OPC SETVAR TVAR4=SUBSTR(&TVAR3,3,2)

```

Wenn das Eingangsdatum des Laufs 08/06/16 ist, werden die Ausdrücke wie folgt substituiert:

```

TVAR1 = 080616
TVAR2 = 06
TVAR3 = 080716
TVAR4 = 07

```

Tatsächlich gibt der Parameter SUBSTR eine Teilzeichenfolge der Werte von TVAR1 und TVAR3 an, die an Position 3 beginnt und 2 Zeichen lang ist. Gemäß dem Format, das durch die Anweisung SETFORM festgelegt wird, gibt sie den MM-Teil des Datumswerts an.

Beispiel 4 (Arithmetik mit temporären Variablen)

```

//%OPC SETVAR TX=(1)
//%OPC SETVAR TY=(2)
//%OPC SETVAR TZ=(&TX+&TY)

```

TZ ist eine temporäre Variable, die auf das Ergebnis der arithmetischen Addition gesetzt wird.

Beispiel 5 (Verkettung temporärer Variablen)

```

//%OPC SETVAR T001=('STRING1')
//%OPC SETVAR T002=('STRING2')
//%OPC SETVAR T003=('&T001 &T002 CONCATENATED STRINGS')

```

T003 ist eine temporäre Variable, die auf den folgenden Wert gesetzt wird: STRING1 STRING2 CONCATENATED STRINGS

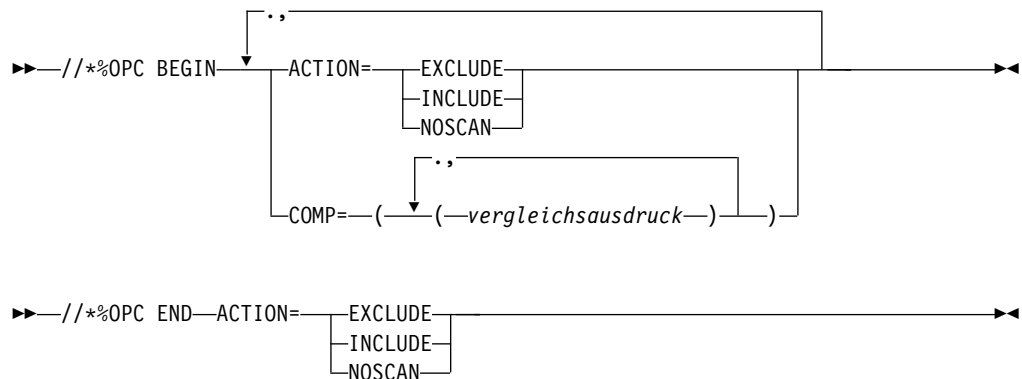
Anweisungen BEGIN und END:

Zweck

Diese Anweisungen geben als Paar abhängig vom Wert des Schlüsselworts ACTION Folgendes an:

- Den Anfang und das Ende der Variablensubstitutionsaktion, die vom Agenten durchgeführt wird
- Den Anfang und das Ende der Zeilen, die in den angepassten Job *einzuschließen* sind
- Den Anfang und das Ende der Zeilen, die aus dem angepassten Job *auszuschließen* sind

Syntax



Parameter

ACTION=(EXCLUDE | INCLUDE | NOSCAN)

Gibt an, welche BEGIN/END-Aktion auszuführen ist.

EXCLUDE

Dieser Wert gibt an, dass die Zeilen nach dieser Anweisung BEGIN bis zur nächsten Anweisung END ACTION=EXCLUDE aus dem Job ausgeschlossen werden sollen, der für diese Operation übergeben wird.

INCLUDE

Dieser Wert gibt an, dass die Zeilen nach dieser Anweisung BEGIN bis zur nächsten Anweisung END ACTION=INCLUDE in den Job eingeschlossen werden sollen, der für diese Operation übergeben wird.

NOSCAN

Dieser Wert gibt an, dass alle Variablen nach dieser Anweisung BEGIN bis zur nächsten Anweisung END ACTION=NOSCAN nicht substituiert werden sollen.

COMP=((vergleichsausdruck), (vergleichsausdruck),...)

Gibt Vergleichsausdrücke an, die zur Entscheidung verwendet werden, ob die Anweisung BEGIN berücksichtigt werden soll. Wenn der Vergleichsausdruck wahr ist, wird die Anweisung BEGIN berücksichtigt. Details zum Definieren von Vergleichsausdrücken finden Sie in „Schlüsselwort COMP in Anweisungen BEGIN und FETCH“ auf Seite 94.

Hinweise

In einem Job muss jede Anweisung BEGIN eine entsprechende Anweisung END mit derselben Angabe für ACTION haben. Beispiel: Betrachten Sie die folgende Anweisung:

```
//*%OPC BEGIN ACTION=EXCLUDE
```

Diese Anweisung erfordert die folgende entsprechende Anweisung END:

```
//*%OPC END ACTION=EXCLUDE
```

Wenn der Agent eine unpaarige Anweisung BEGIN oder END erkennt, wird die Verarbeitung mit einem Fehler beendet. Auch eine Anweisung BEGIN, die nicht berücksichtigt wird, weil der zugehörige Vergleichsausdruck kein wahres Ergebnis hat, erfordert eine entsprechende Anweisung END.

Nur die folgenden Anweisungen können sich im Geltungsbereich einer Anweisung BEGIN ACTION=NOSCAN und einer Anweisung END ACTION=NOSCAN befinden:

```
SETFORM  
SETVAR
```

Wenn sich diese Anweisungen im Bereich einer Anweisung NOSCAN befinden, werden sie immer berücksichtigt, selbst wenn eine Vergleichsbedingung gilt, die falsch ist.

Anweisungen BEGIN und END, die ACTION=INCLUDE oder ACTION=EXCLUDE angeben, können nicht verschachtelt werden und dürfen sich nicht überschneiden. Sie können jedoch einen verschachtelten NOSCAN-Bereich vollständig beinhalten.

Betrachten Sie die folgenden Beispiele:

Beispiel 1

```
//*%OPC SCAN  
//*%OPC BEGIN ACTION=INCLUDE  
//DDNAME1 DD DSN=&HIONE..&DATASET1,DISP=SHR  
//DDNAME2 DD DSN=&HIONE..&DATASET2,DISP=SHR  
//*%OPC END ACTION=INCLUDE
```

1
2
3
4

Beispiel 1 ist gültig. Die Zeilen **2** und **3** werden in den Job für die Operation eingeschlossen.

Beispiel 2

```
//*%OPC SCAN  
//*%OPC BEGIN ACTION=EXCLUDE  
//EXEC PGM=MYPROG  
//*%OPC BEGIN ACTION=INCLUDE  
//DDNAME1 DD DSN=&HIONE..&DATASET1,DISP=SHR  
//SYSOUT DD SYSOUT=A  
//*%OPC END ACTION=EXCLUDE  
//DDNAME2 DD DSN=&HIONE..&DATASET2,DISP=SHR  
//*%OPC END ACTION=INCLUDE
```

1
2
3
4
5
6
7
8
9

Beispiel 2 ist ungültig. Eine Aktion EXCLUDE (Zeilen **2** und **7**) überschneidet sich mit einer Aktion INCLUDE (Zeilen **4** und **9**). Eine Fehlermeldung wird ausgegeben.

Beispiel 3


```

// *%OPC SCAN
// *%OPC BEGIN ACTION=INCLUDE
// EXEC PGM=MYPROG
// *%OPC BEGIN ACTION=NOSCAN
// DDNAME1 DD DSN=&HIONE..&DATASET1,DISP=SHR
// SYSOUT DD SYSOUT=A
// *%OPC END ACTION=NOSCAN
// DDNAME2 DD DSN=&HIONE..&DATASET2,DISP=SHR
// *%OPC END ACTION=INCLUDE

```

1
2
3
4
5
6
7
8
9

Beispiel 3 ist gültig. Der NOSCAN-Bereich, der in den Zeilen **4** und **7** definiert wird, ist vollständig innerhalb den Bereichs von BEGIN und END mit ACTION=INCLUDE (Zeilen **2** und **9**) enthalten.

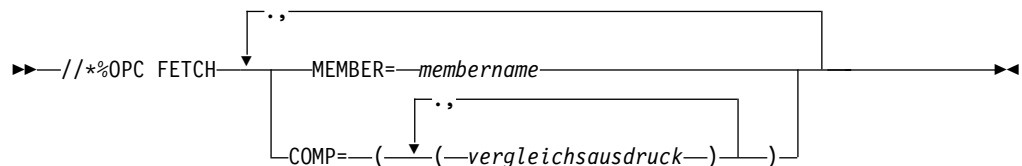
Beachten Sie außerdem, dass die Variable HIONE in Zeile **8** substituiert wird, während die Variable HIONE in Zeile **5** nicht substituiert wird, weil sie innerhalb eines NOSCAN-Bereichs liegt.

Anweisung FETCH:

Zweck

Mithilfe dieser Anweisung können Sie Zeilen, die aus einem Member einer partitionierten Datei abgerufen oder durch einen Exit bereitgestellt werden, in Ihren Job einschließen.

Syntax



Parameter

MEMBER=(membername)

Gibt den Membernamen einer partitionierten Datei an, die dem Datendefinitionsnamen EELJBLIB zugeordnet ist. Die Zeilen in diesem Member werden unmittelbar nach der Anweisung FETCH eingefügt.

COMP=((vergleichsausdruck), (vergleichsausdruck),...)

Gibt Vergleichsausdrücke an, die zur Entscheidung verwendet werden, ob die Anweisung FETCH berücksichtigt werden soll. Wenn der Vergleichsausdruck wahr ist, wird die Anweisung FETCH berücksichtigt. Details zum Definieren von Vergleichsausdrücken finden Sie in „Schlüsselwort COMP in Anweisungen BEGIN und FETCH“ auf Seite 94.

Hinweise

Die Anweisung FETCH dient zum Einschließen von Zeilen aus anderen partitionierten Dateien oder von Zeilen, die durch einen Exit bereitgestellt werden. Zeilen, die durch eine Anweisung FETCH eingefügt werden, können keine weitere Anweisung FETCH enthalten. Anweisungen BEGIN und END mit der Aktion INCLUDE oder EXCLUDE können nicht in Zeilen enthalten sein, die durch eine Anweisung FETCH eingefügt werden.

Vordefinierte Variablen können zum Darstellen der Werte beliebiger Schlüsselwörter, jedoch nicht der Schlüsselwörter selbst verwendet werden. Eine Anweisung `FETCH` kann nicht innerhalb eines `BEGIN/END`-Paares stehen, das `ACTION=INCLUDE` oder `ACTION=EXCLUDE` angibt.

Beispiel

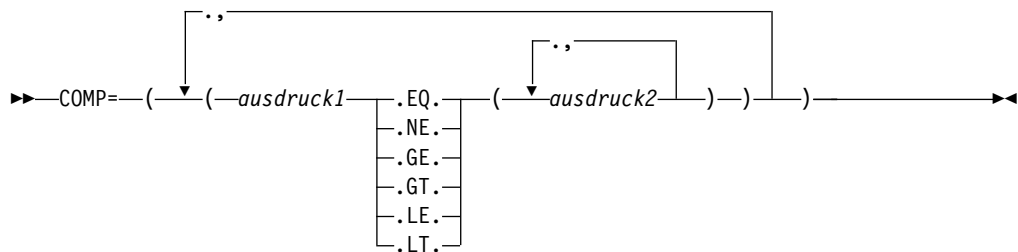
```
//*%OPC SCAN
//*%OPC FETCH,
//*%OPC MEMBER=JCL1,
//*%OPC COMP=(&DAY..EQ.1)
```

Schlüsselwort `COMP` in Anweisungen `BEGIN` und `FETCH`:

Zweck

Ein Vergleichsausdruck ermöglicht die Angabe von Bedingungen für die Berücksichtigung von Anweisungen `BEGIN` und `FETCH`.

Syntax



Parameter

ausdruck1

Diese Angabe ist eine Zeichenfolge, die aus `&`-Variablen und alphanumerischen Literalen besteht. Alle enthaltenen Zeichen für globale Suche werden als Literale behandelt. Der Wert von *ausdruck1*, der aus der Auflösung der angegebenen Variablen resultiert, wird an den Werten getestet, die durch *ausdruck2* angegeben werden.

.Operatoren.

Diese Werte sind Operatoren, die angeben, welcher Vergleich zwischen dem Wert von *ausdruck1* und beliebigen Werten von *ausdruck2* durchgeführt werden soll.

- .EQ.** Der Wert von *ausdruck1* muss gleich einem der Werte von *ausdruck2* sein, damit der Ausdruck wahr ist.
- .NE.** Alle Werte von *ausdruck2* müssen ungleich dem Wert von *ausdruck1* sein, damit der Ausdruck wahr ist.
- .GT.** Der Wert von *ausdruck1* muss größer als der Wert von *ausdruck2* sein, damit der Ausdruck wahr ist.
- .GE.** Der Wert von *ausdruck1* muss größer-gleich dem Wert von *ausdruck2* sein, damit der Ausdruck wahr ist.
- .LT.** Der Wert von *ausdruck1* muss kleiner als der Wert von *ausdruck2* sein, damit der Ausdruck wahr ist.
- .LE.** Der Wert von *ausdruck1* muss kleiner-gleich dem Wert von *ausdruck2* sein, damit der Ausdruck wahr ist.

ausdruck2

Dieser Parameter kann aus &-Variablen, Literalen oder, bei Angabe der Operatoren .EQ. oder .NE., einem der beiden Zeichen für globale Suche (% und *) bestehen.

Die Länge des aufgelösten Werts darf 44 Zeichen nicht überschreiten. Das Zeichen für globale Suche % stellt ein beliebiges einzelnes alphanumerisches Zeichen dar. Das Zeichen für globale Suche * stellt eine beliebige alphanumerische Zeichenfolge, einschließlich der Nullzeichenfolge, dar.

Wenn GT, GE, LT oder LE angegeben wird, gilt:

- Mehrere Werte in *ausdruck2* werden nicht unterstützt.
- Zeichen für globale Suche werden nicht unterstützt.
- Wenn die Zeichenfolgen auf beiden Seiten der Operatoren verschiedene Längen haben, erfolgt der Vergleich unter Verwendung der kürzeren Zeichenfolge.

Anmerkung: Das Symbol % bezeichnet innerhalb eines COMP-Schlüsselworts keine vordefinierte %-Variable. In einer Anweisung COMP sind %- und ?-Variablen nicht gültig.

Hinweise

Der COMP-Ausdruck kann vor der Auflösung nicht länger als 256 Zeichen sein und nach der Substitution nicht mehr als 1024 Zeichen enthalten. Der Wert *ausdruck2* kann eine beliebige vordefinierte oder benutzerdefinierte &-Variable sein. Weder *ausdruck1* noch *ausdruck2* darf eingebettete Leerzeichen enthalten.

Betrachten Sie die folgenden Beispiele:

Beispiel 1

```
//*%OPC FETCH,  
//*%OPC MEMBER=MYJCL,  
//*%OPC COMP=(&APPL..EQ.(APPL1,APPL2,APPL3))
```

Wenn &APPL. gleich APPL1 oder APPL2 oder APPL3 ist, ist der Ausdruck wahr und die Anweisung FETCH wird berücksichtigt.

Beachten Sie die beiden Punkte nach &APPL.. Der erste bezeichnet das Ende der Variablen APPL. Der zweite bezeichnet den Anfang des Vergleichsoperators EQ.

Beispiel 2

```
//*%OPC FETCH,  
//*%OPC MEMBER=MYJCL,  
//*%OPC COMP=(&DAY..NE.(1,3,5))
```

Wenn in Beispiel 2 die Variable &DAY nicht den Wert 1 oder 3 oder 5 hat, ist der Ausdruck wahr und die Anweisung FETCH wird berücksichtigt. Wenn DAY gleich einem der Vergleichswerte ist, hat der Ausdruck den Wert 'Falsch'.

Damit das Schlüsselwort COMP den Wert *wahr* liefert, müssen alle Vergleichsausdrücke, aus denen es besteht, den Wert *wahr* liefern. Dies wird im folgenden Beispiel demonstriert:

Beispiel 3

```

//%OPC BEGIN ACTION=INCLUDE,
//%OPC      COMP=((&APPL..EQ.(APPL1,APPL2,APPL3)),
//%OPC      (&DAY..NE.(1,3,5)))
.
.
.
//%OPC END ACTION=INCLUDE

```

Damit die Anweisung COMP in Beispiel 3 den Wert 'wahr' liefert, müssen die Ausdrücke (&APPL..EQ.(APPL1,APPL2,APPL3)) und (&DAY..NE.(1,3,5)) beide wahr sein.

Die Werte in *ausdruck2*, die Sie angeben, können aus &-Variablen, alphanumerischen Literalen sowie den Zeichen für globale Suche '*' und '%' bestehen. Nationale Sonderzeichen, linke und rechte Klammern, '(' und ')', sowie Leerzeichen sind nicht zulässig. Wenn sie angegeben werden, sind die Ergebnisse unvorhersehbar. Das Zeichen für globale Suche '*' stellt eine Zeichenfolge beliebiger Länge dar; das Zeichen für globale Suche '%' stellt genau 1 Zeichen dar. Wenn Variablen und Zeichen für globale Suche kombiniert werden, werden die Variablen aufgelöst, bevor Vergleiche unter Verwendung der Zeichen für globale Suche durchgeführt werden.

Beispiel 4

```

//%OPC BEGIN ACTION=INCLUDE,
//%OPC      COMP=(&MYVAR..EQ.(TS0199,TS02%.,&VALUE1.*))
.
.
.
//%OPC END ACTION=INCLUDE

```

In Beispiel 4 muss die Variable &MYVAR einen der folgenden Werte haben, damit der Vergleichsausdruck wahr ist:

- TSO199
- TSO2 gefolgt von zwei beliebigen alphanumerischen Zeichen außer Leerzeichen
- Wert der Variablen &VALUE1, gefolgt von einer alphanumerischen Zeichenfolge beliebiger Länge (einschließlich der Länge 0)

Beispiel 5

```

//%OPC BEGIN ACTION=EXCLUDE,
//%OPC      COMP=(&CYMMDD..GE.120101)
.
.
.
//%OPC END ACTION=EXCLUDE

```

Beachten Sie, dass Anweisungen COMP bei einigen der Datumsformate der vordefinierten Variablen unerwartete Ergebnisse liefern können. Wenn Datumsvariablen substituiert werden, werden sie als Numerale und nicht als Datumswerte verglichen.

In Beispiel 5 ist &CYMMDD das aktuelle Datum und 120101 stellt den 1. Januar 2012 dar. Wenn der Wert von *ausdruck1* größer als 120101 ist, ist der Ausdruck wahr.

Jobinstanzen verwalten

Zur Verwaltung der Jobs, die an einen Agenten für z/OS übergeben werden, verwenden Sie dieselben Befehle, die für alle IBM Workload Scheduler-Jobs verfügbar sind.

Verwenden Sie die Conman-Befehlszeile oder Dynamic Workload Console, um die Jobinstanzen zu verwalten. Die meisten Jobverwaltungsbefehle und -aktionen von IBM Workload Scheduler sind auch für Jobinstanzen des Agenten für z/OS anwendbar, jedoch nicht alle. In den folgenden Abschnitten sind Befehle und Aktionen aufgelistet, die Sie ausführen können, und solche, die Sie nicht ausführen können.

Befehle und Aktionen, die Sie ausführen können

Alle mit Ausnahme von "kill" für Jobinstanzen.

Befehle und Aktionen, die Sie nicht ausführen können

Sie können den Befehl kill nicht für Jobinstanzen ausführen.

Dies schränkt die Verwendung des Jobstromschlüsselworts maxdur für Jobs des Agenten für z/OS ein. Wenn Sie also in einem Jobstrom das Schlüsselwort maxdur mit dem Argument onmaxdur kill angeben, wird der Job nicht beendet, wenn er das Zeitlimit überschreitet, obwohl die Markierungen MaxDurationExceeded und KillSubmitted vom Befehl conman showjobs zurückgegeben werden. Im Bereich der ereignisgesteuerten Auslastungsautomatisierung wird das Ereignis MaxDurationExceeded generiert. Dieses Ereignis kann zum Auslösen einer Reihe von Aktionen verwendet werden, jedoch nicht zum Auslösen der Aktion KillJob.

Jobs überwachen

Der Agent verwendet die JES-/SMF-Exits, um die Statusänderungen für alle Jobs zu überwachen, die durch ihn übergeben werden.

Jobstatus

Die Statusereignisse für die Jobs, die über den Agenten für z/OS übergeben werden, werden normalerweise von JES und SMF ausgegeben und vom Agenten für z/OS abgefangen. Der Agent leitet sie anschließend in einer XML-Nachricht an Dynamic Workload Broker weiter. Die Jobstatus werden in Dynamic Workload Console und in der Conman-Befehlszeile zurückgemeldet.

In der folgenden Tabelle sind die Status zusammengefasst, die ein Job von der Übergabe bis zum Abschluss durchläuft. Zudem sind die entsprechenden Statusnamen aufgeführt, die von JES/SMF, von dem Agenten für z/OS und in den Benutzerschnittstellen zurückgemeldet werden.

Tabelle 20. Jobereignisse und -status in den verschiedenen Komponenten

Ereignisbeschreibung	Internes Ereignis des Agenten	Vom Agenten für z/OS an Dynamic Workload Broker in XML weitergeleiteter Statusname	In Dynamic Workload Console und Conman angezeigter Statusname	Weitere Details
Der Agent hat einen Fehler in der JCL gefunden, die in dem Job enthalten ist, und stoppt die Verarbeitung des Jobs.	IJO	-	FAIL	Dieses Ereignis wird vom Agenten für z/OS generiert, wenn Dynamic Workload Broker den Job übergibt.

Tabelle 20. Jobereignisse und -status in den verschiedenen Komponenten (Forts.)

Ereignisbeschreibung	Internes Ereignis des Agenten	Vom Agenten für z/OS an Dynamic Workload Broker in XML weitergeleiteter Statusname	In Dynamic Workload Console und Conman angezeigter Statusname	Weitere Details
<p>Job-JCL. Ein Job wurde übergeben.</p> <p>Der Job wurde von Dynamic Workload Broker übergeben und die JCL wurde vom Agenten für z/OS in den internen JES-Leser gestellt. Der Job wurde noch nicht von JES übergeben, aber die IDs, die Dynamic Workload Broker und JES dem Job zugeordnet haben, wurden einander zugeordnet.</p>	KJ1	INFO	BOUND	Dieses Ereignis wird vom Agenten für z/OS generiert, nachdem die JCL in den internen JES-Leser kopiert wurde.
<p>Lesereignis: Ein Job ist in das JES-System eingetreten.</p>	A1 oder B1	SUBMITTED	WAIT	Dieses Ereignis wird von JES auf dem System generiert, auf dem der Agent für z/OS installiert ist, und vom Agenten für z/OS an Dynamic Workload Broker gesendet.
<p>Jobstartereignis: Die Ausführung eines Jobs wurde gestartet.</p>	A2 oder B2	EXECUTING	EXEC	Dieses Ereignis wird von SMF auf dem System generiert, auf dem der Job übergeben wird, und vom Agenten für z/OS an Dynamic Workload Broker gesendet.
<p>Schrittdeereignis: Ein Jobschritt wurde fertig ausgeführt.</p>	A3S oder B3S	EXECUTING	EXEC	Dieses Ereignis wird auf dem System generiert, auf dem der Job übergeben wird. Es wird jedoch gelöscht, wenn kein ABEND bei einem Jobabschnitt auftritt.
<p>Jobendereignis: Die Ausführung eines Jobs wurde fertig gestellt.</p>	A3J oder B3J	SUCCEEDED EXECUTION oder FAILED EXECUTION	SUCC oder ABEND	Dieses Ereignis wird von SMF auf dem System generiert, auf dem der Job übergeben wird, und vom Agenten für z/OS an Dynamic Workload Broker gesendet.

Tabelle 20. Jobereignisse und -status in den verschiedenen Komponenten (Forts.)

Ereignisbeschreibung	Internes Ereignis des Agenten	Vom Agenten für z/OS an Dynamic Workload Broker in XML weitergeleiteter Statusname	In Dynamic Workload Console und Conman angezeigter Statusname	Weitere Details
Jobabschlussereignis: Ein Job wurde der JES-Ausgabewarteschlange hinzugefügt.	A3P oder B3P	SUCCEEDED EXECUTION oder FAILED EXECUTION	SUCC oder ABEND	Dieses Ereignis wird von JES auf dem System generiert, auf dem der Job übergeben wird, und vom Agenten für z/OS an Dynamic Workload Broker gesendet.
Jobausgabeende	A4 oder B4	-	-	Optionales Ereignis, das vom JCL-Inhalt abhängig ist. Das Ereignis wird in der Ereignisdatei (EELEVDS) aufgezeichnet, jedoch nicht an Dynamic Workload Broker weitergeleitet.
Job wird bereinigt.	A5 oder B5	-	-	Optionales Ereignis, das vom JCL-Inhalt abhängig ist. Das Ereignis wird in der Ereignisdatei (EELEVDS) aufgezeichnet, jedoch nicht an Dynamic Workload Broker weitergeleitet.

JCL-Fehler

Der Agent analysiert die von Dynamic Workload Broker gesendete JCL, bevor er sie an JES übergibt. Stellt er einen Syntaxfehler in der JCL fest, stoppt er den Übergabeprozess und zeichnet die Fehlernachricht im Jobprotokoll auf.

Jobfehlercodes

Treten nach der Übergabe eines Jobs andere Rückgabecodes als null auf, handelt es sich um Fehler. Der Rückgabecode wird vom Agenten für z/OS über HTTP in einem XML-POST zusammen mit der ID und dem Namen des Jobs an Dynamic Workload Broker zurückgesendet. Der z/OS-Fehlercode wird einem numerischen Code zugeordnet, der in Conman oder Dynamic Workload Console angezeigt wird. In der folgenden Tabelle sind die Fehlercodes aufgelistet, die nach der Übergabe eines Jobs zurückgegeben werden können. Zudem sind ihre Entsprechungen in z/OS und IBM Workload Scheduler aufgeführt.

Tabelle 21. Zurückgegebene Fehlercodes nach der Übergabe eines Jobs

In IBM Workload Scheduler zugeordneter Rückgabecode	Fehlerbeschreibung	Als erweiterte Jobeigenschaft in Conman oder Dynamic Workload Console angezeigt
1nnnn (nnnn ist die in Dezimalziffern konvertierte Zeichenfolge xxx)	Fehlercodes für Systemabbruch in Hexadezimalschreibweise.	Sxxx
2nnnn (nnnn ist die in Dezimalziffern konvertierte Zeichenfolge xxx)	Fehlercodes für Benutzerabbruch in Hexadezimalschreibweise.	Uxxx
30007	Ein JCL-Fehler ist sofort aufgetreten; d. h., der Fehler wurde festgestellt, bevor der Job gestartet wurde. Dieser Code ist auch möglich, wenn sowohl das Jobstartereignis (Typ A2) als auch das Jobendereignis (Typ A3J) fehlen.	JCLI
30020	Ein Fehler ist aufgetreten, als der Agent versucht hat, einen Job zu übergeben. Die Operation sollte als 'mit Fehler beendet' markiert werden.	OSUB
30021	Ein Fehler ist aufgetreten, als der Agent versucht hat, die JCL für einen Job abzurufen.	OSUF
30026	Der Job wurde vor der Ausführung vom Bediener oder von einem TSO-Benutzer abgebrochen. Dieser Code ist auch möglich, wenn das Jobabschlussereignis (Typ A3P) fehlt.	CAN
30027	Der Beendigungscode ist unbekannt. Der Job wurde beendet, aber es ist kein Beendigungscode verfügbar. Dieser Code ist auch möglich, wenn das Jobendereignis (Typ A3J) fehlt.	CCUN
30029	Ein JCL-Fehler wurde erkannt, nachdem die Jobausführung begonnen hat, oder ein JCL-Fehler wurde nach der Syntaxprüfung im internen Leser erkannt.	JCL
90000	Benutzerdefinierte Fehlercodes in Hexadezimalschreibweise.	xxxx

Ereignisdateien

Der Agent verwendet die Ereignisdateien EELHTDS und EELEVDS. Die darin enthaltenen Datensätze beschreiben die Ereignisse, die von den Jobüberwachungsfunktionen des Agenten erstellt werden. Eine Ereignisschreibertask schreibt Datensätze in diese Dateien und liest Datensätze aus diesen Dateien.

EELHTDS zeichnet die Ereignisse auf, die von Dynamic Workload Broker erstellt wurden und sich auf die Übergabe von Workloads beziehen.

EELEVDS zeichnet die von JES und SMF erstellten Ereignisse auf, die sich auf die Jobausführung beziehen und die an Dynamic Workload Broker zurückgesendet werden sollen.

Eine weitere Datei mit dem Namen EELHTREF wird als Servicedatei verwendet, um die Jobs, die in den Übergabeanforderungen aus IBM Workload Scheduler gesendet werden, kurz zu speichern. Wenn die Jobs einen Abruf einer im z/OS-System gespeicherten JCL oder eine Variablensubstitution erfordern, werden diese Aktionen hier ausgeführt.

Da alle Ereignisse in den Ereignisdateien aufgezeichnet sind, gehen Ereignisse nicht verloren, wenn der Agent oder eine Komponente zur Ereignisverarbeitung erneut gestartet werden muss. Der Prozess für das Übergabepunktverfahren stellt sicher, dass Übergabeanforderungen mit Dynamic Workload Broker synchronisiert sind. Auf diese Weise wird verhindert, dass Anforderungen aufgrund von Kommunikationsfehlern verloren gehen.

Wichtig: Die Dateien werden bei ihrer ersten Verwendung formatiert. Wenn Sie eine der Dateien aus irgendeinem Grund erneut formatieren, gehen alle Ereignisse verloren, die sich in der Warteschlange für alle Dateien befinden.

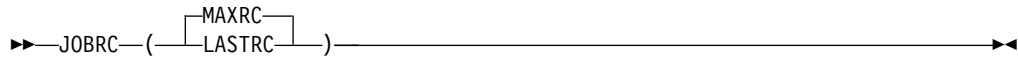
Steuern, wie der Ereignisschreiber Jobbeendigungscode für bestimmte Jobs aufzeichnet

Unter z/OS 1.13 und höheren Versionen können Sie den Parameter JOBRC in Jobkartenanweisungen bestimmter Jobs angeben, um vorzudefinieren, wie die Beendigungscode dieser Jobs in der Ereignisdatei EELEVDS aufgezeichnet werden.

Für alle Jobs, die durch den Agenten für z/OS an JES übergeben werden, definiert das Schlüsselwort RETCODE der Initialisierungsanweisung EWTROPTS, welchen Beendigungscode der Ereignisschreiber in der Ereignisdatei EELEVDS für den Jobendeereignissatz (A3J) aufzeichnet, wobei eine Auswahl unter den Codes erfolgt, die von den Jobabschnitten zurückgegeben werden. Die Standardauswahl für RETCODE besteht darin, den Jobbeendigungscode auf den Rückkehrcode des letzten Jobabschnitts (Schritts) zu setzen. Alternativ können Sie ihn so festlegen, dass der höchste Rückkehrcode eines Jobabschnitts ausgewählt wird.

Für bestimmte Jobs Ihrer Wahl können Sie die Einstellung von RETCODE überschreiben, indem Sie den Parameter JOBRC in der JOB-Anweisung der JCL angeben. In der Regel verwenden Sie den Parameter JOBRC, um für einen bestimmten Job die in EWTROPTS RETCODE definierte Einstellung, die im Allgemeinen für alle Auslastungen (Workloads) gilt, die vom Agenten für z/OS an JES übergeben werden, zurückzunehmen.

JOBRC



Parameter

MAXRC

Der Jobbeendigungscode wird auf den höchsten Rückkehrcode eines Abschnitts im Job oder, falls die Ausführung des Jobs wegen einer abnormalen Beendigung fehlschlägt, auf den letzten ABEND-Code gesetzt. Dies ist die Standardeinstellung des Parameters.

LASTRC

Der Jobbeendigungscode wird auf den Rückkehrcode bzw. den ABEND-Code des letzten Abschnitts gesetzt, der im Job ausgeführt wird.

Beispiel

Die Initialisierungsanweisung EWTROPTS Ihres Agenten für z/OS gibt RETCODE=HIGHEST an. Dies legt fest, dass für alle Jobs, die vom Agenten an JES übergeben werden, der Jobbeendigungscode, der vom Ereignisschreiber zurückgegeben wird, jeweils der höchste Rückkehrcode aller ausgeführten Abschnitte ist.

Für den aus mehreren Abschnitten bestehenden Job ACCT1254JCL soll der Ereignisschreiber jedoch den Rückkehrcode des letzten ausgeführten oder abnormal beendeten Abschnitts auswählen. Um dies festzulegen, schreiben Sie den Parameter JOBRC=LASTRC in die JOB-Anweisung von ACCT1254JCL.

Jobprotokolle anzeigen

Anzeigen der Protokolle von Jobs, die vom Agenten für z/OS übergeben wurden.

Sie können die Protokolle der Jobs, die vom Agenten für z/OS übergeben wurden, in Dynamic Workload Console oder in der Conman-Befehlszeile anzeigen.

Die hierfür auszuführende Prozedur ist für alle IBM Workload Scheduler-Jobs gleich:

- Klicken Sie in Dynamic Workload Console auf **Systemstatus und Zustand** → **Auslastungsüberwachung** → **Jobs überwachen**. Anschließend klicken Sie für eine ausgewählte Jobinstanz auf die Schaltfläche **Jobprotokoll...**
- In der IBM Workload Scheduler-Befehlszeile führen Sie `conman showjobs` aus.

Wenn Sie eine dieser Prozeduren ausführen, werden die Protokolle vom z/OS-System abgerufen und angezeigt. Die Protokolle sind nur für Jobs verfügbar, die abgeschlossen sind.

Die Protokolle sind nur so lange für die Anzeige verfügbar, bis sie von JES bereinigt werden. Nach diesem Zeitpunkt stehen sie nicht mehr zur Verfügung.

Systembefehle zur Steuerung des Agenten verwenden

Sie können z/OS-Systembefehle verwenden, um den Agenten für z/OS zu starten, zu stoppen, abzubrechen oder zu ändern.

Verwenden Sie die folgenden Bedienerbefehle zur Steuerung des Agenten:

S	START
P	STOP
C	CANCEL
F	MODIFY

Sie können diese Befehle an einer MSC-Konsole (MSC - Multiple Console Support) oder mithilfe eines Programms wie SDSF (Spool Display and Search Facility) eingeben. In beiden Fällen muss der Bediener des Terminals oder der Konsole über die erforderlichen Berechtigungen zum Eingeben von Bedienerbefehlen verfügen.

Agent starten

Zum Starten des Agenten für z/OS geben Sie den folgenden z/OS-Bedienerbefehl ein:

