

IBM Rational Developer for System z
Version 9.0.1

Konfiguration



IBM Rational Developer for System z
Version 9.0.1

Konfiguration



Hinweis

Vor Verwendung dieser Informationen sollten die allgemeinen Hinweise im Abschnitt „Bemerkungen zur Dokumentation für IBM Rational Developer for System z“ auf Seite 219 gelesen werden.

Elfte Ausgabe (Dezember 2013)

Diese Ausgabe bezieht sich auf IBM Rational Developer for System z Version 9.0.1 (Programmnummer 5724-T07) und - sofern in neuen Ausgaben nicht anders angegeben - auf alle nachfolgenden Releases und Modifikationen.

Veröffentlichungen können über den zuständigen IBM Ansprechpartner oder die zuständige IBM Geschäftsstelle bezogen werden. Veröffentlichungen sind nicht bei der unten angegebenen Adresse erhältlich.

IBM schätzt Ihre Kommentare. Wenn Sie Anmerkungen zu diesem Handbuch oder einer anderen Veröffentlichung haben, füllen Sie bitte das Antwortformular am Ende des Handbuchs aus.

Werden an IBM Informationen eingesandt, gewährt der Einsender IBM ein nicht ausschließliches Recht zur beliebigen Verwendung oder Verteilung dieser Informationen, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM Rational Developer for System z, Version 9.0.1, Configuration Guide,
IBM Form SC23-7658-10,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2000, 2013

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
Dezember 2013

Inhalt

Abbildungen	vii
-----------------------	-----

Tabellen	ix
--------------------	----

Zu diesem Handbuch.	xi
------------------------------------	-----------

Zielgruppe	xii
Zusammenfassung der Änderungen	xii
Beschreibung des Dokumentinhalts	xvi
Planung	xvi
Basisanpassung.	xvi
Common Access Repository Manager (optional)	xvi
SCLM Developer Toolkit (optional)	xvi
Application Deployment Manager (optional,	
veraltet)	xvii
Hostbasierte Codeanalyse (optional)	xvii
Weitere Anpassungstasks (optional).	xvii
Installationsprüfung.	xviii
Sicherheitsdefinitionen	xviii
Leitfaden für die Migration	xviii
Bedienerbefehle	xviii
Hostkonfigurationsreferenz	xviii

IBM Rational Developer for System z	
Hostkonfiguration.	1

Kapitel 1. Planung	3
-------------------------------------	----------

Aspekte der Migration	3
Hinweise zur Planung	3
Produktübersicht	3
Erforderliche Qualifikationen	4
Zeitbedarf	4
Hinweise zur Installationsvorbereitung	4
ID des Installationsbenutzers	5
Vorausgesetzte Produkte	5
Erforderliche Ressourcen	5
Hinweise zur Konfigurationsvorbereitung	10
Workload-Management	10
Ressourcennutzung und Systembegrenzungen.	10
Erforderliche Konfiguration für vorausgesetzte	
Produkte	10
Hinweise zur Benutzer-ID	10
Hinweise zum Server	11
Konfigurationsmethode	12
Hinweise zur Implementierungsvorbereitung	12
Clientprüfliste	14

Kapitel 2. Basisanpassung	17
--	-----------

Voraussetzungen und Prüfliste	17
Anpassungskonfiguration.	17
PARMLIB-Änderungen	19
z/OS UNIX-Grenzwerte in BPXPRMxx festlegen	19
Gestartete Tasks zu COMMNDxx hinzufügen	20
SVC-Definitionen in IEASVCxx.	21
LPA-Definitionen in LPALSTxx.	21