```
/S proz-name
```

Dabei ist *proz-name* der Prozedurname der JCL oder gestarteten Task des Agenten für z/OS.

Soll der Agent als Stapeljob ausgeführt werden, dürfen Sie ihn nicht mit einem Bedienerbefehl starten. Übergeben Sie stattdessen einen Stapeljob mit dem gleichen Namen wie das Subsystem des Agenten für z/OS. JES startet diesen Job auf die gleiche Weise wie einen normalen Job.

Da der Agent für z/OS u. a. JES-Exits verwendet, um den Fortschritt von z/OS-Jobs zu überwachen, wird er erst gestartet, wenn JES aktiv ist.

Agent stoppen

Zum Stoppen des Agenten für z/OS geben Sie den folgenden z/OS-Bedienerbefehl ein:

```
/P proz-name
```

Dabei ist *proz-name* der Name der JCL-Prozedur für den Agenten für z/OS.

Agent abbrechen

Wenn der Agent für z/OS 5 Minuten nach der Eingabe des Bedienerbefehls STOP noch immer aktiv ist, müssen Sie ihn abbrechen.

Zum Abbrechen des Agenten für z/OS geben Sie den folgenden z/OS-Bedienerbefehl ein:

```
/C proz-name
```

Dabei ist *proz-name* der Name der JCL-Prozedur für den Agenten für z/OS.

Ist der Befehl STOP unwirksam und verfügen Sie noch nicht über eine Dokumentation des Problems, fügen Sie die Option DUMP beim Abbrechen des Agenten hinzu, sodass das Problem bestimmt werden kann. Geben Sie Folgendes ein:

```
/C proz-name,DUMP
```

Dies bewirkt, dass der Agent für z/OS beendet und ein Speicherauszug in die Datei SYSMDUMP gestellt wird (falls der Datendefinitionsname sich in der JCL der gestarteten Task befindet).

Agent ändern

Verwenden Sie den Befehl MODIFY, um eine der folgenden Subtasks des Agenten für z/OS zu starten oder zu stoppen:

- Datenrouter
- Ereignisschreiber
- Übergabe

Außerdem können Sie mit diesem Befehl den Status der Subtasks auflisten.

Der Befehl MODIFY hat die folgende Syntax:

```
/F prozedurname,änderungsoption
```

Dabei gilt Folgendes:

- *prozedurname* ist der Name der JCL-Prozedur für den Agenten für z/OS.
- *änderungsoption* kann Folgendes sein:

STATUS, SUBTASK

Listet alle Subtasks mit ihrem Status auf. Der Status kann ACTIVE oder INACTIVE sein.

S=*subtask*

Startet die angegebene Subtask.

P=*subtask*

Stoppt die angegebene Subtask.

subtask ist eine der folgenden Subtasks des Agenten für z/OS:

DRT Datenrouter

EWTR

Ereignisschreiber

SUB Übergabe

Beachten Sie Folgendes: Wird der Ereignisschreiber gestoppt, wird der Agent für z/OS in den Status OFFLINE versetzt (als UNAVAILABLE in Dynamic Workload Broker angezeigt). Für Jobs, die übergeben werden, während der Agent offline ist, wird der Status READY angezeigt. Jobs, die übergeben, jedoch noch nicht ausgeführt waren, als der Agent in den Status OFFLINE eintrat, werden in den Status WAIT versetzt. Wenn das Zeitlimit abläuft, das für Dynamic Workload Broker konfiguriert wurde, werden sie in den Status FAIL versetzt.

Sie können nur eine Task starten, die zuvor in der aktuellen Sitzung gestoppt wurde. Wenn Sie versuchen, eine gestartete Subtask zu starten oder eine gestoppte Subtask zu stoppen, wird die Fehlermeldung EELZ049W ausgegeben und keine Aktion ausgeführt.

Domänenmanager wechseln

In diesem Abschnitt ist der Mechanismus beschrieben, der bewirkt, dass der Agent für z/OS mit Ihrem IBM Workload Scheduler-Netz verbunden bleibt, wenn Sie zu einem Sicherungsmanager wechseln.

Reaktion des Agenten auf einen Wechsel des Domänenmanagers

Jedes Mal, wenn Sie den Domänenmanager (oder Master) für die Dynamic Workload Broker-Instanz ändern, mit der der Agent für z/OS verbunden ist, wird die Verbindung zwischen dem Agenten und seinem Dynamic Workload Broker unterbrochen. Der HTTP-Client des Agenten für z/OS ist so konzipiert, dass er die nächste verfügbare Dynamic Workload Broker-Instanz (d. h. die Instanz, die in dem frisch aktivierten Domänenmanager ausgeführt wird) sucht und eine Verbindung zu dieser herstellt, sobald er die Unterbrechung erkennt.

Nach seiner Initialisierung sendet der Agent für z/OS in regelmäßigen Intervallen eine Pinganforderung an den Dynamic Workload Broker. Jede Pinganforderung ist eine HTTP-POST-Anforderung, in der der Agent seine Kennung und andere Attribute als Dynamic Workload Broker-Ressource sendet. Nachdem der Dynamic Workload Broker die HTTP-Anforderung akzeptiert und verarbeitet hat, sendet er als Antwort die Liste der zurzeit definierten Dynamic Workload Broker-Sicherungsinstanzen. Diese Liste basiert auf der Liste der IBM Workload Scheduler-Agenten, die als Sicherungsdomänenmanager definiert sind.

Nach der ersten erfolgreichen Pinganforderung verfügt der Agent für z/OS über eine Liste aller Dynamic Workload Broker-Sicherungsinstanzen, die im IBM Workload Scheduler-Netz verfügbar sind. Diese Liste wird bei jeder nachfolgenden Pinganforderung aktualisiert. Ist die Pinganforderung des Agenten an den Dynamic Workload Broker bei seinem Start nicht erfolgreich, ist keine Liste der Sicherungen verfügbar und der Wechsel zu einer Sicherungsinstanz des Dynamic Workload Broker ist nicht möglich.

Tritt ein Netzfehler auf, während der Agent für z/OS eine Anforderung an den aktuellen Dynamic Workload Broker ausgibt, wird ein Offlineereignis generiert. Dieses Offlineereignis löst den Mechanismus aus, durch den der HTTP-Client in dem Agenten für z/OS eine Pinganforderung an den nächsten Dynamic Workload Broker in der Liste sendet.

- Ist diese Dynamic Workload Broker-Instanz verfügbar und wird eine Verbindung hergestellt, wird diese Instanz zur neuen Dynamic Workload Broker-Instanz, mit der der Agent für z/OS interagiert. Der neue Dynamic Workload Broker stellt auch eine aktualisierte Liste der Dynamic Workload Broker-Sicherungsinstanzen bereit.
- Ist die Instanz nicht verfügbar, sendet der HTTP-Client eine Pinganforderung an die nächste Instanz in der Liste usw. Ist die Anforderung auch bei der letzten Instanz ohne Erfolg, wird wieder am Anfang der Liste begonnen. Dieser Prozess wird fortgesetzt, bis eine Pinganforderung erfolgreich an eine der Dynamic Workload Broker-Instanzen gesendet wird.

Agent stoppen und erneut starten, nachdem der primäre Dynamic Workload Broker sich geändert hat

Ab der erstmaligen Herstellung einer Verbindung zum Dynamic Workload Broker ist die Liste der Sicherungsinstanzen im Hauptspeicher des Agenten gespeichert und sie verbleibt dort, solange der Agent ausgeführt wird. Sie geht verloren, wenn

Sie den Agenten stoppen. Wenn Sie den Agenten stoppen und erneut starten, sendet er eine Pinganforderung an die ursprüngliche Dynamic Workload Broker-Instanz, die in seinen Konfigurationsparametern angegeben ist. Ist diese Instanz nicht verfügbar, weil ein Wechsel durchgeführt wurde oder die Instanz inzwischen inaktiv ist, kann der Agent keine Verbindung zu einer Sicherungsinstanz herstellen, da er noch nicht über eine Liste verfügt. Wenn Sie den Agenten stoppen und erneut starten, nachdem sich der primäre Dynamic Workload Broker geändert hat, vergessen Sie also nicht, in der Agentenkonfiguration die TDWBHOSTNAME- und TDWBPORT-Werte des neuen primären Dynamic Workload Broker anzugeben. Nachdem der Agent eine Verbindung zu dem neuen Dynamic Workload Broker hergestellt hat, empfängt er wieder die Liste.

Kapitel 4. Fehlerbehebung und Referenz

In diesem Kapitel sind potenzielle Probleme und Referenzinformationen dokumentiert.

Informationen zu Resynchronisationsnachrichten

In diesem Abschnitt werden die Informationsnachrichten beschrieben, die der Agent für z/OS ausgibt, wenn er die Jobübergabe nach einem Neustart erneut mit Dynamic Workload Broker synchronisiert.

Übersicht über die Resynchronisation

Die erste Task, die der Agent für z/OS versucht, wenn er nach einem geplanten oder unfreiwilligen Stopp erneut gestartet wird, ist die ordnungsgemäße Synchronisation mit Dynamic Workload Broker, sodass keine Informationen zu der Workload verloren gehen, die vom Agenten verarbeitet wurde, als es zu dem Stopp gekommen ist.

Wie in „Jobs überwachen“ auf Seite 97 beschrieben, liegen die Informationen zur Workload, die dem Agenten zugewiesen ist, in Form von Ereignissen vor, die die Arbeit, die noch zu übergeben ist, und das Ergebnis der bereits verarbeiteten Arbeit aufzeichnen. Bei der normalen Verarbeitung verwendet der Agent drei Ereignisdateien, um diese Ereignisse zu verwalten. Die gleichen Dateien werden auch zu Wiederherstellungszwecken nach einem Neustart verwendet. Diese sind (wie sie mit ihrem DD-Namen in der gestarteten Task des Agenten für z/OS bekannt sind):

EELEVDS

Speichert die Ereignisse, die den gesamten Lebenszyklus eines übergebenen Jobs (Übergabe, Ausführung, Ergebnis) verfolgen.

Nach einem Stopp/Neustart durchsucht der Agent diese Datei nach den zuletzt ausgegebenen Ereignissen, um sie erneut an Dynamic Workload Broker zu senden.

EELHTREF

Speichert die Jobübergabeanforderungen, die von Dynamic Workload Broker empfangen wurden. Enthält Informationen zum Typ der Übergabe (nach Verweis oder nach Definition), zur JCL bzw. ihrer Position und zur Variablen-tabelle (sofern zutreffend). Sobald eine in EELHTREF gespeicherte Anforderung vollständig verarbeitet wurde (der Job wurde in die Warteschlange für die Übergabetask submit gestellt), wird sie als abgeschlossen markiert.

Nach einem Stopp/Neustart durchsucht der Agent diese Datei nach unvollständigen Anforderungen und verarbeitet diese erneut.

EELHTDS

Speichert die JCLs, die für die Übergabetask (submit) in die Warteschlange gestellt wurden.

Nach einem Stopp/Neustart durchsucht der Agent diese Datei nach in der Warteschlange befindlichen JCLs, die noch nicht übergeben wurden, um sie erneut in die Warteschlange zu stellen und ihre Verarbeitung abzuschließen.

Resynchronisationsnachrichten

Nach einem Stopp/Neustart gibt der Agent eine Reihe von Informationsnachrichten aus, die die Aktionen beschreiben, die er zur Resynchronisation ausführt. Obwohl sie keine Benutzeraktion erfordern, werden sie hier als Hilfe zur Verfolgung der Resynchronisationsverarbeitung dokumentiert.

EELHT28I

EELHTDS REPROCESSING CAUSED RESUBMISSION OF FOLLOWING EVENT:
SSEQ: SEQ RECORD: REC CYCLE: CYC JOBALIAS:
JOBALIA1
JOBALIA2

Erläuterung: Beim Verarbeiten der Datei EELHTDS (Wiederherstellungsdatei für anstehende Übergaben) nach einem Neustart hat der Agent den Job, der durch den in *JOBALIA1* und *JOBALIA2* angegebenen Aliasnamen identifiziert wird, im anstehenden Status gefunden.

Der EELHTDS-Satz, in dem der Job gefunden wurde, wird durch die Satznummer *REC* und den Schreibzyklus *CYC* angegeben.

Systemaktion: Der Agent sendet den Job erneut an die Übergabetask.

EELHT36I

INFORMATION ABOUT RESTART PROCESSING OF EELHTDS DATA SET FOLLOWS:
HTDS NEXT TO WRITE : REC *INREC* CYC *INCYC*
HTDS RESEND START POSITION: REC *STREC* CYC *STCYC*
HTDS RESEND STOP POSITION : REC *CUREC* CYC *CUCYC*
HTDS LAST SUBMIT DONE : REC *LSREC* CYC *LSCYC* SSEQ: *LSSS*

Erläuterung: Beim Start verarbeitet der Agent die Datei EELHTDS (Wiederherstellungsdatei für anstehende Übergaben) erneut, um die Jobs zu ermitteln, die erneut übergeben werden müssen, weil ihre Übergabe weiterhin ansteht.

Vor der erneuten Verarbeitung der Jobs werden die folgenden Informationen bereitgestellt:

- Die EELHTDS-Position, an der der nächste Satz geschrieben wird, angegeben durch die Satznummer *INREC* und den Schreibzyklus *INCYC*.
- Der EELHTDS-Bereich, der analysiert wird, angegeben durch:
 - Startposition: Satznummer *STREC*, Schreibzyklus *STCYC*
 - Endposition: Satznummer *CUREC*, Schreibzyklus *CUCYC*
- Die letzte vom Agenten ausgeführte Übergabe vor dem Neustart, angegeben durch Sequenznummer *LSSS* und zugehörigen Satz in EELHTDS mit Satznummer *LSREC* und Schreibzyklus *LSCYC*.

Systemaktion: Die Verarbeitung wird fortgesetzt.

EELHT37I

INFORMATION ABOUT RESTART PROCESSING OF EELHTREF DATA SET FOLLOWS:
HTREF NEXT TO WRITE : REC *INREC* CYC *INCYC*
HTREF START POSITION : REC *STREC* CYC *STCYC*
HTREF LAST POSITION : REC *LAREC* CYC *LACYC*
HTDS LAST SUBMIT STORED: REC *HTREC* CYC *HTCYC* SSEQ: *HTSS*

Erläuterung: Beim Start verarbeitet der Agent die Datei EELHTREF (Wiederherstellungsdatei für anstehende Anforderungen) erneut, um die Jobs zu ermitteln, die erneut verarbeitet werden müssen, weil sie weiterhin anstehen.

Vor der erneuten Verarbeitung der Jobs werden die folgenden Informationen bereitgestellt:

- Die EELHTDS-Position, an der der nächste Satz geschrieben wird, angegeben durch die Satznummer *INREC* und den Schreibzyklus *INCYC*.
- Der EELHTDS-Bereich, der analysiert wird, angegeben durch:
 - Startposition: Satznummer *STREC*, Schreibzyklus *STCYC*
 - Endposition: Satznummer *LAREC*, Schreibzyklus *LACYC*.
- Die letzte in EELHTDS aufgezeichnete, vom Agenten ausgeführte Übergabe

vor dem Neustart, angegeben durch Sequenznummer *HTSS* und zugehörigen Satz in EELHTDS mit Satznummer *HTREC* und Schreibzyklus *HTCYC*.

Systemaktion: Die Verarbeitung wird fortgesetzt.

EELHT38I

EELHTREF REPROCESSING RECOVERED FOLLOWING EVENT:

JOBALIA1

JOBALIA2

Erläuterung: Beim Start verarbeitet der Agent die Datei EELHTREF (Wiederherstellungsdatei für anstehende Anforderungen) erneut, um die Jobs zu ermitteln, die erneut übergeben werden müssen, weil sie weiterhin anstehen.

Diese Nachricht wird für jeden erneut verarbeiteten Job ausgegeben, der durch den Aliasnamen *JOBALIA1* und *JOBALIA2* identifiziert wird.

Systemaktion: Die Verarbeitung wird fortgesetzt.

EELHT42I

THE HTTP SERVER FINISHED THE SYNCHRONIZATION WITH THE SUBMIT TASK.

NUM SUBMISSION EVENTS HAVE BEEN REPROCESSED.

Erläuterung: Beim Start verarbeitet der Agent die Datei EELHTDS (Wiederherstellungsdatei für anstehende Übergaben) erneut, um die Jobs zu ermitteln, die erneut übergeben werden müssen, weil ihre Übergabe weiterhin ansteht. Während des Prozesses wird die Nachricht EELHT28I für jeden erneut übergebenen Job ausgegeben. Am Ende des Prozesses wird die Nachricht EELHT42I ausgegeben, um die Gesamtzahl der erneut verarbeiteten Ereignisse *NUM* mitzuteilen.

Beachten Sie, dass es geschehen kann, dass die Anzahl der ausgegebenen Nachrichten EELHT28I kleiner als *NUM* ist. Dies tritt ein, wenn unvollständige Ereignisse gefunden werden (wie durch die Nachricht EELHT35W gemeldet), die nicht übergeben werden können.

Systemaktion: Keine.

EELHT44I

THE HTTP SERVER STARTED THE EELHTREF DATA SET PROCESSING.

Erläuterung: Beim Start prüft der Agent die Datei EELHTREF (Wiederherstellungsdatei für anstehende Anforderungen), um anstehende Jobübergaben zu ermitteln, die verarbeitet werden müssen.

Systemaktion: Der Agent verarbeitet die Datei EELHTREF.

EELHT45I

THE HTTP SERVER ENDED THE EELHTREF DATA SET PROCESSING.

NUM SUBMISSION EVENTS HAVE BEEN REPROCESSED.

Erläuterung: Beim Start prüft der Agent die Datei EELHTREF (Wiederherstellungsdatei für anstehende Anforderungen), um anstehende Jobübergaben zu ermitteln, die verarbeitet werden müssen.

Systemaktion: Keine.

Komponentenversionen müssen zur vollständigen Nutzung der aktuellen Funktionalität gleichen Stand haben

Stellen Sie zur vollständigen Nutzung der neuesten Funktionen des Agenten für z/OS sicher, dass die Versionen aller Komponenten übereinstimmen.

Um die komplette Funktionalität, die der Agent für z/OS gegenwärtig zur Verfügung stellt, nutzen zu können, müssen die Version des Agenten und die Version von Dynamic Workload Broker, mit der er verbunden ist (sei es auf einem dynamischen Domänenmanager oder auf einem Masterdomänenmanager) den gleichen Stand aufweisen.

Insbesondere ist zu beachten, dass wenn der Agent für z/OS Version 8.6.0.2 mit einem dynamischen Domänenmanager oder einem Masterdomänenmanager der Version 8.6 oder 8.6.0.1 verbunden wird, keine Jobs nach Verweis definiert und keine Variablensubstitution in den JCLs verwendet werden kann.

Wenn Sie dies dennoch tun, werden die Jobs nach der Übergabe mit einem Fehler beendet und können in einigen Fällen unendlich lang blockieren, ohne einen Fehlercode auszugeben.

Vollständige Füllung des DB2-Transaktionsprotokolls stoppt Jobverarbeitung

Die Verarbeitung von Jobs für den Agenten für z/OS wird beendet, wenn das DB2-Transaktionsprotokoll vollständig gefüllt ist.

Wenn das DB2-Transaktionsprotokoll während der Ausführung eines Jobs vollständig gefüllt wird, wird die Ausführung des Jobs unterbrochen und die Verbindung zwischen dem Agenten und dem Dynamic Workload Broker gestoppt. Alle Jobs des Agenten für z/OS, die noch ausgeführt werden müssen, bleiben inaktiv. In dem Sonderfall, dass der übergebene Plan ausschließlich Jobs des Agenten für z/OS enthält, wird der Plan sogar ganz gestoppt.

Suchen Sie nach den folgenden Nachrichten, um sicherzugehen, dass ein vollständig gefülltes DB2-Transaktionsprotokoll das Problem ist:

In EELMLOG unter z/OS:

```
EELHT15E THE HTTP CLIENT FAILED TO PROCESS A REQUEST FOR BROKER
EELHT43I HTTP RESPONSE MESSAGE WITH CODE RDBMS_TRANSPORT_PROBLEM
```

In IBM Workload Scheduler

1. Aus BATCHMAN (tritt bei aktiven Jobs auf, wenn sich das Transaktionsprotokoll vollständig füllt):

```
jobname has failed with the error: An error occurred reading the
job from the job table
```

```
jobname has failed with the error: AWKJDB801E An internal error has
been found while accessing the database. The internal error message is:
"Not enough storage is available in the application heap to process"
```

```
AWSBHT032I Workstation broker-workstationname is now inactive,
no jobs will be scheduled.
```

2. Aus MAILMAN (meldet, dass der Agent und der Dynamic Workload Broker die Verbindung aufheben):

```
AWSBCV082I Workstation broker-workstationname, message: AWSDEB014I
Connection timed out
AWSBCV027I Unlinking from broker-workstationname
```

3. Aus BATCHMAN (nach Aufhebung der Verbindung von Dynamic Workload Broker und Stoppen der Jobübergabe an den Agenten):

```
AWSBDY103I Received command MY:UNLINK for run number 42 for
workstation broker-workstationname from workstation
workstationname
```

```
Workstation broker-workstationname State is being changed: UNSETTING:
LINKED=TCP AWSBHT032I Workstation broker-workstationname is now
inactive, no jobs will be scheduled.
```

4. In der Datei SystemOut.log:

```
AWSJDB801E An internal error has been found while accessing the
database.The internal error message is: "Not enough storage is
available in the application heap to process the statement..
SQLCODE=-954, SQLSTATE=57011, DRIVER=3.61.75".
```

In der Datei db2diag.log des DB2-Servers:

```
MESSAGE : ZRC=0x85100009=-2062548983=SQLP_NOSPACE
          "Log File has reached its saturation point"
          DIA8309C Die Protokolldatei war voll.

RETCODE : ZRC=0x8B0F0001=-1961951231=SQLO_NOMEM_APPH
          "No memory available in 'Application Heap'"
          DIA8301C Im Zwischenspeicher für Anwendungen
          ist kein freier Speicherbereich verfügbar.
```

Die vollständige Füllung des Transaktionsprotokolls des DB2-Servers ist auf die Tatsache zurückzuführen, dass die Protokollgröße infolge ihrer Standardeinstellungen nicht ausreicht. Zur künftigen Vermeidung einer vollständigen Füllung des Protokolls ändern Sie die DB2-Konfigurationseinstellungen mindestens auf die folgenden Werte:

```
Protokolldateigröße (4 KB)          (LOGFILSIZ) = 10000
Anzahl primärer Protokolldateien    (LOGPRIMARY) = 80
Anzahl sekundärer Protokolldateien  (LOGSECOND) = 40
```

Weitere Informationen finden Sie in der DB2-Dokumentation.

Datenbereiche

Stellt eine grafische Darstellung der Datenbereiche bereit, die vom Agenten für z/OS verwendet werden.

In den nächsten Abschnitten wird eine grafische Darstellung der Datenbereiche bereitgestellt, die vom Agenten für z/OS verwendet werden.

Die Datenbereiche sind in alphabetischer Reihenfolge nach dem Namen des Zuordnungsmakros aufgeführt. Die hier beschriebenen Datenbereiche sind nicht als Benutzerschnittstelle für Kunden gedacht; Kenntnisse zu ihrem Aufbau können jedoch für die Fehlerbehebung nützlich sein. Die Bereiche sind vom Produkt abhängig und können während des aktuellen Release jederzeit geändert werden, ohne dass diese Dokumentation entsprechend aktualisiert wird.

Datenbereichsübersicht

Jeder Datenbereich wird Feld für Feld beschrieben. Diese Feldbeschreibungen sind direkt aus dem Systemcode übernommen.

Für jedes Feld in dem Datenbereich enthält die Tabelle die folgenden Informationen:

Offsets

Die Adresse des Feldes sowohl in dezimaler als auch in hexadezimaler Schreibweise (Hexadezimaladresse in runden Klammern), relativ zum Anfang des Datenbereichs.

Typ Die Art der für dieses Feld definierten Programmdatei, z. B. CHARACTER, SIGNED, UNSIGNED.

Länge Die Größe des Feldes in Byte (dezimal).

Name Der Name des Feldes, des Bits oder der Maske.

Dem Bit- oder Maskennamen wird eine Beschreibung der Bitposition und der Bitwerte vorangestellt wie im Folgenden dargestellt:

1... Bezieht sich auf Bit 0.
11 Bezieht sich auf die Bits 6 und 7.
 ...1 Bezieht sich auf Bit 3.
 11.. 1111 Bezieht sich auf die Bits 0, 1, 4, 5, 6 und 7.

Beschreibung

Eine Beschreibung des Zwecks oder der Bedeutung des Feldes, des Bits oder der Maske.

Querweistabelle

Für jeden Datenbereich mit mehr als 40 Feldern enthält eine Querweistabelle die folgenden Informationen:

- Hexadezimaler Offset: Der hexadezimale Offset des Feldes vom Anfang des Datenbereichs (für Bits der hexadezimale Offset des Feldes, das das Bit enthält).
- Hexadezimalwert: Die Hexadezimalwerte werden nur für Bits aufgeführt. Der aufgeführte Hexadezimalwert gibt die Position des Bits in dem Feld an, das das Bit enthält.

DQE - Data Queue Element (Datenwarteschlangenelement)

Name : DCLDQE

Funktion:

Dieses Segment ordnet Warteschlangenelemente für verschiedene IBM Workload Scheduler-Warteschlangen (WS) zu. Die Zuordnung des Feldes DQEDATA ist vom Wert von DQETYPE abhängig. Werden Datenpuffer verwendet (angegeben durch dqebp^r ^= 0), werden diese von WS-Hinzufügeprogrammen immer in Subpool 2 zugeordnet und freigegeben, wenn sie von den Warteschlangenservern nicht mehr benötigt werden.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	152	dqe	data router q element	
0	(0)	CHARACTER	4	dqedesc	block descriptor, DQE	
4	(4)	CHARACTER	2	dqever	version number, 01	
6	(6)	BITSTRING	2	*	reserved flags	
8	(8)	CHARACTER	3	dqetype	data type	
11	(B)	BITSTRING	1	dqeflags	flags	
		1...		dqeflres	reset seqds	
		.1..		*	free	
		..1.		dqenckpt	do not chkpt this submit	
		...1		*	free	
	 1...		*	free	
	1..		*	free	
	1.		*	free	
	1		*	free on=used	
12	(C)	ADDRESS	4	*	free	

Offsets						
16	(10)	ADDRESS	4	dqebpnr	external data buffer ptr	
20	(14)	SIGNED	4	dqeblen	size of external buffer	
24	(18)	CHARACTER	8	dqedest	destination id	
32	(20)	SIGNED	4	dqermax	max # of recs per cyc in eds NOTE: the EELHTDS record number and write cycle are stored in EELEVDS header	
36	(24)	SIGNED	4	dqeevtr	used in SUTOP to pass ht ds rec	
40	(28)	SIGNED	4	dqeevtc	used in SUTOP to pass ht ds cyc	
44	(2C)	CHARACTER	100	dqedata	local data buffer	
144	(90)	SIGNED	4	dqeadder	Additional recs w buffer	
148	(94)	BITSTRING	1	*	free	
149	(95)	CHARACTER	1	*	reserved	
150	(96)	UNSIGNED	2	dqeevts	used in SUTOP to pass sseq	
152	(98)	CHARACTER		dqeend	end of dqe	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	100	dqeevt	event data buffer mapping	
0	(0)	SIGNED	2	dqennum	tw s node number	
2	(2)	SIGNED	2	dqerdrn	event reader number in node	
4	(4)	CHARACTER	8	dqepos	event ds position	
4	(4)	SIGNED	4	dqecyc	write cycle number	
8	(8)	SIGNED	4	dqerec	record number in cycle	
12	(C)	CHARACTER	8	*	free	
20	(14)	CHARACTER	80	dqeexr	exit record, see dcl exr	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	6	dqeclj	a DASD tracker needs	

Offsets

0	(0)	CHARACTER	5	dqecljob	clnjob prefix
5	(5)	CHARACTER	1	dqedscas	datastore class if JCC

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	20	dqerfw	rflw data
0	(0)	CHARACTER	8	dqerfwdest	destination name
8	(8)	CHARACTER	8	dqeedp	edp information
8	(8)	SIGNED	4	dqeedpwcy	write cycle number
12	(C)	SIGNED	4	dqeedprec	record # of last record
16	(10)	SIGNED	4	dqerfwnum	node number

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	48	dqerel	release job mapping
0	(0)	CHARACTER	8	dqerjbnm	job name
8	(8)	CHARACTER	8	dqerjbid	job number
16	(10)	CHARACTER	8	dqercnje	current nje node
24	(18)	CHARACTER	8	dqeronje	origin nje node
32	(20)	UNSIGNED	2	dqeaseq	submit sequence #
34	(22)	CHARACTER	4	dqeawsid	work station id
38	(26)	UNSIGNED	2	*	reserved
40	(28)	SIGNED	4	dqerojid	original job number
44	(2C)	CHARACTER	4	dqeassnam	controller subsystem name

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	100	dqesub	submit data mapping
0	(0)	CHARACTER	46	*	reserved - do not use
46	(2E)	CHARACTER	50	dqesubesp	DQESUBESP MUST BE MAXIMUM 50 CHARS TO BE FIT IN ESP DATA
46	(2E)	BITSTRING	1	dqesubFLA	DQE submit flags

Offsets

		1...		dqeERRO	dqe in error
		.1..		dqeUSED	dqe used
		..11 1111		*	free
47	(2F)	CHARACTER	5	*	free
52	(34)	CHARACTER	8	dqetso	Tso user id or blank
60	(3C)	CHARACTER	4	dqewsid	work station id
64	(40)	CHARACTER	4	dqessnam	controller subsystem name
68	(44)	CHARACTER	8	dqejobn	job/STC name
76	(4C)	CHARACTER	8	dqejid	job# of last subbed job
84	(54)	CHARACTER	1	dqeSubType	J, C, D, or O
85	(55)	CHARACTER	1	*	free
86	(56)	UNSIGNED	2	dqesseq	submit sseq (J1)
88	(58)	SIGNED	4	dqerecsub	record in HTDS
92	(5C)	SIGNED	4	dqecycsub	cycle in HTDS
96	(60)	CHARACTER	4	*	reserved - do not use

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	28	dqersseq	sseq# sync request (J0)
0	(0)	CHARACTER	4	dqerwsid	work station id
4	(4)	CHARACTER	8	dqecreat	request evt creation time
4	(4)	SIGNED	4	dqedate	date format (00yydddf)
8	(8)	SIGNED	4	dqetime	time format (secs*100)
12	(C)	CHARACTER	4	dqerssnam	controller subsystem name
16	(10)	UNSIGNED	1	*	free
17	(11)	BITSTRING	1	dqerflg1	flagbyte byte 1
		1...		dqerask	req for curr evds sseq#
		.1..		dqercold	cold start the ws
		..1.		dqerdlte	remove the ws from evds
		...1		dqerset	req to set evds to wseq#
	 1...		dqeinit	Initialization event

Offsets

	111		*	reserved
18	(12)	UNSIGNED	2	dqewseq	Actual WS submit sseq (J0)
20	(14)	CHARACTER	8	dqeJdest	dqedest

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimensi- on)	Beschreibung
0	(0)	STRUCTURE	18	dqecmini	initialize CM for oper
0	(0)	CHARACTER	8	dqecmjob	job/STC name
8	(8)	SIGNED	4	dqecmocc	occurrence number
12	(C)	SIGNED	4	dqecmopr	operation number
16	(10)	CHARACTER	2	dqecmrt	TASK ID requestor : EM = Event Manager ; AR = Automatic Recovery ; GS = General Service

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimensi- on)	Beschreibung
0	(0)	STRUCTURE	100	dqerfp	RODM request for parms
0	(0)	CHARACTER	8	dqerfpd	requestor destina- tion
8	(8)	CHARACTER	92	*	reserved

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimensi- on)	Beschreibung
0	(0)	STRUCTURE	100	dqersl	RODM subsystem lost
0	(0)	CHARACTER	8	dqersldn	requestor destina- tion
8	(8)	CHARACTER	4	dqerslssn	subsystem name
12	(C)	CHARACTER	88	*	reserved

Dynamischer kritischer Pfad:

DQEDATA mit Daten wird an Handler für kritischen Pfad gesendet, wenn eine MCP-Aktion an einem kritischen vorangegangenen Job ausgeführt wird und der Datenraum deshalb aktualisiert werden muss.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	100	dqecrt	Crit Path recalc	
0	(0)	CHARACTER	4	dqews	workstation name	
4	(4)	CHARACTER	1	dqestatus	operation status	
5	(5)	CHARACTER	1	dqeexstat	oper extended status	
6	(6)	BITSTRING	1	dqeflags2	flags	
		1...		dqenoped	noped operation	
		.1..		dqemanheld	oper manually held	
		..1.		dqemheldch	first dqe of a seq	
		...1		dqenopch	last dqe of a seq	
	 1...		dqetmdch	time dep chg	
	1..		dqetmdep	time dep	
	1.		dqeopiach	oper ia changed	
	1		*	free	
		7		(7)	BITSTRING	1
1...	dqeisfirst		first dqe of a seq			
.1..	dqeisfirst		add job to dataspace			
..1.	dqeislast		last dqe of a seq			
...1 1111	*		free			
8	(8)	CHARACTER	8	dqejobname	job name	
16	(10)	SIGNED	4	dqejobenix	job table entry index	
20	(14)	CHARACTER	1	dqejobprty	job priority	
21	(15)	CHARACTER	1	dqereqtype	MCP request type S - status change A - Add operation/dep D - Delete (oper, dep) M - Modify P - Critical indicat	
22	(16)	CHARACTER	2	*	free	
24	(18)	SIGNED	4	dqeindpred	pred j table entry ix	
28	(1C)	CHARACTER	1	dqeoldstat	operation old status	
29	(1D)	CHARACTER	1	dqecritind	critical indicator	
30	(1E)	CHARACTER	10	dqeinparr	input arrival time	
30	(1E)	CHARACTER	6	dqeinparrd	date	
36	(24)	CHARACTER	4	dqeinparrt	time	
40	(28)	CHARACTER	2	*	free	
42	(2A)	CHARACTER	10	dqedeadl	deadline	

Offsets

42	(2A)	CHARACTER	6	dqedeadld	date
48	(30)	CHARACTER	4	dqedeadlt	time
52	(34)	CHARACTER	2	*	free
54	(36)	CHARACTER	10	dqeastart	actual start time
54	(36)	CHARACTER	6	dqeastartd	date
60	(3C)	SIGNED	4	dqeastartt	time
64	(40)	CHARACTER	2	*	free
66	(42)	CHARACTER	10	dqeaend	actual end time
66	(42)	CHARACTER	6	dqeaendd	date
72	(48)	SIGNED	4	dqeaendt	time
76	(4C)	SIGNED	4	dqeduration	duration
80	(50)	SIGNED	4	dqeactdur	actual duration
84	(54)	CHARACTER	4	dqeoprkey	operation index
84	(54)	UNSIGNED	3	dqeoccidx	occ number
87	(57)	UNSIGNED	1	dqeopridx	oper number
88	(58)	CHARACTER	4	dqeerrcode	job error code
92	(5C)	CHARACTER	4	dqeprekey	pred oper index
92	(5C)	UNSIGNED	3	dqepoccidx	occ number
95	(5F)	UNSIGNED	1	dqepopridx	oper number
96	(60)	CHARACTER	4	*	free

Externer Puffer für DQECRT
Enthält WLM-Daten: Richtlinie und Serviceklasse

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	40	dqecrtBUF	DQE Type CRT
0	(0)	CHARACTER	1	dqewlmpol	WLM policy \$CRFA
1	(1)	CHARACTER	5	*	free
6	(6)	CHARACTER	8	dqewlmclass	WLM Service Class
14	(E)	CHARACTER	10	dqelstart	latest start
14	(E)	CHARACTER	6	dqelstartd	latest start date
20	(14)	SIGNED	4	dqelstartt	ltst start time
24	(18)	CHARACTER	16	dqeopIA	operation IA
24	(18)	CHARACTER	6	dqeopIAD	date
30	(1E)	CHARACTER	4	dqeopIAT	time
34	(22)	CHARACTER	6	*	free

Dynamischer kritischer Pfad:
DQEDATA mit Daten wird an Handler für kritischen Pfad gesendet, wenn eine Statusänderung (EM) stattfindet oder der Job verzögert bzw. mit langer Laufzeit aktiv ist (WA).

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	100	dqecr1	Crit Path recalc
0	(0)	CHARACTER	1	dqestat1	operation status
1	(1)	CHARACTER	1	dqeexstat1	oper extended status
2	(2)	CHARACTER	2	*	free
4	(4)	SIGNED	4	dqejobeni1	job table entry index
8	(8)	CHARACTER	1	dqereqtyp1	MCP request type S - status change L - Late R - Long Running
9	(9)	CHARACTER	1	dqeoldsta1	operation old status
10	(A)	BITSTRING	1	dqeflags4	flags
		1...		dqeurgch1	doa urgent queue chng
		.1..		dqedoaur1	doa urgent queue flag
		..1.		dqewlmpro1	WLM promotion flag
		...1 1111		*	free
11	(B)	CHARACTER	3	*	free
14	(E)	CHARACTER	10	dqeastar1	actual start time
14	(E)	CHARACTER	6	dqeastar1d	date
20	(14)	SIGNED	4	dqeastar1t	time
24	(18)	CHARACTER	2	*	free
26	(1A)	CHARACTER	10	dqeaen1	actual end time
26	(1A)	CHARACTER	6	dqeaen1d	date
32	(20)	SIGNED	4	dqeaen1t	time
36	(24)	SIGNED	4	dqeactdu1	actual duration
40	(28)	CHARACTER	4	dqeoprke1	operation index
40	(28)	UNSIGNED	3	dqeoccid1	occ number
43	(2B)	UNSIGNED	1	dqeoprid1	oper number
44	(2C)	CHARACTER	4	dqeerrcod1	job error code
48	(30)	CHARACTER	52	*	free

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung

Offsets						
0	(0)	STRUCTURE	100	dqearc		DQE Type TWS Data Sto D90C
0	(0)	CHARACTER	4	dqearctyp		Service Required: D90C LOG = get MVS JobLog D90C SLO = get Struct. Log D90A OPI = ask Oper_info D90A SDEL= start delete D90A of old entries D90A
common part						
4	(4)	CHARACTER	24	dqearcJRkey		Job Log retrieval key D90C
4	(4)	CHARACTER	8	dqearcJobId		Job Log Id D90C
12	(C)	CHARACTER	8	dqearcJobNa		Job Log Name D90C
20	(14)	SIGNED	4	dqearcRdrD		Job Start Rdr Date D90C
24	(18)	SIGNED	4	dqearcRdrT		Job Start Rdr Time D90C
28	(1C)	CHARACTER	8	dqearcdest		output DEST D90C
36	(24)	CHARACTER	16	dqearcADID		Application Name D90C
52	(34)	CHARACTER	10	dqearcIA		IA Date and Time D90C
52	(34)	CHARACTER	6	dqearcIADate		IA Date D90C
58	(3A)	CHARACTER	4	dqearcIATime		IA Time D90C
62	(3E)	CHARACTER	9	dqearcOPkey		Operation VSAM key D90C
62	(3E)	CHARACTER	8	dqearcOCC		OCC token D90C
70	(46)	UNSIGNED	1	dqearcOPR		Opr num D54A
71	(47)	CHARACTER	2	dqearccaller		EM=Event manager D90A WA=WorkStat. Analyzer D90A AR=Automatic Recover D90A GS=General Server D90A
OPI type only						
73	(49)	CHARACTER	1	dqearcfla1		flags (Ask Oper_Info)
		1... ..		ARC_PSUpst		1 = post PSU D90A
		.1.. ..		ARC_CleanUp		1 = Stand Alone CLNUP
		..1.		ARC_SL		1 = StepList required
		...1		ARC_DS		1 = DSList required
	 1...		ARC_ExpJCL		1 = use expanded JCL
	1..		ARC_BestStep		1 = start from BSTEP

Offsets

	1.		ARC_noask	1 = cp14nostr ON
	1		ARC_AskSimGDG	1 = GDG sim required
74	(4A)	CHARACTER	8	*	free
82	(52)	CHARACTER	8	dqearcUSER	original job userid
				Clean Up only	
90	(5A)	CHARACTER	8	dqearcEXdest	execution destination
				AR restart only	
98	(62)	UNSIGNED	1	dqearcopiRet	retry counter
99	(63)	UNSIGNED	1	dqearcARste	AR restart step 01A (currently not used)

Nur AR-Neustart
Struktur in DQEARC-Puffer zugeordnet:

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	16	dqearcBUF	DQE Type TWS Data Sto
0	(0)	CHARACTER	8	dqearcSTEPN	Stepname (AR)
8	(8)	CHARACTER	8	dqearcPSTEPN	Proc Step Name (AR)

Anmerkung: Für den SDEL-Typ dqearc sind alle Informationen in dem Puffer enthalten, auf den dqebptr verweist: dqebptr-> CP16-Satzaufbau

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	100	dqePSU	DQE Type TWS Data Sto
0	(0)	CHARACTER	4	dqePSUtyp	Service Required: APL = Apply logic A CLN = Stand Alone CLN REM = resume suspended DEL = delete request

----- D90A - common part D90A -----
D90A

4	(4)	CHARACTER	45	dqePSUcommon	D90A
4	(4)	CHARACTER	2	dqePSUcaller	FL=Fetch JobLog D90A GS=General Server D90A EM=Event manager D90A

Offsets					
6	(6)	CHARACTER	8	dqePSUJobNa	Job Name D90A
14	(E)	CHARACTER	9	dqePSUOPkey	Operation VSAM key D90A
14	(E)	CHARACTER	8	dqePSUOCC	Occ token D90A
22	(16)	UNSIGNED	1	dqePSUOPR	Opr num D90A
23	(17)	CHARACTER	16	dqePSUADID	Application Name D90A
39	(27)	CHARACTER	10	dqePSUIA	IA Date and Time D90A
39	(27)	CHARACTER	6	dqePSUIADate	IA Date D90A
45	(2D)	CHARACTER	4	dqePSUIATime	IA Time D90A
----- D90A - GS only D90A -----					
49	(31)	CHARACTER	8	dqePSUuser	TSO userid D90A
57	(39)	CHARACTER	4	dqePSUtoken	GS request token D90A
----- D90A - FL for AR only D90A -----					
61	(3D)	CHARACTER	8	dqeStepname	AR restart step D90A
69	(45)	CHARACTER	8	dqePStepName	AR restart proc step D90A
----- D90A - CLN type only -D90A -----					
77	(4D)	CHARACTER	8	dqePSUexdest	execution DEST D90A
----- D90A - APL type only -D90A -----					
85	(55)	CHARACTER	1	dqePSUfla1	D90A
		1... ..		PSU_ExpJCL	1 = use expanded JCL D90A
		.1.. ..		PSU_Suspend	1 = suspend DQE D90A
		..1.		PSU_Operinfo	1 = SL from buffer D90A
		...1		PSU_UserSys	1 = add usersys DD
	 1...		PSU_SimGDG	1 = simulate GDG
	1..		PSU_Root	1 = GDG root list
	1.		PSU_BestStep	1 = start from BSTEP
	1		PSU_AskSimGDG	1 = req simulation
		2 char(2), free 2 dqePSULen,			
86	(56)	UNSIGNED	2	PSU_SLlen	SL length
88	(58)	UNSIGNED	3	PSU_DSlen	DL length
91	(5B)	UNSIGNED	3	PSU_GDGlen	Sim. GDG info
94	(5E)	UNSIGNED	2	PSU_Rootlen	Root len
96	(60)	SIGNED	4	PSU_JCLlen	JCL length D90A

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	80	dqeWLM	sub WLM reset request D63A	
0	(0)	CHARACTER	8	dqetsoW	Tso user id or blank D63A	
8	(8)	CHARACTER	8	dqejobnW	job/STC name D63A	
16	(10)	CHARACTER	8	dqeSVCnW	WLM Hi perf service class	
24	(18)	CHARACTER	4	dqewsidW	work station id D63A	
28	(1C)	CHARACTER	16	dqeadidW	application ID D63A	
44	(2C)	CHARACTER	10	dqeociaW	occurrence input arrival	
44	(2C)	CHARACTER	6	dqeociaD	occurrence IA date D63A	
50	(32)	CHARACTER	4	dqeociaT	occurrence IA time D63A	
54	(36)	SIGNED	2	dqeopnumW	operation number D63A	
56	(38)	CHARACTER	4	dqessnamW	controller subsystem name	
60	(3C)	UNSIGNED	2	*	free	
		2 dqejidW char(8) , job of last subbed job 2 char(2) , reserved D63D				
62	(3E)	SIGNED	2	dqeasidW	job asid D63A	
		2 dqetoken_fullW , operation token D63d 3 dqetoken_preW char(4) , operation token prefix 3 dqetokenW unsigned bin(32), operation token 2 dqeaccmpW ptr(31) , acc meth parms Q0A 2 dqenetidW char(8) , APPC netid (final dest) 2 dqenetLUW char(8) ; APPC netLU (final dest)				
64	(40)	SIGNED	4	dqeretcW	WLM promot request RC	
68	(44)	SIGNED	2	dqersncW	WLM promot request RSN	
70	(46)	CHARACTER	10	*	reserved D63A	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	8	dqeTcpIp	tcpip evt mapping	
0	(0)	ADDRESS	4	dqeSocketIdPtr		
4	(4)	CHARACTER	4	dqeSocketDomain		

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	

Offsets						
0	(0)	STRUCTURE	18	dqeConfFile	Conf File mapping	
0	(0)	CHARACTER	1	dqeRequestType		
1	(1)	CHARACTER	3	*		
4	(4)	CHARACTER	4	dqeRequestCRC		
8	(8)	CHARACTER	8	dqeCRCOwnerDest		
16	(10)	SIGNED	2	dqeDtbDestIndex		

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	44	dqeHTTAction	HTTP Joblog retrieval Kill Query job status	
0	(0)	CHARACTER	8	dqeHTocctoken	HTTP occurrence token	
8	(8)	CHARACTER	8	dqeHTsubtoken	HTTP submission token	
16	(10)	CHARACTER	8	dqeHTjobname	HTTP job name	
24	(18)	CHARACTER	4	dqeHTwsname	HTTP ws name	
28	(1C)	CHARACTER	4	dqeHTssname	HTTP subsystem name	
32	(20)	SIGNED	2	dqeHTopnum	HTTP operation number	
34	(22)	SIGNED	2	*	reserved	
36	(24)	CHARACTER	8	dqeHTuser	HTTP joblog req user	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	100	dqeHTnotify	HTTP notify	
0	(0)	CHARACTER	56	dqeHNalias	alias	
56	(38)	CHARACTER	8	dqeHNjobid	job number	
64	(40)	CHARACTER	4	dqeHNerrc	error code (stat E)	
68	(44)	CHARACTER	6	dqeHNstartd	start date	
74	(4A)	CHARACTER	6	dqeHNendd	end date	
80	(50)	SIGNED	4	dqeHNstartt	start time	
84	(54)	SIGNED	4	dqeHNendt	end time	
88	(58)	CHARACTER	8	dqeHNxdtoken	XD99 key	
96	(60)	CHARACTER	1	dqeHNstatus	status (S/C/E)	
97	(61)	CHARACTER	3	*	free	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	100	dqeHTinfo	HTTP bind info/failed	
0	(0)	CHARACTER	56	dqeHlalias	alias	
56	(38)	CHARACTER	8	dqeHljobname	matched jobname	
64	(40)	CHARACTER	10	dqeHlia	matched ia	
64	(40)	CHARACTER	6	dqeHliad	..IA date	
70	(46)	CHARACTER	4	dqeHliat	..IA time	
74	(4A)	CHARACTER	1	dqeHltype	I=info, F=failed	
75	(4B)	CHARACTER	1	*	free	
76	(4C)	CHARACTER	8	dqeHlxdtoken	XD99 key	
84	(54)	CHARACTER	4	dqeHlwsname	matched ws name	
88	(58)	CHARACTER	12	*	free	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	100	dqeHTsubscr	HTTP subscribe	
0	(0)	CHARACTER	35	dqeHSrkeyz	remote job key	
0	(0)	CHARACTER	16	dqeHSadid	..remote adid/ jsname (D/Z)	
16	(10)	CHARACTER	16	dqeHSjsws	..remote jsws (D)	
32	(20)	CHARACTER	3	dqeHSopno	..remote opno (Z)	
35	(23)	CHARACTER	1	dqeHSflags	flags	
		1... ..		dqeHSzos	..ON: remote eng type is Z	
		.111 1111		*	..free	
36	(24)	CHARACTER	2	*	free	
38	(26)	CHARACTER	10	dqeHSia	IA for match	
38	(26)	CHARACTER	6	dqeHSiad	..IA date	
44	(2C)	CHARACTER	4	dqeHSiat	..IA time	
48	(30)	CHARACTER	52	dqeHSalias	data for alias	
48	(30)	CHARACTER	8	dqeHSocctoken	..occurence token	
56	(38)	CHARACTER	8	dqeHSsubtoken	..submission to- ken	
64	(40)	CHARACTER	8	dqeHSjobname	..job name	
72	(48)	CHARACTER	4	dqeHSwsname	..ws name	
76	(4C)	CHARACTER	4	dqeHSsname	..subsystem name	
80	(50)	SIGNED	2	dqeHSopnum	..operation number	
82	(52)	CHARACTER	18	*	free	

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	40	dqeHSbuff	DQE Type HTS
0	(0)	CHARACTER	40	dqeHSrjobnm	remote jobname

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	24	dqeJKJes	
0	(0)	CHARACTER	8	dqeJKJJobName	
8	(8)	CHARACTER	8	dqeJKJJobId	
16	(10)	SIGNED	4	dqeJKJRdrDate	
20	(14)	SIGNED	4	dqeJKJRdrTime	

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	56	dqeJKBroker	
0	(0)	CHARACTER	56	dqeJKBJobId	

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	8	dqeJAlias	
0	(0)	SIGNED	4	dqeJALength	
4	(4)	ADDRESS	4	dqeJApr	

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
ARC_AskSimGDG	49	01	3
ARC_BestStep	49	04	3
ARC_CleanUp	49	40	3
ARC_DS	49	10	3
ARC_ExpJCL	49	08	3
ARC_noask	49	02	3
ARC_PSUpost	49	80	3
ARC_SL	49	20	3
dqe	0		1
dqeactdur	50		2
dqeactdu1	24		2
dqeadder	90		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
dqeaddjob	7	40	3
dqeaidW	1C		2
dqeaend	42		2
dqeaendd	42		3
dqeaendt	48		3
dqeaen1	1A		2
dqeaen1d	1A		3
dqeaen1t	20		3
dqearc	0		1
dqearcADID	24		2
dqearcARste	63		2
dqearcBUF	0		1
dqearccaller	47		2
dqearcdest	1C		2
dqearcEXdest	5A		2
dqearcfla1	49		2
dqearcIA	34		2
dqearcIADate	34		3
dqearcIATime	3A		3
dqearcJobId	4		3
dqearcJobNa	C		3
dqearcJRkey	4		2
dqearcOCC	3E		3
dqearcopiRet	62		2
dqearcOPkey	3E		2
dqearcOPR	46		3
dqearcPSTEPN	8		2
dqearcRdrD	14		3
dqearcRdrT	18		3
dqearcSTEPN	0		2
dqearctyp	0		2
dqearcUSER	52		2
dqeaseq	20		2
dqeasidW	3E		2
dqeassnam	2C		2
dqeastart	36		2
dqeastartd	36		3
dqeastartt	3C		3
dqeastar1	E		2
dqeastar1d	E		3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
dqeastar1t	14		3
dqeawsid	22		2
dqeblen	14		2
dqebptr	10		2
dqeclj	0		1
dqecljob	0		2
dqecmini	0		1
dqecmjob	0		2
dqecmocc	8		2
dqecmopr	C		2
dqecmrt	10		2
dqeConfFile	0		1
dqeCRCOwnerDest	8		2
dqecreat	4		2
dqecritind	1D		2
dqecrt	0		1
dqecrtBUF	0		1
dqecr1	0		1
dqecyc	4		3
dqecycsub	5C		3
dqedata	2C		2
dqedate	4		3
dqedeadl	2A		2
dqedeadld	2A		3
dqedeadlt	30		3
dqedesc	0		2
dqedest	18		2
dqedoaur1	A	40	3
dqedsc1as	5		2
dqeDtbDestIndex	10		2
dqeduration	4C		2
dqeedp	8		2
dqeedprec	C		3
dqeedpwcy	8		3
dqeend	98		2
dqeerrcode	58		2
dqeerrcod1	2C		2
dqeERRO	2E	80	4
dqeevt	0		1
dqeevtc	28		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
dqeevtr	24		2
dqeevts	96		2
dqeexr	14		2
dqeexstat	5		2
dqeexstat1	1		2
dqeeflags	B		2
dqeeflags2	6		2
dqeeflags3	7		2
dqeeflags4	A		2
dqeeflres	B	80	3
dqeHlalias	0		2
dqeHlia	40		2
dqeHliad	40		3
dqeHliat	46		3
dqeHljobname	38		2
dqeHltype	4A		2
dqeHlwsname	54		2
dqeHlxdtoken	4C		2
dqeHNalias	0		2
dqeHNendd	4A		2
dqeHNendt	54		2
dqeHNerrc	40		2
dqeHNjobid	38		2
dqeHNstartd	44		2
dqeHNstartt	50		2
dqeHNstatus	60		2
dqeHNxdtoken	58		2
dqeHSadid	0		3
dqeHSalias	30		2
dqeHSbuff	0		1
dqeHSflags	23		2
dqeHSia	26		2
dqeHSiad	26		3
dqeHSiat	2C		3
dqeHSjobname	40		3
dqeHSjsws	10		3
dqeHSocctoken	30		3
dqeHSopno	20		3
dqeHSopnum	50		3
dqeHSrjobnm	0		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
dqeHSrkeyz	0		2
dqeHSsname	4C		3
dqeHSsubtoken	38		3
dqeHSwsname	48		3
dqeHSzos	23	80	3
dqeHTinfo	0		1
dqeHTjobname	10		2
dqeHTnotify	0		1
dqeHTocctoken	0		2
dqeHTopnum	20		2
dqeHTssname	1C		2
dqeHTsubscr	0		1
dqeHTsubtoken	8		2
dqeHTTAction	0		1
dqeHTuser	24		2
dqeHTwsname	18		2
dqeindpred	18		2
dqeinit	11	08	3
dqeinparr	1E		2
dqeinparrd	1E		3
dqeinparrt	24		3
dqeisfirst	7	80	3
dqeislast	7	20	3
dqeJALength	0		2
dqeJAlias	0		1
dqeJAptr	4		2
dqeJdest	14		2
dqeJid	4C		3
dqeJKBJobId	0		2
dqeJKBroker	0		1
dqeJKJes	0		1
dqeJKJJobId	8		2
dqeJKJJobName	0		2
dqeJKJRdrDate	10		2
dqeJKJRdrTime	14		2
dqejobenix	10		2
dqejobeni1	4		2
dqejobn	44		3
dqejobname	8		2
dqejobnW	8		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
dqejobprty	14		2
dqelstart	E		2
dqelstartd	E		3
dqelstartt	14		3
dqemanheld	6	40	3
dqemheldch	6	20	3
dqenckpt	B	20	3
dqennum	0		2
dqenopch	6	10	3
dqenoped	6	80	3
dqeoccidx	54		3
dqeoccid1	28		3
dqeociaD	2C		3
dqeociaT	32		3
dqeociaW	2C		2
dqeoldstat	1C		2
dqeoldsta1	9		2
dqeopIA	18		2
dqeopiach	6	02	3
dqeopIAD	18		3
dqeopIAT	1E		3
dqeopnumW	36		2
dqeopridx	57		3
dqeoprid1	2B		3
dqeoprkey	54		2
dqeoprke1	28		2
dqepoccidx	5C		3
dqepopridx	5F		3
dqepos	4		2
dqeprekey	5C		2
dqePStepName	45		2
dqePSU	0		1
dqePSUADID	17		3
dqePSUcaller	4		3
dqePSUcommon	4		2
dqePSUexdest	4D		2
dqePSUfla1	55		2
dqePSUIA	27		3
dqePSUIADate	27		4
dqePSUIATime	2D		4

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
dqePSUJobNa	6		3
dqePSUOCC	E		4
dqePSUOPkey	E		3
dqePSUOPR	16		4
dqePSUtoken	39		2
dqePSUtyp	0		2
dqePSUuser	31		2
dqerask	11	80	3
dqercnje	10		2
dqercold	11	40	3
dqerdlte	11	20	3
dqerdrn	2		2
dqerec	8		3
dqerecsub	58		3
dqerel	0		1
dqereqtype	15		2
dqereqtyp1	8		2
dqeRequestCRC	4		2
dqeRequestType	0		2
dqeretcW	40		2
dqerflg1	11		2
dqerfp	0		1
dqerfpd	0		2
dqerfw	0		1
dqerfwdest	0		2
dqerfwnum	10		2
dqerjbid	8		2
dqerjbnm	0		2
dqermax	20		2
dqerojid	28		2
dqeronje	18		2
dqerset	11	10	3
dqersl	0		1
dqersldn	0		2
dqerslssn	8		2
dqersncW	44		2
dqersseq	0		1
dqerssnam	C		2
dqerwsid	0		2
dqeSocketDomain	4		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
dqeSocketIdPtr	0		2
dqesseq	56		3
dqessnam	40		3
dqessnamW	38		2
dqestatus	4		2
dqestatu1	0		2
dqeStepname	3D		2
dqesub	0		1
dqesubesp	2E		2
dqesubFLA	2E		3
dqeSubType	54		3
dqeSVCnW	10		2
dqeTcpIp	0		1
dqetime	8		3
dqetmdch	6	08	3
dqetmdep	6	04	3
dqetso	34		3
dqetsoW	0		2
dqetype	8		2
dqeurgch1	A	80	3
dqeUSED	2E	40	4
dqever	4		2
dqeWLM	0		1
dqewlclass	6		2
dqewlmpol	0		2
dqewlmpol1	A	20	3
dqews	0		2
dqewseq	12		2
dqewsid	3C		3
dqewsidW	18		2
PSU_AskSimGDG	55	01	3
PSU_BestStep	55	02	3
PSU_DSlen	58		2
PSU_ExpJCL	55	80	3
PSU_GDGlen	5B		2
PSU_JCLlen	60		2
PSU_Operinfo	55	20	3
PSU_Root	55	04	3
PSU_Rootlen	5E		2
PSU_SimGDG	55	08	3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
PSU_SLen	56		2
PSU_Suspend	55	40	3
PSU_UserSys	55	10	3

ESP - Übersicht der Ereignisdatensätze in HTDS

Name : DCLESP

Funktion:

Ermöglicht die Fortsetzung von Ereignissen, die über die Ereignisschreiberwarteschlange und die Ereignisdatei an den Datenrouter gesendet werden. Diese Übersicht sollte nur verwendet werden, wenn mehr als 80 Byte wie oben beschrieben weitergeleitet werden sollen. ESP wird in DQE umgesetzt, wobei ESP_dqetype als DQE-Typ, die Daten im Typ F ESP als interner DQE-Puffer und die übrigen ESP-Daten als externer DQE-Puffer verwendet werden. Der Subpool der zugeordneten externen Puffer sollte in ESP_sp angegeben werden. EW führt freemain für diesen Bereich aus. Nullen ergeben SP0.

Verwenden Sie nur Speicher mit Schlüssel 0.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	80	ESP	ESP head or continuation
0	(0)	CHARACTER	1	ESP_exrtyp	event type = N
1	(1)	CHARACTER	1	ESP_type	type of ESP F = first / head ESP N = continued ESP
2	(2)	CHARACTER	1	ESP_syst	contained event type ! blank === above MUST match DCLEXR ==
3	(3)	CHARACTER	3	ESP_dqetype	actual contained event type
6	(6)	SIGNED	2	ESP_seqF	ESP F-type sequence number
8	(8)	SIGNED	4	ESP_seqN	ESP N-type sequence number
12	(C)	SIGNED	4	ESP_seqT	ESP seq totals
16	(10)	ADDRESS	4	ESP_extptr	address of data buffer ! NULL
20	(14)	SIGNED	4	ESP_datasize	Total size in buffer when ESP on WRTQ, and for F-type. Data size for N-types.

Offsets					
24	(18)	UNSIGNED	1	ESP_sp	subpool number for ext buffer
25	(19)	CHARACTER	1	ESP_flags	a flag byte
		1... ..		ESP_app	APP extension buffer
		.111 1111		*	reserved
26	(1A)	CHARACTER	50	ESP_data	actual event data
76	(4C)	CHARACTER	4	ESP_id	event identification (offs 76)
80	(50)	CHARACTER		ESP_xdata0	eXtended data portion plachold

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
ESP	0		1
ESP_app	19	80	3
ESP_data	1A		2
ESP_datasize	14		2
ESP_dqetype	3		2
ESP_exrtyp	0		2
ESP_extptr	10		2
ESP_flags	19		2
ESP_id	4C		2
ESP_seqF	6		2
ESP_seqN	8		2
ESP_seqT	C		2
ESP_sp	18		2
ESP_syst	2		2
ESP_type	1		2
ESP_xdata0	50		2

EVT - Übersicht über den Datensatzaufbau in Ereignisdateien

Name : DCLEVT

Funktion:

Dieses Segment beschreibt den Datensatzaufbau in den Ereignisdateien.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	820	evt	event record

Offsets					
0	(0)	CHARACTER	20	evtkey	event record key
0	(0)	SIGNED	4	evtrec	for header = 0 rec number for non-header
4	(4)	SIGNED	4	evtlrrec	latest record number = 0 for non-header
8	(8)	SIGNED	4	evtlrcyc	latest cycle for header
8	(8)	SIGNED	4	evtcyc	write cycle for non-header
12	(C)	SIGNED	4	evtmax	max rclds excl hdr = 0 for non-header
16	(10)	SIGNED	4	evttrcap	track capacity in rclds = 0 for non-header
20	(14)	CHARACTER	80	evtexr (10)	exit record (see dclxpr)
20	(14)	CHARACTER	80	evtsur	not used
20	(14)	CHARACTER	80	evtrel	not used

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	*	*	evds header record data
0	(0)	CHARACTER	16	evtckpte (*)	checkpoint entry
0	(0)	BITSTRING	1	*	free
1	(1)	UNSIGNED	1	*	free
2	(2)	CHARACTER	4	evtwsnm	workstation name
6	(6)	UNSIGNED	2	evtsseq	current submit seq
8	(8)	SIGNED	4	evtchkrec	latest submit record num
12	(C)	SIGNED	4	evtchkcyc	latest submit cycle num

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
evt	0		1
evtchkcyc	C		3
evtchkrec	8		3
evtckpte	0		2
evtcyc	8		4

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
evtexr	14		2
evtkey	0		2
evtlrcyc	8		3
evtlrrec	4		3
evtmax	C		3
evtrec	0		3
evtrel	14		4
evtsseq	6		3
evtsur	14		3
evtrcap	10		3
evtwsnm	2		3

EXI - Ix-Ereignisdefinition

Name : DCLEXI

Funktion:

Dieses Segment deklariert ein Initialisierungsereignis. Dieses Ereignis enthält Informationen zu einer eingeleiteten Operation. Initialisierungsereignisse werden von der Übergabetask erstellt und der Ereignisschreiberwarteschlange hinzugefügt. Beachten Sie, dass der Anfang von exi als exr zugeordnet werden muss.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	80	exi	initialization event
0	(0)	CHARACTER	1	exievtyp	event type = I
1	(1)	CHARACTER	3	exitype	type of initiated process J0 = submit seq event J1 = jcl submit J2 = jcl started task J3 = jcl for clean up D90A J4 = jcl submit SE failure \$CCUA WTO= WTO message REL= job release OSI= OSI type submit WLM= WLM job promotion
4	(4)	CHARACTER	4	exiwsid	work station id
8	(8)	CHARACTER	16	*	exiadid
8	(8)	CHARACTER	16	exiJ4schenv	Se name (J4)

Offsets						
8	(8)	CHARACTER	8	exircr		Request create time (J0)
8	(8)	SIGNED	4	exircrd		date (format: 00YYDDDF)
12	(C)	SIGNED	4	exircrct		time (format: secs*100)
16	(10)	CHARACTER	8	exiocctok		
16	(10)	UNSIGNED	2	exiwseq		submit seq from ws (IJ0)
24	(18)	CHARACTER	10	*		
24	(18)	SIGNED	4	exierec		current evds rec (IJ0)
28	(1C)	SIGNED	4	exiecy		current evds cyc (IJ0)
32	(20)	SIGNED	2	*		
34	(22)	SIGNED	2	exiopnum		operation number
36	(24)	SIGNED	4	exiWLMrc		WLM promotion request RC
36	(24)	BITSTRING	4	exiflags		flag area P10A
36	(24)	BITSTRING	1	exiflag1		flag byte 1 P10C
		1... ..		exifail		init of process failed P10C
		.1.. ..		exisfail		submit fail
		..1.		exinoedp		no edp updates for this ev31CLVA
		...1		exiWLMah		on= job already Hi perfo D52A
	 1...		exiSEund		on=SE not defined
	1..		exiSEnoAva		on=SE not available
	1.		exiSEnoJPL		on=SE not avail at JPLEX 31Clevel only
	1		exiinit		J0 related to initialization
37	(25)	BITSTRING	1	exiflag2		flag byte 2 P10C
		1... ..		exi2fail		init of process failed P10C
		.1.. ..		exi2plte		possible late I event P10C
		..1.		exi2bad		the STC funct not active P10C
		...1		exi2njss		not JES managed subsyst P10C
	 1...		exi2fjes		STC fail JES input servic31CP10C

Offsets

	1..		exi2ncf	NCF destination not valid31CP10C
	11		*	reserved P10C
38	(26)	BITSTRING	1	exiflag3	flag byte 3 P10C
		1...		exi3oos	submit out-of-sequence
		.1..		exi3nckp	checkpoint not active
		..1.		exi31st	1st time switch
		...1		exi3misr	seq request mismatch
	 1...		exi3allok	submit seq WA vs. SU okay
	1..		*	reserved
	1.		exi3badj0	bad J0 event (or cp)
	1		exi3rdop	error read cp oper
39	(27)	BITSTRING	1	exiflag4	flag byte 4 P10C
		1...		exi4rdjn	error read cp jnt
		.1..		exi4nsub	sub04 is blank!
		..1.		exi4job	jes04 is blank!
		...1		*	reserved
	 1...		exi4nsop	cplsubop is of..!?!?
	11.		*	reserved
	1		exi4succ	successful submit
40	(28)	CHARACTER	8	exijobn	job/STC name
40	(28)	UNSIGNED	2	exiwork	current catchup value(J0)
42	(2A)	UNSIGNED	2	*	free
44	(2C)	UNSIGNED	2	exiedoa	current doa value (J0)
46	(2E)	UNSIGNED	2	exijasid	job/STC asid D52C9TA
48	(30)	CHARACTER	8	exijobid	job/STC number (J1,J2,J3)
48	(30)	CHARACTER	8	exiWLMClS	WLM class (WLM only)
48	(30)	CHARACTER	8	exiJ4dest	tracker dest (J4)
48	(30)	CHARACTER	1	exievlog	reserved (WLM) 32C
49	(31)	CHARACTER	7	*	reserved (WLM)
56	(38)	SIGNED	2	exigmtof	gmt offset in minutes

Offsets					
58	(3A)	SIGNED	2	exiWLMrsn	WLM promotion request rsn code31A
58	(3A)	CHARACTER	1	exiVirt	Virtual WS support: Y/N
59	(3B)	CHARACTER	1	*	
60	(3C)	CHARACTER	8	exicreat	event record creation time word bndry
60	(3C)	SIGNED	4	exidate	date format (00yydddf)
64	(40)	SIGNED	4	exitime	time format (secs*100)
68	(44)	CHARACTER	4	exissnm	DQE originating TWS subsys name
72	(48)	UNSIGNED	2	exieseq	current evds seq (IJ0)
74	(4A)	UNSIGNED	2	*	free
76	(4C)	CHARACTER	4	exiid	event id EELx

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exi	0		1
exicreat	3C		2
exidate	3C		3
exiecyd	1C		3
exiedoa	2C		3
exierec	18		3
exieseq	48		2
exievlog	30		5
exievtyp	0		2
exifail	24	80	5
exiflags	24		3
exiflag1	24		4
exiflag2	25		4
exiflag3	26		4
exiflag4	27		4
exigmtof	38		2
exiid	4C		2
exiinit	24	01	5
exijasid	2E		3
exijobid	30		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exijobn	28		2
exij4dest	30		4
exij4schenv	8		3
exinoedp	24	20	5
exiocctok	10		4
exiopnum	22		2
exircr	8		4
exircr	8		5
exircr	C		5
exiSEnoAva	24	04	5
exiSEnoJPL	24	02	5
exiSEund	24	08	5
exisfail	24	40	5
exissnm	44		2
exitime	40		3
exitype	1		2
exiVirt	3A		3
exiWLMah	24	10	5
exiWLMClS	30		3
exiWLMrc	24		2
exiWLMrsn	3A		2
exiwork	28		3
exiwseq	10		5
exiwsid	4		2
exi2bad	25	20	5
exi2fail	25	80	5
exi2fjes	25	08	5
exi2ncf	25	04	5
exi2njss	25	10	5
exi2plte	25	40	5
exi3allok	26	08	5
exi3badj0	26	02	5
exi3misr	26	10	5
exi3nckp	26	40	5
exi3oos	26	80	5
exi3rdop	26	01	5
exi31st	26	20	5
exi4job	27	20	5
exi4nsop	27	08	5
exi4nsub	27	40	5

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exi4rdjn	27	80	5
exi4succ	27	01	5

EXK - Kx-Ereignisdefinition

Name : DCLEXX

Funktion:

Dieses Segment deklariert ein Initialisierungsereignis. Dieses Ereignis enthält Informationen zu einer eingeleiteten Operation. Initialisierungsereignisse werden von der Übergabetask erstellt und der Ereignisschreiberwarteschlange hinzugefügt. Beachten Sie, dass der Anfang von exk als exr zugeordnet werden muss.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	80	exk	initialization event
0	(0)	CHARACTER	1	exkevtyp	event type = K
1	(1)	CHARACTER	2	exktype	type of initiated process J1= jcl submit
3	(3)	CHARACTER	1	*	free
4	(4)	SIGNED	4	exkJKJJobIdN	JES Job Id (only number)
8	(8)	CHARACTER	36	exkJKBJobId	Broker Job Key
44	(2C)	CHARACTER	8	exkJobname	Jobname
52	(34)	SIGNED	4	exkdate	00YYDDDF exkdate
56	(38)	SIGNED	4	exktime	time format (secs*100)exktime
60	(3C)	CHARACTER	4	exkssname	origi subsys
64	(40)	BITSTRING	2	exkflags	
64	(40)	BITSTRING	1	exkflags1	
		1...		exkfail	init of process failed
		.1..		exkSEnoAva	on=SE not available
		..1.		exkSEnoJPL	on=SE not avail at JPLEX
		...1		exkSEund	on=SE not defined
	 1...		exk2fail	init of process failed
	1..		exk2plte	possible late I event

Offsets

	1.		exk2bad	the STC funct not active
	1		exk2njss	not JES managed subsystem
65	(41)	BITSTRING	1	exkflags2	
		1...		exk2fjes	STC fail JES input servic
		.1..		exk3nckp	checkpoint not active
		..1.		exk3oos	submit out-of-sequence
		...1 1111		*	free
66	(42)	BITSTRING	1	exkreason	free
		1...		exkJCLRopen	open joblib failure
		.1..		exkJCLRdyn	dynalloc failure
		..1.		exkJCLRjnMISS	missing jobname
		...1		exkJCLRjnINVA	invalid jobname
	 1...		exkJCLRstor	storage problems
	1..		exkOJCV	variable sub error
	1.		exknojCL	JCL not found
	1		*	free
67	(43)	BITSTRING	1	*	free
68	(44)	SIGNED	4	*	free
72	(48)	SIGNED	2	exkgmtof	gmt offset in minutes
74	(4A)	UNSIGNED	2	exksseq	current submit seq
76	(4C)	CHARACTER	4	exkid	event id EELx

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exk	0		1
exkdate	34		2
exkevtyp	0		2
exkfail	40	80	4
exkflags	40		2
exkflags1	40		3
exkflags2	41		3
exkgmtof	48		2
exkid	4C		2
exkJCLRdyn	42	40	3
exkJCLRjnINVA	42	10	3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exkJCLRjnMISS	42	20	3
exkJCLRopen	42	80	3
exkJCLRstor	42	08	3
exkJKBJobId	8		2
exkJKJJobIdN	4		2
exkJobname	2C		2
exknoJCL	42	02	3
exkOJCV	42	04	3
exkreason	42		2
exkSEnoAva	40	40	4
exkSEnoJPL	40	20	4
exkSEund	40	10	4
exksseq	4A		2
exkssname	3C		2
exktime	38		2
exktype	1		2
exk2bad	40	02	4
exk2fail	40	08	4
exk2fjes	41	80	4
exk2njss	40	01	4
exk2plte	40	04	4
exk3nckp	41	40	4
exk3oos	41	20	4

EXR - Exit Record (Exitdatensatz)

Name : DCLEXR

Funktion:

Dieses Segment deklariert einen Exitdatensatz. Exitdatensätze werden von SMF- und JES-Exits erstellt, über CSA-Puffer an den Ereignisschreiber übergeben und als Teil eines Ereignisdatensatzes vom Ereignisschreiber in eine Ereignisdatei geschrieben.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	80	exr	exit record	
0	(0)	CHARACTER	2	exrtype	record type	
0	(0)	CHARACTER	1	exrsyst	system type A!B	
1	(1)	CHARACTER	1	exrevtyp	event type 1!2!3!4!5!6	
2	(2)	CHARACTER	1	exrstype	event subtype (only type3)	

Offsets					
3	(3)	BITSTRING	1	exrflags	exit flags
		1... ..		exrretry	retry release command
		.1.. ..		exropcho	this job is in hold
		..1.		exroheld	job was held by tws
		...1		exrjkill	job killed by jes in rdr
	 1...		exrjcecc	error code from jcc
	1..		exrjccch	checked by jcc
	1.		exrjccer	error in jcc
	1		exrrlast	retcode(last) active
4	(4)	BITSTRING	1	exrtermf	termination flags
		1... ..		exrjcsct	exrjcsct is valid
		.1.. ..		exrjcsct	job has jcc sysout
		..1.		*	free D90C
		...1		exrmchld	1= msgclass is held
	 1...		exrcmq	1= requeue msg class
	1..		*	free D90C
	1.		exrjbtb	0=normal, 1=abend (job)
	1.		exrabend	0=normal, 1=abend (step)
	1		exrflush	0=normal, 1=flushed step
5	(5)	BITSTRING	1	exrerror	job error switch lcterror
		1... ..		exrfail	job failed
		.1.. ..		exrJQA	ON= is a JQA
		..1.		exrZ2level	
		...1		exrR4level	
	 1...		*	alloc but not unalloc done
	1..		exrcfal	job failed on cond codes
	1.		exrjjob	JCJOB processed ok D54A
	1		exrlastfl	last step flushed \$BGIA
6	(6)	SIGNED	2	exrgmtof	gmt offset in minutes
8	(8)	CHARACTER	8	exrjobn	job name

Offsets						
16	(10)	CHARACTER	8	exrjobid	job number	
24	(18)	CHARACTER	8	exrcreat	event creation time	
24	(18)	SIGNED	4	exrdate	date format (00yydddf)	
28	(1C)	SIGNED	4	exrtime	time format (secs*100)	
32	(20)	CHARACTER	8	exrjsrdr	jes reader date and time	
32	(20)	SIGNED	4	exrrdate	date format (00yydddf)	
36	(24)	SIGNED	4	exrrtime	time format (secs*100)	
40	(28)	SIGNED	4	exrsdate	operation start date	
44	(2C)	SIGNED	4	exrstime	operation start time	
48	(30)	SIGNED	4	exredate	operation end date	
52	(34)	SIGNED	4	exretime	operation end time	
52	(34)	SIGNED	4	exrorgid	nje origin job number	
56	(38)	CHARACTER	8	exrstepn	job step name	
56	(38)	CHARACTER	8	exronje	name of orig nje nod	
56	(38)	CHARACTER	1	exrclass	printout class	
57	(39)	CHARACTER	1	*	reserved	
58	(3A)	SIGNED	2	exrasid	job asid	
60	(3C)	SIGNED	4	exrexeid	NJE execution jobn	
64	(40)	CHARACTER	8	exrpstep	procedure step name	
64	(40)	CHARACTER	8	exrnnje	this/next nje node	
64	(40)	CHARACTER	8	exrform	form number	
72	(48)	SIGNED	2	exrcode	completion/condition code	
74	(4A)	CHARACTER	1	exrindic	status indicators	
		1...		exrjesv4	jes sp4 or above1/3P/	
		.1..		exrspun	spun off ds rcd	
		..1.		exrterm	oper terminated datagroup	
		...1		exrinter	oper interrupted :-	
	 1...		exrrstrt	oper restarted :-	

Offsets

	1..		exrndest	not final f/\$sysmsg 3P
	1.		exrnods4	no \$sysmsg found 3P
	1		exrsuspd	suspended
75	(4B)	UNSIGNED	1	exrstpnr	step number
75	(4B)	BITSTRING	1	exrpurge	job purge bits
		111.		*	not used
		...1		exrSDEP	SDEP filter used
	 1...		exrlastab	last step abended
	1..		exrstill	stepevents(all)
	1.		exrstnz	stepevents(no) flag
	1		exropcan	cancelled by oper
76	(4C)	CHARACTER	4	exropcid	tw's identifier

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exr	0		1
exrabend	4	02	4
exrasid	3A		4
exrcfal	5	04	3
exrclass	38		4
exrcmreq	4	08	3
exrcode	48		2
exrcreat	18		2
exrdate	18		3
exredate	30		2
exrerror	5		2
exretime	34		2
exrevtyp	1		3
exrexeid	3C		4
exrfail	5	80	3
exrflags	3		2
exrflush	4	01	3
exrform	40		4
exrgmtof	6		2
exrindic	4A		2
exrinter	4A	10	3
exrjtab	4	02	3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exrjccch	3	04	3
exrjceec	3	08	3
exrjccer	3	02	3
exrjjob	5	02	3
exrjcout	4	40	3
exrjcsct	4	80	3
exrjesv4	4A	80	3
exrjkill	3	10	3
exrjobid	10		2
exrjobn	8		2
exrJQA	5	40	3
exrjsrdr	20		2
exrlastab	4B	08	4
exrlastfl	5	01	3
exrmchld	4	10	3
exrndest	4A	04	3
exrnnje	40		3
exrnods4	4A	02	3
exrohld	3	20	3
exronje	38		3
exropcan	4B	01	4
exropcho	3	40	3
exropcid	4C		2
exrorgid	34		3
exrpstep	40		2
exrpurge	4B		3
exrrdate	20		3
exrretry	3	80	3
exrrlast	3	01	3
exrrstrt	4A	08	3
exrrtime	24		3
exrR4level	5	10	3
exrsdate	28		2
exrSDEP	4B	10	4
exrspun	4A	40	3
exrstall	4B	04	4
exrstepn	38		2
exrstime	2C		2
exrstnz	4B	02	4
exrstpnr	4B		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
exrstype	2		2
exrsuspd	4A	01	3
exrsyst	0		3
exrterm	4A	20	3
exrtermf	4		2
exrtime	1C		3
exrtype	0		2
exrZ2level	5	20	3

HTI - HTTP-Schnittstelle von C-Seite zu PLX-Seite (Modul EELHTCEC)

Name : DCLHTI

Funktion:

Das Array enthält Informationen für die PLX-Schnittstelle, die im Modul EELHTCEC implementiert ist.

Offsets				
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)
0	(0)	STRUCTURE	168	htiSubmitJob
0	(0)	CHARACTER	16	htiOccName
16	(10)	CHARACTER	10	htiOccIA
26	(1A)	CHARACTER	1	htiSubtype
27	(1B)	CHARACTER	1	htiReprocess
28	(1C)	SIGNED	4	htijclNrec
32	(20)	ADDRESS	4	htijclPtr
36	(24)	CHARACTER	36	htibrokerKey
36	(24)	CHARACTER	36	htibrokerKeyId
72	(48)	CHARACTER	8	htialias
72	(48)	SIGNED	4	htialiasLength
76	(4C)	ADDRESS	4	htialiasPtr
80	(50)	CHARACTER	8	htioutput
80	(50)	SIGNED	4	htioutputLength
84	(54)	ADDRESS	4	htioutputPtr
88	(58)	CHARACTER	52	htiREFinfo
88	(58)	CHARACTER	44	htiREFdsname
132	(88)	CHARACTER	8	htiREFmember
140	(8C)	CHARACTER	8	htiPosition
140	(8C)	SIGNED	4	htiwcycle
144	(90)	SIGNED	4	htirecnum
148	(94)	SIGNED	4	htiThreadnum

Offsets

152	(98)	ADDRESS	4	htiVARTABptr
156	(9C)	CHARACTER	12	htiVARIA

 Prüfpunktaktualisierung (UPCP) ==> zHTTPDqe.h eventTypeUpdCP

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	8	htiUpdChkPt	
0	(0)	SIGNED	4	htiUpdcyc	write cycle number
4	(4)	SIGNED	4	htiUpdrrec	record number in cycle

 Jobprotokollanforderung (JLGT) ==> zHTTPDqe.h eventTypeGetJoblog
 Jeder Jobprotokollsatz hat 133 Zeichen.
 htiStageArea ist 150 Sätze lang.

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	32	htijLRequest	
0	(0)	CHARACTER	8	htijLJobname	name of the JES job
8	(8)	CHARACTER	8	htijLJESid	ID of the JES job
16	(10)	SIGNED	4	htiFirstRec	num of the first requested rec (>=1)
20	(14)	SIGNED	4	htiNumOfRecs	num of the requested recs
24	(18)	ADDRESS	4	htiStageArea	stage JL area (150 records long)
28	(1C)	CHARACTER	1	htiIsComplete	'Y': JL retrieved up to the end; 'N': otherwise
29	(1D)	CHARACTER	3	htijLfiller	free

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
htialias	48		2
htialiasLength	48		3
htialiasPtr	4C		3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
htibrokerKey	24		2
htibrokerKeyId	24		3
htiFirstRec	10		2
htiIsComplete	1C		2
htijclNrec	1C		2
htijclPtr	20		2
htijLfiller	1D		2
htijLJESid	8		2
htijLJobname	0		2
htijLRequest	0		1
htiNumOfRecs	14		2
htiOccIA	10		2
htiOccName	0		2
htioutput	50		2
htioutputLength	50		3
htioutputPtr	54		3
htiPosition	8C		2
htirecnum	90		3
htiREFdsname	58		3
htiREFinfo	58		2
htiREFmember	84		3
htiReprocess	1B		2
htiStageArea	18		2
htiSubmitJob	0		1
htiSubtype	1A		2
htiThreadnum	94		2
htiUpdChkPt	0		1
htiUpdcyc	0		2
htiUpdrrec	4		2
htiVARIA	9C		2
htiVARTABptr	98		2
htiwcycle	8C		3

HTSA - Taskparameterbereich für HTTP-Server

Name : DCLHTSA

Funktion:

Dieser Steuerblock wird erstellt, initialisiert und freigegeben durch das PLX-Hauptmodul für die HTTP-Server-Task.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	584	htsa		
0	(0)	CHARACTER	456	htsaComm		
0	(0)	CHARACTER	4	htsadesc	block descriptor = 'HTSA'	
4	(4)	CHARACTER	2	htsaver	block mapping version	
6	(6)	BITSTRING	2	htsaflags		
		1...		htsassl		
8	(8)	ADDRESS	4	htsamcap	mca address	
12	(C)	ADDRESS	4	htsastopecbptr	pointer to stop ecb	
16	(10)	SIGNED	4	htsaPortNum	Local port number of server	
20	(14)	ADDRESS	4	htsaSyncecbptr	ptr to Cli to Serv Sync ecb	
24	(18)	SIGNED	4	htsaTCPtime	time out for recv	
28	(1C)	CHARACTER	8	htsacodep	code page	
36	(24)	CHARACTER	9	htsatcpjn	tcpip job name	
45	(2D)	CHARACTER	1	*	free	
46	(2E)	CHARACTER	1	htsasslamod	ssl auth mode	
47	(2F)	CHARACTER	1	htsasslktyp	ssl key ring type	
48	(30)	CHARACTER	53	htsahostn	local hostname used	
101	(65)	CHARACTER	3	*	free	
104	(68)	CHARACTER	65	htsaSSLastr	SSL authorization string	
169	(A9)	CHARACTER	3	*	free	
172	(AC)	CHARACTER	121	htsaSSLkrnm	SSL key ring name	
293	(125)	CHARACTER	3	*	free	
296	(128)	CHARACTER	121	htsaSSLkrpw	SSL key password	
417	(1A1)	CHARACTER	27	*	free	
444	(1BC)	ADDRESS	4	htsaIDMvsPtr	ID pointer	
448	(1C0)	ADDRESS	4	htsadiafp	DIAGNOSE flags address	
452	(1C4)	ADDRESS	4	htsadiadp	DIAGNOSE data address	
456	(1C8)	SIGNED	4	htsaproto	0 HTTP 1 HTTPS	
460	(1CC)	ADDRESS	4	htsaioc	pointer to ioc HTDS	
464	(1D0)	SIGNED	4	htsannum	entry in EDP table	
468	(1D4)	SIGNED	4	htsatnum	num of threads for server	

Offsets

472	(1D8)	SIGNED	4	htsahtsnum	number of hts entries
476	(1DC)	ADDRESS	4	htsahtsptr	hts address
480	(1E0)	CHARACTER	8	htsasyslvl	system level
488	(1E8)	ADDRESS	4	htsaiochTREF	pointer to ioc HTREF
492	(1EC)	ADDRESS	4	htsaJlibDCB	pointer to JBLIB dcb
496	(1F0)	ADDRESS	4	htsaJBuf	JBLIB buffer
500	(1F4)	CHARACTER	1	htsaVARSUB	Y: varsub needed N: varsub not needed
501	(1F5)	CHARACTER	3	*	free
504	(1F8)	SIGNED	4	htsareconf	HTREF recovery usage
508	(1FC)	SIGNED	4	htsacyconf	HTREF recovery usage
512	(200)	UNSIGNED	2	htsasseqconf	HTREF recovery usage
514	(202)	CHARACTER	54	*	free
568	(238)	SIGNED	4	*	free
572	(23C)	SIGNED	4	*	free
576	(240)	CHARACTER	8	*	free
584	(248)	CHARACTER		htsaend	end of htsa

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
htsa	0		1
htsacodep	1C		3
htsaComm	0		2
htsacyconf	1FC		2
htsadesc	0		3
htsadiadp	1C4		3
htsadiafp	1C0		3
htsaend	248		2
htsaflags	6		3
htsahostn	30		3
htsahtsnum	1D8		2
htsahtsptr	1DC		2
htsaIDMvsPtr	1BC		3
htsaioch	1CC		2
htsaiochTREF	1E8		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
htsaJBuf	1F0		2
htsaJlibDCB	1EC		2
htsamcap	8		3
htsannum	1D0		2
htsaPortNum	10		3
htsaproto	1C8		2
htsareconf	1F8		2
htsasseqconf	200		2
htsassl	6	80	4
htsasslamod	2E		3
htsaSSLastr	68		3
htsaSSLkrnm	AC		3
htsaSSLkrpw	128		3
htsasslktyp	2F		3
htsastopecbptr	C		3
htsaSyncecbptr	14		3
htsasyslvl	1E0		2
htsatcpjn	24		3
htsaTCPtime	18		3
htsatnum	1D4		2
htsaVARSUB	1F4		2
htsaver	4		3

JCFB - Rückmeldeinformation für JS-Schnittstelle

Name : DCLJCFB

Funktion:

Dieser Block wird immer an das aufrufende Modul zurückgegeben, wenn Daten über die JS-Schnittstelle durch die GET-Routine abgerufen werden.

Offsets für Daten werden nur festgelegt, wenn Daten über GS übergeben werden. HINWEIS: Das aufrufende Modul ist dafür zuständig, 'freemain' für den Speicher Speicher auszuführen, auf den jcfdatp verweist.

Die Speichergröße ist in 'jcfstg' und die Subpoolnummer in 'jcfsubp'.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	88	jcf	feedback area for js interf
0	(0)	CHARACTER	4	jcfdesc	descriptor always 'JCF'
4	(4)	CHARACTER	2	jcfvers	block version

Offsets					
6	(6)	CHARACTER	8	jcfdtype	data type that datp points t see dclcrj
14	(E)	BITSTRING 1...111 1111	1	jcfflags jcfjcl *	processing flags jcl indicator reserved
15	(F)	CHARACTER	1	*	not used
16	(10)	ADDRESS	4	jcfbuftp	pointer to jcl record in buf
20	(14)	ADDRESS	4	jcfdatp	pointer to data
24	(18)	SIGNED	4	jcfdato	offset to data (JS record,Variables..)
28	(1C)	SIGNED	4	jcfstg	amount of storage pointed to by jcfdatp
32	(20)	SIGNED	4	jcsubp	subpool in which stg pointed to by jcfdatp is getmained
36	(24)	SIGNED	4	jcflino	offset to js data lines
36	(24)	SIGNED	4	jcfsvaro	offset to prompt vars
40	(28)	ADDRESS	4	jcflinp	addr to js data lines
40	(28)	ADDRESS	4	jcfsvarp	addr to prompt vars entry
44	(2C)	SIGNED	4	jcflin	number of js data lines
44	(2C)	SIGNED	4	jcfvars	number of prompt vars
48	(30)	CHARACTER	8	jcjfrom	where jcl retrived ux002name,js, library
56	(38)	CHARACTER	8	jcjmem	jcl member name if jcjfrom = library name else blank
64	(40)	CHARACTER	8	*	reserved
72	(48)	CHARACTER	1	jcfvstat	var subst. status V or ' '
73	(49)	CHARACTER	15	*	reserved

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
jcf	0		1

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
jcflin	2C		2
jcfvars	2C		3
jcfbuftp	10		2
jcfdato	18		2
jcfdatp	14		2
jcfdesc	0		2
jcfdtype	6		2
jcfflags	E		2
jc fjcl	E	80	3
jc fjfrom	30		2
jc fjmem	38		2
jc flino	24		2
jc flinp	28		2
jc fstg	1C		2
jc fsubp	20		2
jc fsvaro	24		3
jc fsvarp	28		3
jc fvers	4		2
jc fvstat	48		2

JCL - LAYOUT

Name : DCLJCL

Funktion:
 JCLTWS - JCL-Layout
 REFTWS - JCL-Layout

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	*	JCLTWS	
0	(0)	CHARACTER	136	JCLBDY	FIXED PART OF JCL RECORD
0	(0)	CHARACTER	28	JCLKEY	
0	(0)	CHARACTER	16	JCLADID	APPLICATION NAME
16	(10)	CHARACTER	10	JCLIA	OCC INPUT ARRIVAL DATE+TIME
16	(10)	CHARACTER	6	JCLIAT1	DATE
22	(16)	CHARACTER	4	JCLIAT2	TIME
26	(1A)	SIGNED	2	JCLOPNUM	OPERATION NUMBER
28	(1C)	CHARACTER	4	JCLEYE	EYE CATCHER

Offsets

32	(20)	UNSIGNED	1	JCLVERS	VERSION NUMBER
33	(21)	CHARACTER	1	*	free
34	(22)	CHARACTER	8	JCLJOBNM	JOBNAME
42	(2A)	CHARACTER	12	JCLVARIA	VAR IA from HTI
54	(36)	CHARACTER	2	*	free

- =====
- 3 JCLWSN CHAR(4), WORK STATION NAME
 - 3 JCLUPDAT, LAST UPDATE, DATE+TIME
 - 4 JCLUPDT1 CHAR(6), DATE
 - 4 JCLUPDT2 CHAR(4), TIME
- =====

56	(38)	CHARACTER	8	JCLUSER	LAST UPDATE, USERID
64	(40)	CHARACTER	1	JCLUPTYP	UPDATE TYPE: not used
65	(41)	CHARACTER	1	JCLSTAT	OP. STATUS: not used
66	(42)	UNSIGNED	2	JCLLINES	NO OF LINES IN RECORD
68	(44)	CHARACTER	1	JCLFLAGS	FLAGBYTE
		1...		JCLJSFND	ON = JCL READ FROM JS
		.1.		JCLELITD	ONCE BEEN EDITED
		..1.		JCLDIRNOP	ON = NOP directive set
		...1 1111		*	NOT USED
69	(45)	CHARACTER	1	*	FREE
70	(46)	SIGNED	2	JCLVLINE	NUMBER OF VARIABLES
72	(48)	SIGNED	4	JCLSUBP	SUBPOOL FOR FREEMAIN
76	(4C)	UNSIGNED	2	JCLALIASLEN	Alias name length
78	(4E)	CHARACTER	2	JCLRES4	FREE
80	(50)	CHARACTER	56	JCLBROKERKEYID	Broker key identifier
136	(88)	CHARACTER	*	JCLVARDATA	Variable data

=====

JCL-Stromsatz

=====

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	80	JCLTAB(*)	
0	(0)	CHARACTER	80	JCLREC	

Offsets					
0	(0)	CHARACTER	72	JCLTEXT	COL 1-72 OF THE JCL RECORD
72	(48)	CHARACTER	8	JCLLNNO	COL 73-80 OF THE JCL RECORD

=====

JCL-Variablen

=====

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	80	JCLVTAB(*)	
0	(0)	CHARACTER	80	JCLV	
0	(0)	CHARACTER	8	CLVNAME	VARIABLE NAME
8	(8)	CHARACTER	16	JCLVTNAM	VARIABLE TABLE NAME
24	(18)	CHARACTER	1	JCLVTYPE	P = PROMPT, Y= SETUP,N=SUB
25	(19)	CHARACTER	1	JCLVSET	E(XIT), D(EFAULT), P(REP), VAR
26	(1A)	CHARACTER	8	JCLVUSER	USER ID
26	(1A)	CHARACTER	8	JCLVEXIT	EXIT NAME
26	(1A)	CHARACTER	8	JCLVSNAM	SETTING VARIABLE NAME
34	(22)	CHARACTER	44	JCLVVAL	VARIABLE VALUE SET
78	(4E)	SIGNED	2	JCLVLGT	LENGTH OF VALUE

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	*	REFTWS	
0	(0)	CHARACTER	188	REFBDY	

ALLGEMEINES LAYOUT BEI TWSJCL:

0	(0)	CHARACTER	28	REFKEY	
0	(0)	CHARACTER	16	REFADID	ADID
16	(10)	CHARACTER	10	REFIA	IA
16	(10)	CHARACTER	6	REFIAT1	
22	(16)	CHARACTER	4	REFIAT2	
26	(1A)	SIGNED	2	REFOPNUM	ALWAYS 1

Offsets

28	(1C)	CHARACTER	4	REFEYE	'REF '
32	(20)	UNSIGNED	1	REFVERS	'01'
33	(21)	CHARACTER	1	*	
34	(22)	CHARACTER	8	REFJOBNM	
42	(2A)	CHARACTER	12	REFVARIA	VAR IA from HTI
54	(36)	CHARACTER	2	*	free
- 3 REFWSN CHAR(4), é WORK STATION NAME					
- 3 REFUPDAT, é LAST UPDATE, DATE+TIME					
- 4 REFUPDT1 CHAR(6), DATE					
- 4 REFUPDT2 CHAR(4), TIME					
56	(38)	CHARACTER	8	REFUSER	LAST UPDATE, USERID
64	(40)	CHARACTER	1	REFUPTYP	UPDATE TYPE: NOT USED
65	(41)	CHARACTER	1	REFSTAT	OP. STATUS: NOT USED
66	(42)	UNSIGNED	2	REFLINES	NO OF LINES IN RECORD
68	(44)	CHARACTER	1	REFFLAGS	FLAGBYTE
69	(45)	CHARACTER	1	*	FREE
70	(46)	SIGNED	2	REFVLINE	NUMBER OF VARIABLES
72	(48)	SIGNED	4	REFSUBP	SUBPOOL FOR FREEMAIN
76	(4C)	UNSIGNED	2	REFALIASLEN	ALIAS NAME LENGTH
78	(4E)	CHARACTER	2	REFRES4	FREE
80	(50)	CHARACTER	56	REFBROKERKEYID	BROKER KEY IDENTIFIER

 ZUSÄTZLICHE DATEN NUR FÜR BYREF:

136	(88)	CHARACTER	44	REFDSNAME	DATA SET NAME
180	(B4)	CHARACTER	8	REFMEMBER	MEMBER NAME
188	(BC)	CHARACTER	*	REFVARDATA	

 JCL-VARIABLEN

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)
0	(0)	STRUCTURE	80	REFVTAB(*)
0	(0)	CHARACTER	80	REFV
0	(0)	CHARACTER	8	REFVNAME
8	(8)	CHARACTER	16	REFVTNAM

Offsets

24	(18)	CHARACTER	1	REFVTYPE
25	(19)	CHARACTER	1	REFVSET
26	(1A)	CHARACTER	8	REFVUSER
26	(1A)	CHARACTER	8	REFVEXIT
26	(1A)	CHARACTER	8	REFVSNAM
34	(22)	CHARACTER	44	REFVVAL
78	(4E)	SIGNED	2	REFVLGT

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
JCLADID	0		4
JCLALIASLEN	4C		3
JCLBDY	0		2
JCLBROKERKEYID	50		3
JCLEEDITD	44	40	4
JCLEYE	1C		3
JCLFLAGS	44		3
JCLIA	10		4
JCLIAT1	10		5
JCLIAT2	16		5
JCLJDIRNOP	44	20	4
JCLJOBNM	22		3
JCLJSFND	44	80	4
JCLKEY	0		3
JCLLINES	42		3
JCLLNNO	48		3
JCLOPNUM	1A		4
JCLREC	0		2
JCLRES4	4E		3
JCLSTAT	41		3
JCLSUBP	48		3
JCLTAB	0		1
JCLTEXT	0		3
JCLTWS	0		1
JCLUPTYP	40		3
JCLUSER	38		3
JCLV	0		2
JCLVARDATA	88		2
JCLVARIA	2A		3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
JCLVERS	20		3
JCLVEXIT	1A		4
JCLVLGT	4E		3
JCLVLINE	46		3
JCLVNAME	0		3
JCLVSET	19		3
JCLVSNAM	1A		5
JCLVTAB	0		1
JCLVTNAM	8		3
JCLVTYPE	18		3
JCLVUSER	1A		3
JCLVVAL	22		3
REFADID	0		4
REFALIASLEN	4C		3
REFBDY	0		2
REFBROKERKEYID	50		3
REFDSNAME	88		3
REFEYE	1C		3
REFFLAGS	44		3
REFIA	10		4
REFIAT1	10		5
REFIAT2	16		5
REFJOBNM	22		3
REFKEY	0		3
REFLINES	42		3
REFMEMBER	B4		3
REFOPNUM	1A		4
REFRES4	4E		3
REFSTAT	41		3
REFSUBP	48		3
REFTWS	0		1
REFUPTYP	40		3
REFUSER	38		3
REFV	0		2
REFVARDATA	BC		2
REFVARIA	2A		3
REFVERS	20		3
REFVEXIT	1A		4
REFVLGT	4E		3
REFVLINE	46		3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
REFVNAME	0		3
REFVSET	19		3
REFVSNAM	1A		5
REFVTAB	0		1
REFVTNAM	8		3
REFVTYPE	18		3
REFVUSER	1A		3
REFVVAL	22		3

JCL1 - Für VARSUB verwendete JCL

Name : DCLJCL1

Funktion:
JCLREC - JCL-Layout

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	*	JCLREC	
0	(0)	CHARACTER	80	JCLBDY	FIXED PART OF JCL RECORD
0	(0)	CHARACTER	28	JCLKEY	
0	(0)	CHARACTER	16	JCLADID	APPLICATION NAME
16	(10)	CHARACTER	10	JCLIA	OCC INPUT ARRIVAL DATE+TIME
16	(10)	CHARACTER	6	JCLIAT1	DATE
22	(16)	CHARACTER	4	JCLIAT2	TIME
26	(1A)	SIGNED	2	JCLOPNUM	OPERATION NUMBER
28	(1C)	CHARACTER	4	JCLEYE	EYE CATCHER
32	(20)	UNSIGNED	1	JCLVERS	VERSION NUMBER
33	(21)	CHARACTER	1	*	FREE
34	(22)	CHARACTER	8	JCLJOBNM	JOBNAME
42	(2A)	CHARACTER	4	JCLWSN	WORK STATION NAME
46	(2E)	CHARACTER	10	JCLUPDAT	LAST UPDATE, DATE+TIME
46	(2E)	CHARACTER	6	JCLUPDT1	DATE
52	(34)	CHARACTER	4	JCLUPDT2	TIME
56	(38)	CHARACTER	8	JCLUSER	LAST UPDATE, USERID

Offsets						
64	(40)	CHARACTER	1	JCLUPTYP	UPDATING FUNCTION: L: LTP W: WSD R: READY LIST M: MCP	
65	(41)	CHARACTER	1	JCLSTAT	OP. STATUS: S:SUB;;ED; V:SAVED; T:TSAVED;	
66	(42)	UNSIGNED	2	JCLLINES	NO OF LINES IN RECORD	
68	(44)	CHARACTER	1	JCLFLAGS	FLAGBYTE	
		1...		JCLJSFND	ON = JCL READ FROM JS	
		.1..		JCLEEDITD	ONCE BEEN EDITED	
		..1.		JCLJDIRNOP	ON = NOP directive set	
		...1 1111		*	NOT USED	
69	(45)	CHARACTER	1	*	FREE JCLC	
70	(46)	SIGNED	2	JCLVLINE	NUMBER OF VA- RIABLES JCLA	
72	(48)	SIGNED	4	JCLSUBP	SUBPOOL FOR FREEMAIN	
76	(4C)	CHARACTER	4	JCLRES4	FREE	
80	(50)	CHARACTER	80	JCLTAB (*)	JCL RECORDS	
80	(50)	CHARACTER	72	JCLTEXT	COL 1-72 OF THE JCL RECORD	
152	(98)	CHARACTER	8	JCLLNNO	COL 73-80 OF THE JCL RECORD	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimensi- on)	Beschreibung	
0	(0)	STRUCTURE	80	JCLVTAB (*)		
0	(0)	CHARACTER	80	JCLV		
0	(0)	CHARACTER	8	JCLVNAME	VARIABLE NAME	
8	(8)	CHARACTER	16	JCLVTNAM	VARIABLE TAB- LE NAME	
24	(18)	CHARACTER	1	JCLVTYPE	P = PROMPT, Y= SETUP,N=SUB	
25	(19)	CHARACTER	1	JCLVSET	E(XIT), D(EFAULT), P(REP), VAR	
26	(1A)	CHARACTER	8	JCLVUSER	USER ID	
26	(1A)	CHARACTER	8	JCLVEXIT	EXIT NAME	

Offsets					
26	(1A)	CHARACTER	8	JCLVSNAM	SETTING VARIABLE NAME
34	(22)	CHARACTER	44	JCLVVAL	VARIABLE VALUE SET
78	(4E)	SIGNED	2	JCLVLGT	LENGTH OF VALUE

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
JCLADID	0		4
JCLBDY	0		2
JCLEEDITD	44	40	4
JCLEYE	1C		3
JCLFLAGS	44		3
JCLIA	10		4
JCLIAT1	10		5
JCLIAT2	16		5
JCLJDIRNOP	44	20	4
JCLJOBNM	22		3
JCLJSFND	44	80	4
JCLKEY	0		3
JCLLINES	42		3
JCLLNNO	98		3
JCLOPNUM	1A		4
JCLREC	0		1
JCLRES4	4C		3
JCLSTAT	41		3
JCLSUBP	48		3
JCLTAB	50		2
JCLTEXT	50		3
JCLUPDAT	2E		3
JCLUPDT1	2E		4
JCLUPDT2	34		4
JCLUPTYP	40		3
JCLUSER	38		3
JCLV	0		2
JCLVERS	20		3
JCLVEXIT	1A		4
JCLVLGT	4E		3
JCLVLINE	46		3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
JCLVNAME	0		3
JCLVSET	19		3
JCLVSNAM	1A		5
JCLVTAB	0		1
JCLVTNAM	8		3
JCLVTYPE	18		3
JCLVUSER	1A		3
JCLVVAL	22		3
JCLWSN	2A		3

JDA - Vordefinierte OPC/ESA-Variablen

Name : DCLJDA

Funktion:

Dieser Block beschreibt die Zuordnung und Adressierung von Variablenwerten, die vom Block 'dcljdv' definiert werden.

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	12	jdat	variable data
0	(0)	ADDRESS	4	jdatdvp	pointer to jdat table
4	(4)	SIGNED	4	jdatnum	number of variables
8	(8)	SIGNED	4	jdatsize	total size of variable entries
12	(C)	CHARACTER		jdatdtab	end of table header

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	21	jdae	variable table entry
0	(0)	CHARACTER	8	jdaevar	variable name
8	(8)	SIGNED	4	jdaevsz	size of variable name
12	(C)	SIGNED	4	jdaevar	offset to value based jdatda
16	(10)	SIGNED	4	jdaesiz	size of variable value
20	(14)	BITSTRING	1	jdaeiflg	additional flags
		1...		jdaesetp	setup avail on Yes

Offsets

.1..	jdajett	Job using vars is ETT
..11 1111	*	reserved

 Dieser Block enthält die Werte aller vordefinierten JCL-Variablen.
 Diese Variablen müssen nicht in benutzerdefinierten
 JCL-Variablentabellen definiert sein.

Hinweis:

Wenn Variablen im Block 'jdav' hinzugefügt oder entfernt werden,
 muss der Wert 'jdavnum' entsprechend aktualisiert werden.

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimensi- on)	Beschreibung
0	(0)	STRUCTURE	479	jdav	
0	(0)	CHARACTER	4	jdavdesc	descriptor always 'JDAV'
4	(4)	CHARACTER	2	jdavvers	block version
6	(6)	CHARACTER	6	oymd1	occ ia yymmdd
12	(C)	CHARACTER	8	oymd2	occ ia yy/mm/dd
20	(14)	CHARACTER	10	oymd3	occ ia yyyy/mm/dd
30	(1E)	CHARACTER	6	odmy1	occ ia ddmmyy
36	(24)	CHARACTER	8	odmy2	occ ia dd/mm/yy
44	(2C)	CHARACTER	6	oym	occ ia yyyyymm
50	(32)	CHARACTER	8	oymd	occ ia yyyyymmdd
58	(3A)	CHARACTER	5	oyydd	occ ia yyddd
63	(3F)	CHARACTER	3	owwd	occ week day within week
66	(42)	CHARACTER	4	oyymm	occ ia date yymm
70	(46)	CHARACTER	4	ommyy	occ ia date mmyy
74	(4A)	CHARACTER	2	oyy	occ ia year
76	(4C)	CHARACTER	4	oyyyy	occ ia year
80	(50)	CHARACTER	2	omm	occ ia month
82	(52)	CHARACTER	2	odd	occ ia day within month
84	(54)	CHARACTER	2	ohh	occ ia hour
86	(56)	CHARACTER	4	ohhmm	occ ia hour minute
90	(5A)	CHARACTER	4	ossid	Subsystem name
94	(5E)	CHARACTER	54	oxjobnam	Extended Job Name
148	(94)	CHARACTER	3	oopno	Oper number
151	(97)	CHARACTER	1	oday	Occ ia day in week 1=monday

Offsets					
152	(98)	CHARACTER	16	oadid	Occ application id
168	(A8)	CHARACTER	4	olhhmm	latest start hour minute
172	(AC)	CHARACTER	2	olhh	latest start hour
174	(AE)	CHARACTER	2	olmm	latest start month
176	(B0)	CHARACTER	6	olymd	latest start yymmdd
182	(B6)	CHARACTER	4	olmd	latest start mmdd
186	(BA)	CHARACTER	2	oldd	latest start dd (day in mon)
188	(BC)	CHARACTER	1	olday	latest start day of wk(1-7)
189	(BD)	CHARACTER	2	olwk	latest start week of year
191	(BF)	CHARACTER	5	olyydd	latest start yyddd
196	(C4)	CHARACTER	6	cyymmdd	current date yymmdd
202	(CA)	CHARACTER	6	cddmmyy	current date ddmmyy
208	(D0)	CHARACTER	6	cyyyymm	current date year month
214	(D6)	CHARACTER	5	cyydd	current date year day number
219	(DB)	CHARACTER	4	cyymm	current date year month
223	(DF)	CHARACTER	4	cmmyy	current date month year
227	(E3)	CHARACTER	2	cy	current date year
229	(E5)	CHARACTER	4	cyyy	current date year
233	(E9)	CHARACTER	2	cmm	current month
235	(EB)	CHARACTER	2	cdd	current day within month
237	(ED)	CHARACTER	3	cwwd	current week, day in week
240	(F0)	CHARACTER	8	cymd	current date, yyyymmdd
248	(F8)	CHARACTER	1	cday	current day in week,1=monday
249	(F9)	CHARACTER	2	chh	current hour of day
251	(FB)	CHARACTER	4	chhmm	current hour and minute
255	(FF)	CHARACTER	3	cddd	current day within year
258	(102)	CHARACTER	2	cww	current week in year

Offsets					
260	(104)	CHARACTER	8	chhmmssx	current time HHMMSSxx
268	(10C)	CHARACTER	1	cfreeday	cur time freeday Y!N AGLA
269	(10D)	CHARACTER	3	oddd	occ ia day within year
272	(110)	CHARACTER	2	oww	occ ia week in year
274	(112)	CHARACTER	16	oadowner	ad owner
290	(122)	CHARACTER	1	ofreeday	occ ia is freeday Y!N
291	(123)	CHARACTER	8	ojobname	jobname
299	(12B)	CHARACTER	8	oaugroup	authority group
307	(133)	CHARACTER	16	ocalid	calendar name
323	(143)	CHARACTER	1	owwmonth	week number in ia month
324	(144)	CHARACTER	1	owwlast	last week in month = Y else N
325	(145)	CHARACTER	6	chhmmss	current time HHMMSS \$CBPA
331	(14B)	CHARACTER	44	oettcrit	ETT criteria
375	(177)	CHARACTER	1	oetttyp	ETT type J or R
376	(178)	CHARACTER	8	oettjob	ETT job name
384	(180)	CHARACTER	8	oettjid	ETT job id
392	(188)	CHARACTER	35	oettgroot	ETT gdg root
427	(1AB)	CHARACTER	44	oettevnam	ETT full event name
471	(1D7)	CHARACTER	8	oettggen	ETT gdg generati- on
479	(1DF)	CHARACTER		jdavend	end label of block

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
cday	F8		2
cdd	EB		2
cddd	FF		2
cddmmyy	CA		2
cfreeday	10C		2
chh	F9		2
chhmm	FB		2
chhmmss	145		2
chhmmssx	104		2
cmm	E9		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
cmmyy	DF		2
cww	102		2
cwwd	ED		2
cymd	F0		2
cyy	E3		2
cyddd	D6		2
cyymm	DB		2
cyymmdd	C4		2
yyyy	E5		2
yyyyymm	D0		2
jdae	0		1
jdaeFlgs	14		2
jdaeJett	14	40	3
jdaeSetp	14	80	3
jdaesiz	10		2
jdaevao	C		2
jdaevar	0		2
jdaevsz	8		2
jdat	0		1
jdatdavp	0		2
jdatnum	4		2
jdatSize	8		2
jdatTab	C		2
jdav	0		1
jdavdesc	0		2
jdavend	1DF		2
jdavvers	4		2
oadid	98		2
oadowner	112		2
oagroup	12B		2
ocalid	133		2
oday	97		2
odd	52		2
oddd	10D		2
odmy1	1E		2
odmy2	24		2
oettcrit	14B		2
oettevnam	1AB		2
oettggen	1D7		2
oettgroot	188		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
oettjid	180		2
oettjob	178		2
oetttyp	177		2
ofreeday	122		2
ohh	54		2
ohhmm	56		2
ojobname	123		2
olday	BC		2
oldd	BA		2
olhh	AC		2
olhhmm	A8		2
olmd	B6		2
olmm	AE		2
olwk	BD		2
olymd	B0		2
olyydd	BF		2
omm	50		2
omyy	46		2
oopno	94		2
ossid	5A		2
oww	110		2
owwd	3F		2
owwlast	144		2
owwmonth	143		2
oxjobnam	5E		2
oym	2C		2
oymd	32		2
oymd1	6		2
oymd2	C		2
oymd3	14		2
oyy	4A		2
oyydd	3A		2
oyymm	42		2
oyyyy	4C		2

JDT - Durch SETVAR definierte temporäre Variablen

Name : DCLJDT

Funktion:

Dieser Block beschreibt die temporären Variablen der Anweisung SETVAR.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	*	jdt		
0	(0)	CHARACTER	20	JdtHead	fixed part of JDT	
0	(0)	CHARACTER	4	Jdteye	descriptor always 'JDT '	
4	(4)	CHARACTER	2	Jdtvers	block version	
6	(6)	CHARACTER	1	JdtWarnMsg	Y= issue "not ref" messages N= do not issue message	
7	(7)	CHARACTER	1	*	not used	
8	(8)	SIGNED	4	Jdtvars	of variables in table	
12	(C)	SIGNED	4	Jdtmax	max of variables allowed	
16	(10)	ADDRESS	4	Jdtnxtp	address of next JDT ! 0	
20	(14)	CHARACTER	*	JdtVarTab	Variable part of JDU	
20	(14)	CHARACTER	64	JdtVariables (*)	Address of variables	
20	(14)	CHARACTER	4	JdtFlags	flags	
20	(14)	CHARACTER	1	JdtUsed	Y = variavle was referenced	
21	(15)	CHARACTER	1	JdtPhase	U = SETUP, S = SUBMIT	
22	(16)	CHARACTER	2	*	Y = variavle was referenced	
24	(18)	SIGNED	4	JdtVarL	length of total string that is to be substituted	
28	(1C)	CHARACTER	48	JdtValue	edit value to be used at substitution of this var.	
76	(4C)	CHARACTER	8	JdtVname	name of variable	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	64	JdtSvar	mask for JDT substitution	
0	(0)	CHARACTER	4	*	HINWEIS: Diese Maske darf nur einem Eintrag in JdtVariables entsprechen.	
0	(0)	CHARACTER	1	JdtSused	var used in phase	
1	(1)	CHARACTER	1	JdtSPhase	current phase	

Offsets

2	(2)	CHARACTER	2	*	
4	(4)	SIGNED	4	JdtSvarL	value length
8	(8)	CHARACTER	48	JdtSValue	substitution value
56	(38)	CHARACTER	8	JdtSVname	name of variable

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
jdt	0		1
Jdtmax	C		3
Jdtvars	8		3
Jdteye	0		3
JdtFlags	14		4
JdtHead	0		2
Jdtnxtp	10		3
JdtPhase	15		5
JdtSPhase	1		3
JdtSused	0		3
JdtSValue	8		2
JdtSvar	0		1
JdtSvarL	4		2
JdtSVname	38		2
JdtUsed	14		5
JdtValue	1C		4
JdtVariables	14		3
JdtVarL	18		4
JdtVarTab	14		2
Jdtvers	4		3
JdtVname	4C		4
JdtWarnMsg	6		3

JDU - Neu definierte OPC/ESA-Variablen für Datum und Zeit

Name : DCLJDU

Funktion:

Dieser Block beschreibt den Inhalt und das Format von vordefinierten OPC-Variablen, wie sie durch die OPC-Anweisung SETFORM in der JCL neu definiert werden.

HINWEIS: Die Steuerblöcke JDU und JDUV befinden sich im zusammenhängenden Speicher. Die Variablen JudCurr, JduOcc, JduOccl werden initialisiert, wenn Jdu-Steuerblöcke (cbs) erstellt werden. Die Werte werden vordefinierten Occ- und CurrDate- /CurrTime-Variablen entnommen.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	1192	jdu	
0	(0)	CHARACTER	168	JduHead	fixed part of JDU
0	(0)	CHARACTER	4	Jdueye	descriptor always 'JDU '
4	(4)	CHARACTER	2	Jduvers	block version
6	(6)	CHARACTER	2	*	not used
8	(8)	SIGNED	4	JduTotsize	total size for JDU* blocks
12	(C)	CHARACTER	8	JduCurrYmd	current date YYYYMMDD
20	(14)	CHARACTER	5	JduCurrJulian	current julian date YYDDD
25	(19)	CHARACTER	6	JduCurrTime	current time HHMMSS
31	(1F)	CHARACTER	5	JduOlJulian	oper latest start julian
36	(24)	CHARACTER	8	JduOlYmd	oper latest start YYYYMMDD
44	(2C)	CHARACTER	4	JduOlTime	oper latest start HHMM
48	(30)	CHARACTER	8	JduOiDate	oper ia date YYYYMMDD
56	(38)	CHARACTER	5	JduOiJulian	oper ia julian date YYDDD
61	(3D)	CHARACTER	4	JduOiTime	oper ia time HHMM
65	(41)	CHARACTER	8	JduOccYmd	occ ia date YYYYMMDD
73	(49)	CHARACTER	5	JduOccJulian	occ ia julian date YYDDD
78	(4E)	CHARACTER	4	JduOccTime	occ ia time HHMM
82	(52)	CHARACTER	8	JduOccLw	occ last work day in month
90	(5A)	CHARACTER	5	JduOccLwJulian	occ last work day JULIAN
95	(5F)	CHARACTER	8	JduOccLc	occ last cal day in month
103	(67)	CHARACTER	5	JduOccLcJulian	occ last cal day JULIAN
108	(6C)	CHARACTER	8	JduOccFc	occ first cal day in month
116	(74)	CHARACTER	5	JduOccFcJulian	occ first cal day JULIAN
121	(79)	CHARACTER	8	JduOccFw	occ first work day in month

Offsets					
129	(81)	CHARACTER	5	JduOccFwJulian	occ first work day JULIAN
134	(86)	CHARACTER	8	JduOccFwYear	occ first work day in year
142	(8E)	CHARACTER	5	JduOccFwJulianYr	occ first work day JULIAN
147	(93)	CHARACTER	8	JduOccLwYear	occ last work day in year
155	(9B)	CHARACTER	5	JduOccLwJulianYr	occ last work day JULIAN
160	(A0)	CHARACTER	8	*	spare
168	(A8)	CHARACTER	1024	JduVarTab	Variable part of JDU
168	(A8)	CHARACTER	64	JduVariables (16)	Address of variables
168	(A8)	SIGNED	4	JduOffset	offset to variable from
172	(AC)	SIGNED	4	JduVarL	length of total string that is to be substituted
176	(B0)	CHARACTER	47	JduValue	edit value to be used at substitution of this var.
223	(DF)	CHARACTER	1	JduVflags	processing flags
		1... ..		JduVSetup	1= variable avail. at setup
		.111 1111		*	not used
224	(E0)	CHARACTER	8	JduVname	name of variable

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	942	jduv	description of a variable
0	(0)	CHARACTER	46	JduvHead	fixed part of description
0	(0)	SIGNED	2	Jduv	number of values in table
2	(2)	CHARACTER	2	JduvFormat	'DA' = Date related 'TI' = Time format
4	(4)	CHARACTER	40	JduvSdelim	string before date/time
44	(2C)	SIGNED	2	JduvSdelimL	length of delim before strg
46	(2E)	CHARACTER	56	JduvDesc (16)	Description

Offsets						
46	(2E)	CHARACTER	4	JduvType	Keyword type (MM,DD,CC,YY,CCYY,HH,SS)	
50	(32)	SIGNED	2	JduvStart	Start position in string	
52	(34)	SIGNED	2	JduvEnd	End position in string	
54	(36)	CHARACTER	4	JduvVal	Value for this type 1993,10bb and so on	
58	(3A)	SIGNED	2	JduvValL	Length of value	
60	(3C)	CHARACTER	40	JduvDelim	Delimiter after this one	
100	(64)	SIGNED	2	JduvDelimL	length of delimiter	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	64	JduSvariable	Used to conform with other methods for addressing vars to be subst. in JHSLN/JHTRN NOTE: This mapping must map a single entry in JduVartab	
0	(0)	SIGNED	4	JduSOffset	offset to variable from	
4	(4)	SIGNED	4	JduSVarL	length of total string that is to be substituted	
8	(8)	CHARACTER	47	JduSValue	edit value to be used at substitution of this var.	
55	(37)	CHARACTER	1	JduSFlags	processing flags	
		1... ..		JduSetup	1= variable avail. art setup	
		.111 1111		*	not used	
56	(38)	CHARACTER	8	JduSVname	name of variable	

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
jdu	0		1
JduCurrJulian	14		3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
JduCurrTime	19		3
JduCurrYmd	C		3
Jdueye	0		3
JduHead	0		2
JduOccFc	6C		3
JduOccFcJulian	74		3
JduOccFw	79		3
JduOccFwJulian	81		3
JduOccFwJulianYr	8E		3
JduOccFwYear	86		3
JduOccJulian	49		3
JduOccLc	5F		3
JduOccLcJulian	67		3
JduOccLw	52		3
JduOccLwJulian	5A		3
JduOccLwJulianYr	9B		3
JduOccLwYear	93		3
JduOccTime	4E		3
JduOccYmd	41		3
JduOffset	A8		4
JduOiDate	30		3
JduOiJulian	38		3
JduOiTime	3D		3
JduOlJulian	1F		3
JduOlTime	2C		3
JduOlYmd	24		3
JduSFlags	37		2
JduSOffset	0		2
JduSetup	37	80	3
JduSValue	8		2
JduSvariable	0		1
JduSVarL	4		2
JduSVname	38		2
JduTotsize	8		3
jduv	0		1
Jduv	0		3
JduValue	B0		4
JduVariables	A8		3
JduVarL	AC		4
JduVarTab	A8		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
JduvDelim	3C		3
JduvDelimL	64		3
JduvDesc	2E		2
JduvEnd	34		3
Jduvers	4		3
JduVflags	DF		4
JduvFormat	2		3
JduvHead	0		2
JduVname	E0		4
JduvSdelim	4		3
JduvSdelimL	2C		3
JduvSetup	DF	80	5
JduvStart	32		3
JduvType	2E		3
JduvVal	36		3
JduvValL	3A		3

JHS - Gemeinsam genutzte Parameter für JHSET und JHUTL

Name : DCLJHS

Funktion:

Dieser Block enthält Parameter, die von JHSET und JHUTL verwendet werden.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	212	JHSETcom		
0	(0)	CHARACTER	60	wformat	format string in local	
60	(3C)	CHARACTER	16	Wcalendar	occurrence calendar name	
		2 HCPAREA char(length(HCPm)), parameter area				
76	(4C)	CHARACTER	6	occia_base	ia date (from OYMD1)	
82	(52)	CHARACTER	6	olate_base	latest start (from OLYMD)	
88	(58)	CHARACTER	6	cdate_base	current d. base (from CYMMDD)	
94	(5E)	CHARACTER	6	occia_baset	ia time (from OHHMM)	
100	(64)	CHARACTER	6	olate_baset	latest start time(from OLHHMM)	

Offsets					
106	(6A)	CHARACTER	8	ctime_baset	current time (from HHMMSSXX)
114	(72)	CHARACTER	8	varname	variable name
122	(7A)	CHARACTER	3	number	number to add/ subtract (date)
125	(7D)	CHARACTER	3	*	free
128	(80)	SIGNED	4	seconds	sec*100 to add/ subtract (time)
132	(84)	CHARACTER	2	Ctype	type to add/ subtract
134	(86)	CHARACTER	6	fase	= parm phase
140	(8C)	SIGNED	4	signpos	position of sign
144	(90)	SIGNED	4	var	1st pos for category
148	(94)	SIGNED	4	var_end	last pos for category
152	(98)	SIGNED	4	digit	1st pos of digits
156	(9C)	SIGNED	4	digit_end	last pos of digits
160	(A0)	SIGNED	4	type_start	1st pos for type (wd,cd,...)
164	(A4)	SIGNED	4	var_cat	variable category
168	(A8)	SIGNED	4	i	
172	(AC)	SIGNED	4	j	
176	(B0)	SIGNED	4	k	
180	(B4)	SIGNED	4	x	
184	(B8)	SIGNED	4	j1	loop indexes
188	(BC)	SIGNED	4	rc	local return code
192	(C0)	SIGNED	4	currvar	work index
196	(C4)	SIGNED	4	currJdt	current JDt entry
200	(C8)	ADDRESS	4	wtubptr	= tubptr
204	(CC)	ADDRESS	4	wjduvptr	= jduvptr \$CQOC
208	(D0)	ADDRESS	4	wjdtptr	= jdtptr \$CQOA

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
cdate_base	58		2
ctime_baset	6A		2
Ctype	84		2
currJdt	C4		2
currvar	C0		2
digit	98		2
digit_end	9C		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
fase	86		2
i	A8		2
j	AC		2
JHSETcom	0		1
j1	B8		2
k	B0		2
number	7A		2
occia_base	4C		2
occia_baset	5E		2
olate_base	52		2
olate_baset	64		2
rc	BC		2
seconds	80		2
signpos	8C		2
type_start	A0		2
var	90		2
var_cat	A4		2
var_end	94		2
varname	72		2
Wcalendar	3C		2
wformat	0		2
wjdtpr	D0		2
wjduvpr	CC		2
wtubpr	C8		2
x	B4		2

JV - Layout für JCL-Variablentabelle

Name : DCLJV

Funktion:
JV-Layout.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	*	fv	jcl variable table	
0	(0)	CHARACTER	98	fvcommon	identifier	
0	(0)	CHARACTER	2	*	reserved for vsam mods/02	
2	(2)	CHARACTER	23	fvkey	key of record table	

Offsets						
2	(2)	CHARACTER	16	jvtable	jcl variable table id	
18	(12)	CHARACTER	7	*	always blank	
25	(19)	CHARACTER	1	*	not used	
26	(1A)	CHARACTER	8	jvlu	last updating user	
34	(22)	CHARACTER	4	jvlt	last update time hhmm	
38	(26)	CHARACTER	6	jvld	last update date yymmdd	
44	(2C)	CHARACTER	24	jvdes	table description	
68	(44)	SIGNED	2	jvvar	number of vars in table	
70	(46)	CHARACTER	16	jvown	owner id	
86	(56)	CHARACTER	2	*	not used	
88	(58)	CHARACTER	8	jvluts	last update timestamp	
96	(60)	CHARACTER	2	*	not used	
98	(62)	CHARACTER	*	jvarsect	variable part of table	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	476	jvtab	jcl var table, var part	
0	(0)	CHARACTER	8	jvvar	jcl variable name	
8	(8)	CHARACTER	44	jvdfi	jcl variable def value	
52	(34)	CHARACTER	1	jvuc	Y = uppercase, N=mixed	
53	(35)	CHARACTER	1	jvstp	prompt / setup / submit	
54	(36)	SIGNED	2	jvlg	value length	
56	(38)	CHARACTER	7	jvtyp	verification type	
63	(3F)	CHARACTER	8	jvex	substitution exit name	
71	(47)	CHARACTER	1	jvinp	input required	
72	(48)	SIGNED	2	jvpos	replace position jcl data	
74	(4A)	CHARACTER	1	jvnum	numeric	
75	(4B)	CHARACTER	2	jvcmp	comparison operator	
77	(4D)	CHARACTER	44	jvpat	validation pattern	
121	(79)	CHARACTER	102	jvvld	valid values	
121	(79)	CHARACTER	51	jvvld1	first line	

Offsets

172	(AC)	CHARACTER	51	javld2	second line
223	(DF)	CHARACTER	204	jvtxt	dialog text
427	(1AB)	CHARACTER	20	jvtdes	description
447	(1BF)	CHARACTER	1	*	reserved
448	(1C0)	SIGNED	2	jvnrp	number of dep values
450	(1C2)	CHARACTER	8	jvind	independent variable name
458	(1CA)	CHARACTER	2	jvvers	version number
460	(1CC)	CHARACTER	2	jvsubs	substring start posVJA
462	(1CE)	CHARACTER	2	jvsubl	substring length
464	(1D0)	CHARACTER	12	*	reserved

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	88	jvd	dependencies
0	(0)	CHARACTER	44	jvdiv	value of setting variable
44	(2C)	CHARACTER	44	jvddv	dependent variable value

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
jv	0		1
jvvar	44		3
jvarsect	62		2
jvcmp	4B		2
jvcommon	0		2
jvd	0		1
jvddv	2C		2
jvdes	2C		3
jvdfi	8		2
jvdiv	0		2
jvex	3F		2
jvind	1C2		2
jvinp	47		2
jvkey	2		3
jvld	26		3
jvlg	36		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
jvlt	22		3
jvlu	1A		3
jvluts	58		3
jvnrp	1C0		2
jvnum	4A		2
jvown	46		3
jvpat	4D		2
jvpos	48		2
jvstp	35		2
jvsubl	1CE		2
jvsubs	1CC		2
jvtab	0		1
jvtable	2		4
jvtdes	1AB		2
jvtxt	DF		2
jvtyp	38		2
jvuc	34		2
jvvar	0		2
jvvers	1CA		2
jvvld	79		2
jvvld1	79		3
jvvld2	AC		3

MCA - Gemeinsamer TWS/ESA-Bereich

Name : DCLMCA

Funktion:

Dieses Segment deklariert den gemeinsamen TWS/ESA-Bereich.

Die meisten TWS/ESA-Steuerblöcke können über den MCA erreicht werden.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	392	mca	TWS/ESA communication area
0	(0)	CHARACTER	4	mcadesc	block descriptor = 'MCA'
4	(4)	CHARACTER	10	mcacode	MCA block code level
4	(4)	CHARACTER	2	mcaver	block mapping version
6	(6)	CHARACTER	8	mcafmid	TWS/ESA tracker fmid
14	(E)	BITSTRING	2	mcaflags	mca options

Offsets

		1..		mcatso	1: tso user built mca
		.1..		mcaasxb	1: asxb was modified
		..1.		mcaacee	1: acee built
		...1		mcaracf	1: subres profiles
	 1..		mcaesa	: active on mvs/esa
	1..		mcasep4	1: mvs/sp 4.1 or later
	1.		mcamlog	1: msg log is reserved
	1		mca313	1: mvs/sp 3.1.3 or later
15	(F)	1..		mcaqfcan	1: ss cancl'd due to qfull
		.1..		mcanoprm	1: do not open parmlib
		..1.		mcaedpw	1: waiting for edp
		...1		mcasep422	1: mvs/sp 4.2.2 or later
	 1..		mcapace	1: event inflow paced
	1..		mcaupace	1: emgr ask for resume
	1.		mcartrq	1: queue to DRTQ
	1		mcatcpr	1: ta task restarting

TWS/ESA-Steuerblockadressen folgen

16	(10)	ADDRESS	4	mcaascb	eqqmajor ascb address
20	(14)	ADDRESS	4	mcamtcb	eqqmajor tcb address
24	(18)	ADDRESS	4	mcaopts	addr of options block
28	(1C)	ADDRESS	4	mcaftab	file table address
32	(20)	ADDRESS	4	mcastemj	addr 1st major subtask
36	(24)	ADDRESS	4	mcastenm	addr 1st nmm subtask
40	(28)	ADDRESS	4	mcastegs	addr 1st genserv subtask
44	(2C)	ADDRESS	4	mcaopecb	addr of oper cmd ecb
48	(30)	ADDRESS	4	mcassvt	addr of tws ssvt
52	(34)	ADDRESS	4	mcassct	addr of tws ssct
56	(38)	ADDRESS	4	mcatsob	addr of tso table block

Offsets						
60	(3C)	ADDRESS	4	mcanmmb	addr of nmm parameters	
64	(40)	ADDRESS	4	mcaedpb	addr of edp block	
68	(44)	ADDRESS	4	mcaprip	addr of pri block	
72	(48)	ADDRESS	4	mcasrap	addr of serv routn array	
76	(4C)	ADDRESS	4	mcasure	addr of su/re table	
80	(50)	ADDRESS	4	mcanabb	addr of vtam i/o params	
84	(54)	ADDRESS	4	mcadiap	addr of diagnose options	
88	(58)	ADDRESS	4	mcacpnq	addr of cp enq statistics	
92	(5C)	ADDRESS	4	mcatmlog	tcb addr of mlog reserver	
96	(60)	ADDRESS	4	mcajanpc	js interface anchor block	
100	(64)	ADDRESS	4	mcaxsip	XCF system info cb	
104	(68)	ADDRESS	4	mcaqfecb	addr to q full ecb	
108	(6C)	ADDRESS	4	mcaasip	APPC system info cb	
112	(70)	ADDRESS	4	mcaSyncEcbPtr	HT Client to Server Sync	
116	(74)	ADDRESS	4	mcaTSRAp	addr of Topology parsing	
120	(78)	ADDRESS	4	mcaFSRAp	addr of Data Store service routines	
124	(7C)	ADDRESS	4	mcasub	addr of sub parm area	
128	(80)	ADDRESS	4	mcassxp	SSX block address	
132	(84)	ADDRESS	4	mcarsip	RODM system info cb	
136	(88)	ADDRESS	4	mcajltbl	mca job log table ptr	
140	(8C)	ADDRESS	4	mcarodmopt	RODM options table	
144	(90)	ADDRESS	4	mcahcm	HCMMain address	
148	(94)	ADDRESS	4	mcaID	Local Id event area	
152	(98)	ADDRESS	4	mcaJopts	address of JOBOPTS parms	
156	(9C)	ADDRESS	4	mcaanmmp	always nmmpointer	
160	(A0)	ADDRESS	4	mcaqueptr	current queue elem ptr	

TWS/ESA-Serviceroutinenadressen folgen

164	(A4)	ADDRESS	4	mcamsgx	message routine address
-----	------	---------	---	---------	-------------------------

Offsets						
168	(A8)	ADDRESS	4	mcaseqx		seq i/o service routine
172	(AC)	ADDRESS	4	mcapdsx		pds i/o service routine
176	(B0)	ADDRESS	4	mcaprmx		param member parse rtn
180	(B4)	ADDRESS	4	mcaquex		queue server routine
184	(B8)	ADDRESS	4	mcanowx		current time routine
188	(BC)	ADDRESS	4	mcavsam		addr vsam file handler
192	(C0)	ADDRESS	4	mcavsamb		addr bex vsam file handler
196	(C4)	ADDRESS	4	mcasubx		addr job submit routine
200	(C8)	ADDRESS	4	mcarelx		addr job release routine
204	(CC)	ADDRESS	4	mcaevhx		addr of event handler
208	(D0)	ADDRESS	4	mcamcpx		addr of modify curr plan
212	(D4)	ADDRESS	4	mcabexp		bex services address
216	(D8)	ADDRESS	4	mcaaidx		ZNOWX format clone address
220	(DC)	ADDRESS	4	mcalvck		level check routine adress
224	(E0)	ADDRESS	4	mcaznqd		ZNQDX lock dsname
228	(E4)	ADDRESS	4	mcaettp		ETT info for CP04
232	(E8)	ADDRESS	4	mcaIDecb		EW ID event ECB

verschiedene IBM Workload Scheduler-Konstanten

236	(EC)	SIGNED	2	mcansubs		number of subsys subtasks
238	(EE)	SIGNED	2	mcagmtof		gmt offset, minutes
240	(F0)	SIGNED	2	mcaracrtrc		racroute trace level
242	(F2)	BITSTRING	2	mcaDSTORE		Data Store task status
		1...		DBAReady		Data base init OK
		.1..		DBAFail		Data base ended
		..1.		JQUReady		Jes queue init OK
		...1		JQUFail		Jes queue ended
	 1..		mcaRefrCP		
242	(F2)	BITSTRING	1	*		
243	(F3)1..		Fprocin		

Offsets

	1.		Fmethod	Data Store method trace
	1		FParser	Data Store parser trace
244	(F4)	SIGNED	4	mcagmtSEC	gmt offset, seconds
248	(F8)	CHARACTER	1	mcajes	primary jes, A=js2, B=js3
249	(F9)	CHARACTER	1	mcajes	jes command 1st character
250	(FA)	CHARACTER	4	mcassnm	TWS/ESA subsystem name
254	(FE)	CHARACTER	8	mcamajnm	TWS/ESA major enq name
262	(106)	CHARACTER	8	mcaclass	racf resource class name
270	(10E)	CHARACTER	8	mcanjenm	nje node name
278	(116)	CHARACTER	4	mcaqfqm	name of full queue
282	(11A)	CHARACTER	8	mcanvid	Netview Receiver ID
290	(122)	CHARACTER	1	mcadsclas	JES class for Datastore
291	(123)	CHARACTER	1	mcaSPIN	Y = SPIN available; N = SPIN not available to server block
292	(124)	ADDRESS	4	mcaphbp	
296	(128)	CHARACTER	5	mcaclnjob	clean up job name
301	(12D)	CHARACTER	1	mcaddrspc	address space type: O = controller/tracker; S = server; D = data store; B = batch; T = Trial EQQDTTOP; L = Batch Loader
302	(12E)	CHARACTER	8	MCAJesfmid	JES fmid
310	(136)	CHARACTER	1	MCAtraces	
		1...		MCAzzSPIN	SPIN traces
		.111 1111		*	
311	(137)	CHARACTER	1	mcallopt	LISTLOG option A!F!N
312	(138)	BITSTRING	6	mcasubrs	protected subresources
		1...		mcaadnm	ada.adname is a resource
		.1..		mcaadow	ado.owner is a resource
		..1.		mcaadgr	adg.group is a resource

Offsets

		...1	mcaadjb	adj.jobname is a resource
	 1...	mcacpad	cpa.adname is a resource
	1..	mcacpow	cpo.owner is a resource
	1.	mcacpgr	cpg.group is a resource
	1	mcacpjb	cpj.jobname is a resource
313	(139)	1...	mcacpws	cpw.wsname is a resource
		.1..	mcajcad	jsa.adname is a resource
		..1.	mcajcjb	jsj.jobname is a resource
		...1	mcajcws	jsw.wsname is a resource
	 1...	mcajcow	jso.owner is a resource
	1..	mcajcgr	jsg.group is a resource
	1.	mcaltad	lta.adname is a resource
	1	mcaltow	lto.owner is a resource
314	(13A)	1...	mcaoiad	oia.adname is a resource
		.1..	mcawsws	wsw.wsname is a resource
		..1.	mcarlad	rla.adname is a resource
		...1	mcarlow	rlo.owner is a resource
	 1...	mcarlgr	rlg.group is a resource
	1..	mcarlws	rlw.wsname is a resource
	1.	mcaclcn	clc.calname is a resource
	1	mcaprpn	prp.pername is a resource
315	(13B)	1...	mcaetnm	ete.name is a resource
		.1..	mcaetad	eta.name is a resource
		..1.	mcasrnm	srs.name is a resource

Offsets

		...1		mcavjvo	jv.owner is a resource
	 1...		mcavjvn	jv.tabname is a resource
	1..		mcapwo	cpz.wsname is a res
	1.		mcapgd	cpd.OCCgrp is a res
	1		mcaltgd	ltd.OCCgrp is a res
316	(13C)	1...		mcaadgd	add.adgrp is a res
		.1..		mcarlwst	rl.wsstat is a res
		..1.		mcardrn	rdr.name is a res
		...1		mcaadex	ade.extname is a res
	 1...		mcapex	cpe.extname is a res
	1..		mcaadse	ad.secelem is a res
	1.		mcapse	cp.secelem is a res
	1		mcadbrp	rp.reptype is a res
317	(13D)	1...		mcaADinuse	AD used by batch
		.1..		mcaADVERrun	AD VER done
		..1.		mcapif	
		...1 1111		*	
318	(13E)	BITSTRING	2	mcaflags2	flags
		1...		mcasp52	1: mvs/sp 5.2 or later
		.1..		mcasymb	1: perform symbol subst
		..1.		mcaux002	exit2 invoked
		...1		mcaawaenq	deq after abend?
	 1...		mcadbg	for debug purpose
	1..		mcajtblock	ON: JTB is locked
	1.		mcaTWSCntlStart	On at controller StartUp
	1		mcaBulkDiscovery	bulc disc is already running
319	(13F)	1...		mcalock2b	
		.1..		mcastopc	On = stop command issued
		..11 1111		*	free
320	(140)	ADDRESS	4	mcatplgyp	Topology CB address
324	(144)	ADDRESS	4	mcaSCLIBdcb	sclib dcb ptr
324	(144)	SIGNED	4	mcaHTDStlastrec	
328	(148)	ADDRESS	4	mcaCtoken	token for C environment
328	(148)	SIGNED	4	mcaHTDStlastcyc	

Offsets						
332	(14C)	SIGNED	4	mcauserf		reserved for tws exits
336	(150)	SIGNED	2	mcaquelen		QUEUELEN changed value
338	(152)	BITSTRING	2	mcaperf		Performance flags
		1... ..		mcaexiDB		EXIT debug
		.1.. ..		mcajclDB		JCL debug
		..1.		mcaE105		E105 msg flag
		...1		mcaZ308		Z308 msg flag
	 1...		mcan069		N069 msg flag
338	(152)	BITSTRING	1	*		free
340	(154)	SIGNED	2	mcaTimeSta		stats msg interval time
342	(156)	SIGNED	2	mcablrc		BL rc with EQQY221E
344	(158)	ADDRESS	4	MCADBGp		
348	(15C)	ADDRESS	4	mcamlogd		mlog dsname address

2 mcafarb ptr(31) , farb ptr
2 lockrc bin(31) ,

352	(160)	UNSIGNED	2	mcaHT_evtseq		
354	(162)	UNSIGNED	2	*		
356	(164)	SIGNED	4	mcaHT_evtchkcyc		
360	(168)	CHARACTER	4	mcaFINDmem		
360	(168)	BITSTRING	3	mcaTTR		
363	(16B)	BITSTRING	1	mcaconc		
364	(16C)	SIGNED	4	mcaENFTOK57		ENFREQ 57 dtoken
364	(16C)	SIGNED	4	mcaHTDSespN		
368	(170)	SIGNED	4	mcaWLMQsz		WLM query size
368	(170)	SIGNED	4	mcaHTDSespT		
372	(174)	SIGNED	4	mcaENFTOK41		ENFREQ 41 dtoken
376	(178)	ADDRESS	4	mcamsgh		bufmsg routine address
380	(17C)	ADDRESS	4	mcaEXTp		address of extended MCA
384	(180)	SIGNED	4	mcaENFTOK53		ENFREQ 53 dtoken
388	(184)	ADDRESS	4	mcaux014		address of eqqux014
392	(188)	CHARACTER		mcaend		end of mca

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	324	mcaEXT	extended MCA	

Offsets					
0	(0)	ADDRESS	4	mcaANCQp	adress of mcaFLque
4	(4)	CHARACTER	8	mcaSYSNAME	&SYSNAME
12	(C)	ADDRESS	4	mcajtab	address of EQQZJTAB
16	(10)	ADDRESS	4	mcadsiox	address of EQQDSIOX
20	(14)	ADDRESS	4	mcadsini	address of EQQDSINI
24	(18)	ADDRESS	4	mcajtbp	ptr to JTB
28	(1C)	ADDRESS	4	mcaX14tabp	exit14 tabptr
32	(20)	SIGNED	4	mcaX14numr	exit14 numrow
36	(24)	SIGNED	4	mcaX14rsiz	exit14 recsize
40	(28)	ADDRESS	4	mcaDSViewp	ptr to DSV command area
44	(2C)	CHARACTER	8	mcaoptmem	options member
52	(34)	ADDRESS	4	mcadtbaux	pointer to refresh dest
56	(38)	SIGNED	4	mcaavildst	destination slots available for refresh
60	(3C)	ADDRESS	4	mcahtca	pointer to htca
64	(40)	SIGNED	4	mcahtcauxn	len of aux htc
68	(44)	ADDRESS	4	mcahtcaux	pointer to auxiliary htc
72	(48)	SIGNED	4	mcahtsauxn	len of aux hts
76	(4C)	ADDRESS	4	mcahtsaux	pointer to auxiliary hts
80	(50)	ADDRESS	4	mcahtsa	pointer to htca
84	(54)	ADDRESS	4	mcahtcp	ptr to HTC block
88	(58)	SIGNED	4	mcahtcl	length of HTC block
92	(5C)	ADDRESS	4	mcahtsp	ptr to HTS block
96	(60)	SIGNED	4	mcahtsl	length of HTS block
100	(64)	CHARACTER	2	*	free
102	(66)	UNSIGNED	4	mcasseqconf	last sseq confirmed
104	(68)	SIGNED	4	mcareconf	last HTDS record confirmed
108	(6C)	SIGNED	4	mcacyconf	last HTDS cycle confirmed
112	(70)	ADDRESS	4	*(52)	free
320	(140)	ADDRESS	4	mcaHTdbfP	free

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	12	mcaFLque	
0	(0)	ADDRESS	4	mcaANCQp1	floptmsgqu1

Offsets

4	(4)	ADDRESS	4	mcaANCQp2	floptmsgqu2
8	(8)	ADDRESS	4	mcaANCQpT	floptmsgqu2

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	28	mcaDSView	
0	(0)	CHARACTER	16	mcaDSV_AD	adid
16	(10)	CHARACTER	10	mcaDSV_IA	ia
26	(1A)	SIGNED	2	mcaDSV_OP	opnum

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)
0	(0)	STRUCTURE	*	mcaHTpage
0	(0)	CHARACTER	12	mcaHTfix
0	(0)	ADDRESS	4	mcaHTnext
4	(4)	ADDRESS	4	mcaHTtot
8	(8)	SIGNED	4	mcaHToff
12	(C)	CHARACTER	*	mcaHTend

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
DBAFail	F2	40	3
DBAReady	F2	80	3
Fmethod	F3	02	3
FParse	F3	01	3
Fprocin	F3	04	3
JQUFail	F2	10	3
JQUReady	F2	20	3
mca	0		1
mcaacee	E	20	3
mcaadex	13C	10	3
mcaadgd	13C	80	3
mcaadgr	138	20	3
mcaADinuse	13D	80	3
mcaadjb	138	10	3
mcaadnm	138	80	3
mcaadow	138	40	3
mcaadse	13C	04	3
mcaADVERrun	13D	40	3
mcaaidx	D8		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
mcaANCQp	0		2
mcaANCQpT	8		2
mcaANCQp1	0		2
mcaANCQp2	4		2
mcaanmmp	9C		2
mcaascb	10		2
mcaasip	6C		2
mcaasxb	E	40	3
mcaavildst	38		2
mcabexp	D4		2
mcablrc	156		2
mcaBulkDiscoveryRunning	13E	01	3
mcacjes	F9		2
mcaclass	106		2
mcacln	13A	02	3
mcaclnjob	128		2
mcaconc	16B		3
mcacpad	138	08	3
mcacpex	13C	08	3
mcacpgd	13B	02	3
mcacpgr	138	02	3
mcacpjb	138	01	3
mcacpnq	58		2
mcacpow	138	04	3
mcacpse	13C	02	3
mcacpwo	13B	04	3
mcacpws	139	80	3
mcaCtoken	148		2
mcacyconf	6C		2
mcadbg	13E	08	3
MCADBGp	158		2
mcadbrp	13C	01	3
mcaddrspc	12D		2
mcadesc	0		2
mcadiap	54		2
mcadsclas	122		2
mcadsini	14		2
mcadsiox	10		2
mcaDSTORE	F2		2
mcaDSV_AD	0		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
mcaDSV_IA	10		2
mcaDSV_OP	1A		2
mcaDSView	0		1
mcaDSViewp	28		2
mcadtbaux	34		2
mcaedpb	40		2
mcaedpw	F	20	3
mcaend	188		2
mcaENFTOK41	174		2
mcaENFTOK53	180		2
mcaENFTOK57	16C		2
mcaesa	E	08	3
mcaetad	13B	40	3
mcaetnm	13B	80	3
mcaettp	E4		2
mcaevhx	CC		2
mcaexiDB	152	80	3
mcaEXT	0		1
mcaEXTp	17C		2
mcaE105	152	20	3
mcaFINDmem	168		2
mcaflags	E		2
mcaflags2	13E		2
mcaFLque	0		1
mcafmid	6		3
mcaFSRAp	78		2
mcaftab	1C		2
mcagmtof	EE		2
mcagmtSEC	F4		2
mcahcm	90		2
mcaHT_evtchkyc	164		2
mcaHT_evtseq	160		2
mcahtca	3C		2
mcahtcaux	44		2
mcahtcauxn	40		2
mcahtcl	58		2
mcahtcp	54		2
mcaHTdbfP	140		2
mcaHTDSespN	16C		3
mcaHTDSespT	170		3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
mcaHTDslastcyc	148		3
mcaHTDslastrec	144		3
mcaHTend	C		2
mcaHTfix	0		2
mcaHTnext	0		3
mcaHToff	8		3
mcaHTpage	0		1
mcahtsa	50		2
mcahtsaux	4C		2
mcahtsauxn	48		2
mcahtsl	60		2
mcahtsp	5C		2
mcaHTtot	4		3
mcaID	94		2
mcaIDecb	E8		2
mcajancp	60		2
mcajcad	139	40	3
mcajcg	139	04	3
mcajcb	139	20	3
mcajclDB	152	40	3
mcajcow	139	08	3
mcajcws	139	10	3
mcajes	F8		2
MCAJesfmid	12E		2
mcajltbl	88		2
mcaJopts	98		2
mcajtab	C		2
mcajtblock	13E	04	3
mcajtbp	18		2
mcalevel	4		2
mcallopt	137		2
mcallock2b	13F	80	3
mcaltad	139	02	3
mcaltgd	13B	01	3
mcaltow	139	01	3
mcalvck	DC		2
mcamajnm	FE		2
mcamcpx	D0		2
mcamlog	E	02	3
mcamlogd	15C		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
mcamsggh	178		2
mcamsgx	A4		2
mcamtcb	14		2
mcanabb	50		2
mcanjenm	10E		2
mcanmmb	3C		2
mcanoprm	F	40	3
mcanowx	B8		2
mcansubs	EC		2
mcanvid	11A		2
mcan069	152	08	3
mcaoiaid	13A	80	3
mcaopecb	2C		2
mcaoptmem	2C		2
mcaopts	18		2
mcapace	F	08	3
mcapdsx	AC		2
mcaperf	152		2
mcaphbp	124		2
mcapif	13D	20	3
mcaprip	44		2
mcaprmx	B0		2
mcaprpn	13A	01	3
mcaqfcan	F	80	3
mcaqfecb	68		2
mcaqfqn	116		2
mcaquelen	150		2
mcaqueptr	A0		2
mcaquex	B4		2
mcaracf	E	10	3
mcaractrc	F0		2
mcardrn	13C	20	3
mcareconf	68		2
mcaRefrCP	F2	08	3
mcarelx	C8		2
mcarlad	13A	20	3
mcarlgr	13A	08	3
mcarlow	13A	10	3
mcarlws	13A	04	3
mcarlwst	13C	40	3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
mcarodmopt	8C		2
mcarsip	84		2
mcartrq	F	02	3
mcaSCLIBdcb	144		2
mcaseqx	A8		2
mcaSPIN	123		2
mcasp4	E	04	3
mcasp422	F	10	3
mcasp52	13E	80	3
mcasrap	48		2
mcasrnm	13B	20	3
mcassct	34		2
mcasseqconf	66		2
mcassnm	FA		2
mcassvt	30		2
mcassxp	80		2
mcastegs	28		2
mcastemj	20		2
mcastenm	24		2
mcastopc	13F	40	3
mcasub	7C		2
mcasubrs	138		2
mcasubx	C4		2
mcasure	4C		2
mcasymb	13E	40	3
mcaSyncEcbPtr	70		2
mcaSYSNAME	4		2
mcatcpr	F	01	3
mcaTimeSta	154		2
mcatmlog	5C		2
mcatplgyp	140		2
MCAtraces	136		2
mcatso	E	80	3
mcatsob	38		2
mcaTSRAp	74		2
mcaTTR	168		3
mcaTWSCntlStart	13E	02	3
mcaupace	F	04	3
mcauserf	14C		2
mcaux002	13E	20	3

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
mcaux014	184		2
mcaver	4		3
mcavjvn	13B	08	3
mcavjvo	13B	10	3
mcavsam	BC		2
mcavsamb	C0		2
mcawaenq	13E	10	3
mcaWLMQsz	170		2
mcawsws	13A	40	3
mcaxsip	64		2
mcaX14numr	20		2
mcaX14rsiz	24		2
mcaX14tabp	1C		2
mcaznqd	E0		2
MCAzzSPIN	136	80	3
mcaZ308	152	10	3
mca313	E	01	3

TJCB - JCL-Anpassungssteuerblock

Name : DCLTJCB

Funktion:

Dies ist der JCL-Anpassungssteuerblock des JS-Handlers. Er enthält Informationen zu allen eingebetteten JCLs.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	92	tjcb	tailoring jcl control block
0	(0)	CHARACTER	4	tjcbdesc	descriptor always 'tjcb'
4	(4)	CHARACTER	2	tjcbvers	block version
6	(6)	CHARACTER	2	*	not used
8	(8)	SIGNED	4	tjcbclin	number of lines in this jcl block
12	(C)	SIGNED	4	tjcbclin	current jcl line being proc
16	(10)	ADDRESS	4	tjcbtubp	pointer to tub block

Offsets						
20	(14)	ADDRESS	4	tjcbstgp	pointer to storage allocated for jcl , or null if it is the first tjob in chain, it also points to the common part of jcl record	
24	(18)	SIGNED	4	tjcbfst	from start of orig/fetched JCL (the value is number of lines)	
28	(1C)	ADDRESS	4	tjcbexp	pointer to expansion work area	
32	(20)	SIGNED	4	tjcbexps	size of exp. work area	
36	(24)	SIGNED	4	tjcbamnt	amount of storage getmained for jcl pointed to by this tjob	
40	(28)	ADDRESS	4	tjcbjclp	pointer to first jcl line	
44	(2C)	ADDRESS	4	tjcbnxtp	pointer to next tjob block	
48	(30)	ADDRESS	4	tjcbprep	pointer to previous tjob block	
52	(34)	SIGNED	4	tjcbnslv	nesting level of current tjob	
56	(38)	CHARACTER	1	tjcbtype	type of data pointed to block J= original JCL M= fetched nominated member X= fetched by user exit	
57	(39)	CHARACTER	1	tjcbactn	action Y= include JCL N= not included / exclude	
58	(3A)	CHARACTER	8	tjcbli	ddname of jcl library	
66	(42)	CHARACTER	8	tjcbmem	member of jcl library	
74	(4A)	CHARACTER	8	tjcbusrx	user exit name	
82	(52)	CHARACTER	1	tjcbfin	data for this block processed	
83	(53)	CHARACTER	1	tjcbstrm	JCL stream switches at start of JCL governed by this tjob values as in prbsw1	

Offsets					
84	(54)	CHARACTER	4	tjcbint	directive introducing block command,main keyword, domain
88	(58)	CHARACTER	1	tjcbsta	saved values for stream sw
89	(59)	CHARACTER	3	*	reserved values as in prsk \$LBC
92	(5C)	CHARACTER		tjcbend	end label of block

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
tjcb	0		1
tjcbclin	C		2
tjcbclin	8		2
tjcbactn	39		2
tjcbamnt	24		2
tjcbdesc	0		2
tjcbend	5C		2
tjcbexpp	1C		2
tjcbexps	20		2
tjcbfin	52		2
tjcbint	54		2
tjcbjclp	28		2
tjcblib	3A		2
tjcbmem	42		2
tjcbnslv	34		2
tjcbnxtp	2C		2
tjcbfst	18		2
tjcbprep	30		2
tjcbstgp	14		2
tjcbstrm	53		2
tjcbsta	58		2
tjcbtubp	10		2
tjcbtype	38		2
tjcbusrx	4A		2
tjcbvers	4		2

TUB - JCL-Anpassungssteuerblock

Name : DCLTUB

Funktion:
Dies ist der JCL-Anpassungssteuerblock des JS-Handlers.
Er enthält Informationen zu allen eingebetteten JCLs.

Offsets					
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	800	tub	tailoring user block
0	(0)	CHARACTER	4	tubdesc	descriptor always 'TUB '
4	(4)	CHARACTER	2	tubvers	block version
6	(6)	CHARACTER	8	tuboreqd	original data type
14	(E)	CHARACTER	8	tubident	tailoring identifier
14	(E)	CHARACTER	8	tubuser	..tso; userid
22	(16)	CHARACTER	28	tubopid	operation identifier
22	(16)	CHARACTER	16	tubadid	..application; id
38	(26)	CHARACTER	6	tubadiad	..input; arrival day
44	(2C)	CHARACTER	4	tubadiat	..input; arrival time
48	(30)	SIGNED	2	tubopno	..operation; number
50	(32)	CHARACTER	1	tubSimulate	time simulation Y or N
51	(33)	CHARACTER	1	*	not used
52	(34)	ADDRESS	4	tubocp	addr of PIF format CP occ
56	(38)	ADDRESS	4	tubopp	addr of PIF format CP opr
60	(3C)	ADDRESS	4	tubwsp	address of PIF format CP WS
64	(40)	SIGNED	4	tubocl	length of PIF format CP occ
68	(44)	SIGNED	4	tubopl	length of PIF format CP opr
72	(48)	SIGNED	4	tubwsl	length of PIF format CP WS
76	(4C)	ADDRESS	4	tubdcbp	current jcl library dcb
80	(50)	ADDRESS	4	tubbufp	address of jblib buffer
84	(54)	SIGNED	4	tubasubp	subpool for subseq allocs
88	(58)	ADDRESS	4	tubworkp	ptr to tailoring work area
92	(5C)	ADDRESS	4	tubmcap	pointer to mca

Offsets						
96	(60)	SIGNED	4	tubworkl		length of tail work area
100	(64)	ADDRESS	4	tubjbcu		current tjcb
104	(68)	ADDRESS	4	tubjcbfp		pointer to first tjcb
108	(6C)	ADDRESS	4	tubjcblp		pointer to last tjcb
112	(70)	ADDRESS	4	tubjvptr		pointer to jv record buffer
116	(74)	ADDRESS	4	tubdatp		pointer predef variables and their values
120	(78)	ADDRESS	4	tubtvp		pointer to jcl vars found when searching jcl
124	(7C)	SIGNED	4	tubtvarl		length of allocated var stg
128	(80)	CHARACTER	2	tubtask		current task (GS,WA)
130	(82)	CHARACTER	8	tubjclib		current jcl library
138	(8A)	CHARACTER	1	tuballv		Y= variable proc. complete
139	(8B)	CHARACTER	1	*		not used
140	(8C)	CHARACTER	16	tubjvtab		occurrence variable table
156	(9C)	CHARACTER	16	tubsearch (16)		current table search order
412	(19C)	CHARACTER	16	tubTabName (16)		table names array \$CXWC
668	(29C)	ADDRESS	4	tubTabNameP (16)		table ptrs array \$CXWA
732	(2DC)	CHARACTER	8	tubfoot		current footprint
740	(2E4)	CHARACTER	2	tubdlm		current dlm
742	(2E6)	CHARACTER	1	*		reserved
743	(2E7)	BITSTRING	1	tubflags		flag byte
		1...		tubosi		osi operation JCL
		.1..		tubboj		osi operation JCL
		..11 1111		*		reserved
744	(2E8)	SIGNED	4	tubosi		number of lines inserted
748	(2EC)	ADDRESS	4	tubjcfp		pointer to feedback area
752	(2F0)	ADDRESS	4	tubjdup		addr of user SETFORM date dates
756	(2F4)	ADDRESS	4	tubtvsp		addr of SAVEVAR variable in stg

Offsets						
760	(2F8)	ADDRESS	4	tubjdtp		addr of SETVAR variables
764	(2FC)	ADDRESS	4	tubysimp		addr of simulation parms
768	(300)	ADDRESS	4	tuboca		addr of PIF format occ alwaysOEA
772	(304)	ADDRESS	4	tubopa		addr of PIF format opr alwaysOEA
776	(308)	ADDRESS	4	tubwsa		address of PIF format ws alw.OEA
780	(30C)	ADDRESS	4	tubxinfp		Extended Job Info addr.
784	(310)	ADDRESS	4	tubDOA		DOA address needed to \$CAEC check DOAPSUJCL \$CAEA
788	(314)	ADDRESS	4	tubcp3c		CP occurrence
792	(318)	ADDRESS	4	tubcp3p		CP operation record
796	(31C)	ADDRESS	4	tubcp3r		CP op record user fields
800	(320)	CHARACTER		tubend		end of tub block

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)		Beschreibung
0	(0)	STRUCTURE	200	tubxinf		
0	(0)	CHARACTER	54	tubxjnm		
54	(36)	CHARACTER	146	*		

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
tub	0		1
tubosi	2E8		2
tubadiad	26		3
tubadiat	2C		3
tubadid	16		3
tuballv	8A		2
tubasubp	54		2
tubboj	2E7	40	3
tubbufp	50		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
tubcp3c	314		2
tubcp3p	318		2
tubcp3r	31C		2
tubdatp	74		2
tubdcbp	4C		2
tubdesc	0		2
tubdlm	2E4		2
tubDOA	310		2
tubend	320		2
tubflags	2E7		2
tubfoot	2DC		2
tubident	E		2
tubjbcu	64		2
tubjcbfp	68		2
tubjcblp	6C		2
tubjcfp	2EC		2
tubjclib	82		2
tubjdtp	2F8		2
tubjdup	2F0		2
tubjvptr	70		2
tubjvtab	8C		2
tubmcap	5C		2
tuboca	300		2
tubocl	40		2
tubocp	34		2
tubopa	304		2
tubopid	16		2
tubopl	44		2
tubopno	30		3
tubopp	38		2
tuboreqd	6		2
tubosi	2E7	80	3
tubsearch	9C		2
tubSimulate	32		2
tubTabName	19C		2
tubTabNameP	29C		2
tubtask	80		2
tubtvarl	7C		2
tubtvp	78		2
tubtvsp	2F4		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
tubuser	E		3
tubvers	4		2
tubworkl	60		2
tubworkp	58		2
tubwsa	308		2
tubwsl	48		2
tubwsp	3C		2
tubxinf	0		1
tubxinfp	30C		2
tubxjnm	0		2
tubysimp	2FC		2

TV - Satzbeschreibung für JCL-Variablentabelle

Name : DCLTV

Funktion:

Definiert das Layout aller nicht zuvor festgelegten Variablen, die im Jobstrom gefunden werden. Die Informationen im Hauptabschnitt des Satzes und die Abhängigkeitsinformationen werden unverändert aus der entsprechenden Variablenbeschreibung im entsprechenden Satz der JCL-Variablentabelle vom Typ 3 übernommen.

HINWEIS: Dieser Block wird durch 'getmain'-Operationen auf das nächste Vielfache von 32K erweitert, wenn zu wenig Speicherplatz für den nächsten Eintrag vorhanden ist. Die aktuelle Adresse und die durch 'getmain' erstellte Länge werden im Block TUB behalten.

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	
0	(0)	STRUCTURE	16	tv	JCL variable table	
0	(0)	CHARACTER	4	tveye	descriptor always 'tvar'	
4	(4)	CHARACTER	2	tvvers	block version	
6	(6)	CHARACTER	2	*	not used	
8	(8)	SIGNED	4	tvvars	number of variables in storage	
12	(C)	SIGNED	4	tvnoff	offset to next free byte	
16	(10)	CHARACTER		tvdata	variable part of table	

Offsets						
Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung	

Offsets						
0	(0)	STRUCTURE	552	tvtab		
0	(0)	CHARACTER	88	tvrun		tailoring run-time info
0	(0)	CHARACTER	16	tvtable		source table name
16	(10)	CHARACTER	8	tvasusr		assigned by USER/EXIT/VAR
24	(18)	CHARACTER	44	tvasg		value assigned
68	(44)	CHARACTER	16	tvfrst		first occurrence
68	(44)	SIGNED	4	tvotjcb		address of source tjcb
72	(48)	SIGNED	4	tvoline		JCL line within tjcb scope
76	(4C)	SIGNED	2	tvovnr		seq of identification in line
78	(4E)	CHARACTER	6	*		reserved
84	(54)	CHARACTER	1	tvasgtyp		type of assignment P/E/V/D
85	(55)	CHARACTER	1	tvset		Y(es) ! N(o) ! D(elayed)
86	(56)	CHARACTER	1	tvusg		type of usage % / & /
87	(57)	CHARACTER	1	*		reserved
88	(58)	CHARACTER	464	tvfxd		unchanged from VSAM
88	(58)	CHARACTER	8	tvvar		JCL variable name
96	(60)	CHARACTER	44	tvdf1		JCL variable default value
140	(8C)	CHARACTER	1	tvuc		uppercae Y/N
141	(8D)	CHARACTER	1	tvstp		prompt / setup / submit
142	(8E)	SIGNED	2	tvlg		value length
144	(90)	CHARACTER	7	tvtyp		verification type
151	(97)	CHARACTER	8	tvex		substitution exit name
159	(9F)	CHARACTER	1	tvinp		input required
160	(A0)	SIGNED	2	tvpos		replace position JCL data
162	(A2)	CHARACTER	1	tvnum		numeric
163	(A3)	CHARACTER	2	tvcmp		comparison operator
165	(A5)	CHARACTER	44	tvpat		validation pattern
209	(D1)	CHARACTER	102	tvvld		valid values
311	(137)	CHARACTER	204	tvtxt		dialog text
515	(203)	CHARACTER	20	tvdes		description

Offsets

535	(217)	CHARACTER	1	*	reserved
536	(218)	SIGNED	2	tvnrp	number of dependent values
538	(21A)	CHARACTER	8	tvind	independent variable name
546	(222)	CHARACTER	2	*	reserved
548	(224)	CHARACTER	2	tvsubs	substring start pos
550	(226)	CHARACTER	2	tvsubl	substring length

Offsets

Dezimal	Hexadezimal	Typ	Länge	Name (Dimension)	Beschreibung
0	(0)	STRUCTURE	88	tvd(*)	
0	(0)	CHARACTER	44	tvdiv	value of independent
44	(2C)	CHARACTER	44	tvddv	value of dependent

Querverweis

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
tv	0		1
tvvars	8		2
tvsg	18		3
tvsgtyp	54		3
tvusr	10		3
tvcmp	A3		3
tvdata	0		1
tvdata	10		2
tvddv	2C		2
tvdes	203		3
tvdf	60		3
tvdiv	0		2
tvex	97		3
tveye	0		2
tvfrst	44		3
tvfxd	58		2
tvind	21A		3
tvinp	9F		3
tvlg	8E		3
tvnoff	C		2

Name	Hexadezimaler Offset	Hexadezimalwert	Ebene
tvnrp	218		3
tvnum	A2		3
tvoline	48		4
tvotjcb	44		4
tvovnr	4C		4
tvpat	A5		3
tvpos	A0		3
tvrun	0		2
tvset	55		3
tvstp	8D		3
tvsubl	226		3
tvsubs	224		3
tvtab	0		1
tvtable	0		3
tvtxt	137		3
tvtyp	90		3
tvuc	8C		3
tvusg	56		3
tvvar	58		3
tvvers	4		2
tvvld	D1		3

Nachrichten

Vom Agenten für z/OS ausgegebene Nachrichten.

Alle Nachrichten, die der Agent für z/OS ausgibt, sind im Handbuch *IBM Workload Automation: Messages and Codes* beschrieben.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. IBM stellt dieses Material möglicherweise auch in anderen Sprachen zur Verfügung. Für den Zugriff auf das Material in einer anderen Sprache kann eine Kopie des Produkts oder der Produktversion in der jeweiligen Sprache erforderlich sein.

Möglicherweise bietet IBM die in diesem Dokument beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für die in diesem Dokument beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

*IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France*

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
USA*

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in dieser Veröffentlichung beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die in diesem Dokument enthaltenen Leistungsdaten wurden unter bestimmten Betriebsbedingungen erzielt. Die tatsächlichen Ergebnisse können davon abweichen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Personen und Unternehmen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

© (Name Ihrer Firma) (Jahr).

Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet.

© Copyright IBM Corp. _Jahr/Jahre angeben_.

Marken

IBM, das IBM Logo und ibm.com sind eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

IT Infrastructure Library ist eine eingetragene Marke von AXELOS Limited.

Linear Tape-Open, LTO, das LTO-Logo, Ultrium und das Ultrium-Logo sind Marken von HP, der IBM Corporation und von Quantum in den USA und/oder anderen Ländern.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder ihrer Tochtergesellschaften in den USA oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.



Java™ und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Cell Broadband Engine wird unter Lizenz verwendet und ist eine Marke der Sony Computer Entertainment, Inc. in den USA und/oder anderen Ländern.

ITIL ist eine eingetragene Marke von AXELOS Limited.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Bedingungen für Produktdokumentation

Die Berechtigungen zur Nutzung dieser Veröffentlichungen werden Ihnen auf der Basis der folgenden Bedingungen gewährt.

Anwendbarkeit

Diese Bedingungen sind eine Ergänzung der Nutzungsbedingungen auf der IBM Website.

Persönliche Nutzung

Sie dürfen diese Veröffentlichungen für Ihre persönliche, nicht kommerzielle Nutzung unter der Voraussetzung vervielfältigen, dass alle Eigentumsvermerke erhalten bleiben. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM nicht weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Kommerzielle Nutzung

Sie dürfen diese Veröffentlichungen nur innerhalb Ihres Unternehmens und unter der Voraussetzung, dass alle Eigentumsvermerke erhalten bleiben, vervielfältigen, weitergeben und anzeigen. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM außerhalb Ihres Unternehmens weder vervielfältigen, weitergeben oder anzeigen noch abgeleitete Werke davon erstellen.

Rechte

Abgesehen von den hier gewährten Berechtigungen werden keine weiteren Berechtigungen, Lizenzen oder Rechte (veröffentlicht oder stillschweigend) in Bezug auf die Veröffentlichungen oder darin enthaltene Informationen, Daten, Software oder geistiges Eigentum gewährt.

IBM behält sich das Recht vor, die hierin gewährten Berechtigungen nach eigenem Ermessen zurückzuziehen, wenn sich die Nutzung der Veröffentlichungen für IBM als nachteilig erweist oder wenn die obigen Nutzungsbestimmungen nicht genau befolgt werden.

Sie dürfen diese Informationen nur in Übereinstimmung mit allen anwendbaren Gesetzen und Verordnungen, einschließlich aller US-amerikanischen Exportgesetze und Verordnungen, herunterladen und exportieren.

IBM übernimmt keine Gewährleistung für den Inhalt dieser Veröffentlichungen. Diese Veröffentlichungen werden auf der Grundlage des gegenwärtigen Zustands (auf "as-is"-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung für die Handelsüblichkeit, die Verwendungsfähigkeit für einen bestimmten Zweck oder die Freiheit von Rechten Dritter zur Verfügung gestellt.

Index

A

Agent abbrechen 103
Agent ändern 104
Agent für z/OS
 Adressraum
 JCL-Prozedurdefinition 25
 Stapeljobdefinition 25
Angabe von Hostnamen und Port 35
Bibliotheken 5
Dateien
 Diagnose 9
 Ereignis 9
 Nachrichtenbibliothek 9
 Nachrichtenprotokoll 9
 Parameterbibliothek 9
herunterladen 4
Laufzeitoptionen 38
Standardinitialisierungsanweisungen 10
STC-Name 18
Subsystemname 18
Workstationname 27
Agent für z/OS, erforderliche Dateien 26
Agent für z/OS, Job 52
Agent für z/OS, optionale Dateien 26
Agent für z/OS, Workstation 52, 53
Agent starten 103
Agent stoppen 103
Agent stoppen und erneut starten 105
Agentenkonfiguration aktualisieren 105
Agentenneustart 107
Anweisungen 84
APARs
 PI09318 62
 PI35949 36
application, Schlüsselwort 52
Aspekte für Zuordnung
 Jobbibliothek (EELJBLIB) 32
Ausgabeereignisse 29
Ausgabemanager 69

B

Befehle
 nicht verfügbar 97
BEGIN, JCL-Anweisung 91
Behindertengerechte Bedienung ix
Beispiel für gestartete Task
 Parameter 25
 Prozedur 25
Beispielexits
 Abschluss des Jobs und des Abschnitts 13
 JCT-Ein-/Ausgabeexit für JES2 bereinigen 13
 JES2, QMOD-Phasenänderung 13
 Jobausgebende 13
 Jobstart 13
Benutzerabbruch 100

Bereinigungsereignisse 29
Betriebssystemtyp 52

C

cancel 2
CANCEL (Befehl) 103
CDATE, JCL-Variable 83
CDAY, JCL-Variable 82
CDD, JCL-Variable 82
CDDD, JCL-Variable 82
CDDMMYY, JCL-Variable 82
CHH, JCL-Variable 82
CHHMM, JCL-Variable 82
CHHMMSS, JCL-Variable 82
CHHMMSSX, JCL-Variable 82
Cloud & Smarter Infrastructure, technische Schulung ix
CMM, JCL-Variable 82
CMMYY, JCL-Variable 82
CODEPAGE, Parameter 62
Codepagedefinition 38
Codieren von JCL-Variablen 76
COMP, Schlüsselwort in JCL-Anweisungen 94
Computername 51
Conman-Befehle 97
CTIME, JCL-Variable 83
CWW, JCL-Variable 82
CWWJ, JCL-Variable 82
CYMD, JCL-Variable 82
CYY, JCL-Variable 82
CYYDDD, JCL-Variable 82
CYYMM, JCL-Variable 82
CYYMMDD, JCL-Variable 82
CYYYY, JCL-Variable 82
CYYYYMM, JCL-Variable 82

D

Dateiüberwachung, Dienstprogramm 62
Datenbereiche
 DCLDQE 112
 DCLESP 134
 DCLEVT 135
 DCLEXI 137
 DCLEXK 142
 DCLEXR 144
 DCLHTI 149
 DCLHTSA 151
 DCLJCFB 154
 DCLJCL 156
 DCLJCL1 162
 DCLJDA 165
 DCLJDT 170
 DCLJDU 172
 DCLJHS 177
 DCLJV 179
 DCLMCA 182
 DCLTJCB 197

Datenbereiche (*Forts.*)

 DCLTUB 199
 DCLTV 204
Datenrouter, Subtask 104
DB2-Transaktionsprotokoll voll 110
Dienstprogramme, Dateiüberwachung 62
Domänenmanager wechseln 105
Dynamic Workload Broker 64
 aktualisieren 105
 ändern 105
 Angabe von Hostnamen und Port 35
 synchronisieren 107
Dynamic Workload Broker-Sicherungsinstanzen 105
Dynamic Workload Console
 behindertengerechte Bedienung ix

E

EBCDIC-Format 61
ECSA-Speicher 19
EELBRDS, Datei 66
EELEVDs 101, 107
EELEXIT, Makro 13
EELHTDS 101, 107
EELHTREF 64, 101, 107
EELINST
 Beispiel-JCL 7
 Beispieljoberstellung 7
 Bibliotheken 6
 Dateierstellung 9
 starten 7
EELJBLIB 64, 65
EELJBLIB (Jobbibliotheksdatei)
 Aspekte für Zuordnung 32
EELUX002 (Exit zum Lesen in der Jobbibliothek) 44
Einfache Variablen 76
END, JCL-Anweisung 91
Endergebnisse für Jobabschnitt 29
Ereignisdatei 29
Ereignisdateien 101, 107
Ereignisfilterungsexit 48
Ereignisschreiber 29, 101
 Laufzeitoptionen 33
Ereignisschreiber, Subtask 104
Ereignisschreibertask 69
Ereignisse
 Fehlerbehebung 30
EWTROPTS 33
EXIT51 14
Exittrichtlinie 33
Exits
 Ereignisfilterung 48
 Lesen in der Jobbibliothek (EELUX002) 44
SMF
 Jobende und Schrittlende 21
 Jobstart 21
 Schreiben von Datensätzen 21

Exits (Forts.)
Start/Stop 43
Exits des Agenten für z/OS 43

F

Failovermechanismus 65
FETCH, JCL-Anweisung 93

G

Generierte Ereignisse 29
Gestartete Tasks 2
Globale Variablen-tabelle 80

H

HTTP
Verbindungsoptionen 35
HTTP-Client 105
HTTPOPTS 24, 35
HTTPOPTS, Initialisierungs-anwei-sung 27
SSLKEYRING 11

I

IBM Workload Scheduler for z/OS 2
SSL 25
SUBSYS STC 21
IBM Workload Scheduler-Variablen 69
IEFACTRT 21
IEFU83 21
IEFUJI 21
In Conman 97
Initialisierungsanweisungen 33
Ereignisschreiber 33
Exitrichtlinie 34
HTTP-Optionen 35
Laufzeitoptionen 38
Internes z/OS-Ereignis 97

J

JCL
Anpassung 2
Anweisung 59
Datei 58
Definition 2
Definitionsregeln 61
Member 58
Namensregeln 61
nationale Sonderzeichen 61
Position in z/OS 58
Programmierernamen 61
Variablensubstitution
Syntax 76
JCL-Anpassungsanweisungen 84
JCL-Fehler 99, 100
JES
interner Leser 98
JES2-Exits
Eingangspunkte 14
Installationsbefehle 14
Lademodule 14

JESEXIT7 14
Job
Definitionsschnittstellen 2
Definitionstyp 52
hohe Verfügbarkeit 67
Protokoll 2
Statusereignis 67
Übergabe 64
Job abgebrochen 100
Job bereinigt, Ereignis 99
Job blockiert 109
Job mit 'kill' beenden 97
Jobabschlussereignis 99
Jobabschlussereignisse 29
Jobanweisung 101
Jobausgabeereignis 99
Jobbeendigungs-codes 101
Jobbibliotheksdatei (EELJBLIB)
Aspekte für Zuordnung 32
Jobdefinition
Beispiel
Dynamic Workload Console 56
Composer 58
Beispiel 58, 59, 60
Dynamic Workload Console 53
JSDL 52
nach JCL-Definition 59
nach JCL-Verweis 58
Variablenuflösung 60
Jobendereignis 98
Jobendereignisse 29
Jobfehlercodes 99
Jobinstanzen verwalten 97
Jobinstanzverwaltung 97
Jobprotokollanforderungen 69
Jobprotokolle
anzeigen
in Conman 102
in Dynamic Workload Conso-
le 102
JOBRC 33, 101
Jobrückgabecodes 99
Jobstartereignis 98
Jobstartereignisse 29
Jobstatus
in Dynamic Workload Console 97
Jobstatusbeschreibung 97
Jobübergabeablauf 64
Jobübergabeereignisse 29
Jobvariablen
Definition
in JCL 69
in JSDL 69
JSDL 2

K

kill 2, 97

L

Laufzeitoptionen 33
Lesen in der Jobbibliothek, Exit (EE-
LUX002) 44
Lesereignis 98
Lesereignisse 29

M

maxdur, Jobstromschlüsselwort 97
maxcsa 19
MODIFY, Befehl 104

N

Nationale Sonderzeichen 61
Netzjobeintrag 50
NJE 50, 69

O

OADID, JCL-Variable 80
OCDATE, JCL-Variable 83
OCTIME, JCL-Variable 83
ODAY, JCL-Variable 80
ODD, JCL-Variable 80
ODDD, JCL-Variable 80
ODMY1, JCL-Variable 80
ODMY2, JCL-Variable 80
OHH, JCL-Variable 80
OHHMM, JCL-Variable 81
OMM, JCL-Variable 81
OMMY, JCL-Variable 81
OPIADATE, JCL-Variable 83
OPIATIME, JCL-Variable 83
OSSID, JCL-Variable 82
OWW, JCL-Variable 81
OWWD, JCL-Variable 81
OWWLAST, JCL-Variable 81
OWWMONTH, JCL-Variable 81
OYM, JCL-Variable 81
OYMD, JCL-Variable 81
OYMD1, JCL-Variable 81
OYMD2, JCL-Variable 81
OYMD3, JCL-Variable 81
OYY, JCL-Variable 81
OYYDDD, JCL-Variable 81
OYYMM, JCL-Variable 81
OYYYY, JCL-Variable 81

P

Planunterbrechung 110
Produktversion 109
Programmierernamen, Länge 61

R

RACF 23
Dateizugriffsschutz 24
rerun 2
Resynchronisationsnachrichten 108

S

SCAN, JCL-Anweisung 84
Schrittendereignis 98
Schulung ix
technisch ix
SETFORM, JCL-Anweisung 85
SETVAR, JCL-Anweisung 87
showjobs, Conman-Befehl 102

- Sicherungsdomänenmanager 105
- SMF-Exits 21
- SMP/E 4
- Sonderzeichen 61
- Speicherauszugsoptionen 22
- SSL 24
 - Sicherheitszertifikate 25
 - Standardzertifikate importieren 11
- SSL-Sicherheitsoptionen 35
- Standardvariablen-tabelle 80
- START, Befehl 103
- Start-/Stoppexit 43
- START, Subtask 104
- STOP, Befehl 103
- STOP, Subtask 104
- Subsystemdefinition
 - Beispiel 19
 - Datensatz 18
- Subsystemname 19, 51
- Subtasks
 - Datenrouter 104
 - Ereignisschreiber 104
 - Übergabe 104
- Sysplex 49
- SYSPLEX 69
- Systemabbruch 100
- Systembefehle 103
- Systemname 51

T

- TDWBHOSTNAME 106
- TDWBPORT 106
- Technische Schulung ix
- Tracker
 - Exits 2
- TWSOPTS 38
 - Codepage 62

U

- Übergabe, Subtask 104
- Übergabefehler 100
- Übergabetask 65

V

- Variablen
 - benutzerdefiniert, in JCL 79
 - in JCL 75
 - in JCL vordefiniert 80
 - datumsbezogen 82
 - jobbezogen 81
 - jobstrombezogen 80
 - mit dynamischem Format 82
 - temporär 83
- Variablenuflösung 69
- Variablen-tabelle
 - global 79
 - Standard 79
- VARSUB, Schlüsselwort in TWSOPTS 75
- Verbindungsbibliothek 22
- Verbindungsoptionen 33
- Verbundvariablen 76

W

- Workstationname 51
 - ändern 51
- Workstations des Agenten für z/OS auflisten 52
- Workstationstyp 52

Z

- z/OS 52
- z/OS-Befehle 103



Programmnummer: 5698-WSH

Gedruckt in Deutschland