APF-Berechtigungen in PROGxx	22
LINKLIST-Definitionen in PROGxx	23
Vorausgesetzte LINKLIST- und LPA-Definitionen	24
LINKLIST-Definitionen für andere Produkte	26
PROCLIB-Änderungen	26
JMON, gestartete Task von JES Job Monitor	26
DBGMR, gestartete Task für den Debug-Mana-	
ger	27
RSED, gestartete Task für den RSE-Dämon	28
Einschränkungen der JCL für die PARM-Variable	29
TMPDIR-Verarbeitung	30
ELAXF*-Prozeduren für ferne Build-Erstellung	31
Sicherheitsdefinitionen.	33
FEJJCNFG - Konfigurationsdatei für JES Job Monitor	34
RSE-Konfigurationsdatei rsed.envvars	39
Verfügbaren Portbereich für den RSE-Server defi-	
nieren	49
Zusätzliche Java-Startparameter mit _RSE_JA-	
VAOPTS definieren.	51
Zusätzliche Java-Startparameter mit _RSE_ISPF-	
_OPTS definieren	59
ISPF.conf - Konfigurationsdatei für TSO/ISPF Client	
Gateway	59
Optionale Komponenten	61
Installationsprüfung	61

Kapitel 3. Common Access Repository	
Manager (optional)	63

Voraussetzungen und Prüfliste	63
Serverstartmethode und aktiven RAM auswählen	64
CARMA-Serverstart	64
CRASTART	64
Batchübergabe	64
TSO/ISPF-Client-Gateway (veraltet)	64
Produktions-RAM	65
CA Endevor SCM-RAM	65
RAM für CA Endevor SCM-Pakete	65
Muster-RAM	65
PDS-RAM	65
Skeleton-RAM	65
SCLM-RAM	65
Kombinationen aus vorkonfiguriertem RAM und	
Serverstart.	65
CRASTART mit CA Endevor SCM-RAM.	66
CARMA-VSAM-Dateien erstellen	66
CRASRV.properties anpassen	66
crastart.endevor.conf anpassen	66
Zusätzliche Anpassung des CA Endevor SCM-	
RAM (optional)	68
CRASTART mit Muster-RAM	68
CARMA-VSAM-Dateien erstellen	68
CARMA	68
Muster-RAM	68
CRASRV.properties anpassen	69
crastart.conf anpassen	69

	Zusätzliche benutzerdefinierte Anpassung des RAM (optional)	70
	Batchübergabe mit CA Endevor SCM-RAM.	70
	CARMA-VSAM-Dateien erstellen	70
	CRASRV.properties anpassen	70
	CRASUBCA anpassen	71
	Zusätzliche Anpassung des CA Endevor SCM-RAM (optional)	72
	Batchübergabe mit Muster-RAM	73
	VSAM-Dateien erstellen	73
	CARMA	73
	Muster-RAM	73
	CRASRV.properties anpassen	73
	CRASUBMT anpassen	74
	Zusätzliche benutzerdefinierte Anpassung des RAM (optional)	74
	CARMA-Konfigurationsdetails	75
	CRASRV.properties: RSE-Schnittstelle zu CARMA crastart*.conf - CRASTART-Serverstart	75
	CRASTART-Protokolldateien erfassen.	81
	CRASUB* - Serverstart mit Batchübergabe	82
	CARMA-VSAM-Dateien	83
	CRADEF - Konfigurationsdatei	84
	CRAMSG - Nachrichtendatei	84
	CRASTRS - Datei mit angepassten Zeichenfolgen	84
	Migrationshinweise für CARMA-VSAM	84
	CARMA-RAM (Repository Access Manager)	85
	CA Endevor SCM-RAM	86
	RAM für CA Endevor SCM-Pakete	86
	PDS-RAM	86
	Skeleton-RAM	87
	SCLM-RAM	87
	CRACFG, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endevor SCM-RAM	87
	CRACFG, Interaktion von CA Endevor SCM-RAM mit dem SCM	88
	"CRASHOW": CA Endevor SCM-RAM-Standardfilter	88
	CRATMAP - Zuordnungen von CA Endevor SCM-RAM-Dateierweiterungen	88
	CRANDVRA - Zuordnungs-Exec für den CA Endevor SCM-RAM	88
	Batch-Aktionen für CA Endevor SCM-RAM	89
	CRABCFG - Konfiguration der Batch-Aktion für den CA Endevor SCM-RAM	90
	CRABATCA - JCL für Batch-Aktionen des CA Endevor SCM-RAM	91
	CRABJOBC - Jobkarte für Batch-Aktionen des CA Endevor SCM-RAM	92
	CRAALLOC - Angepasste Zuordnungs-Exec für den RAM	92
	CARMA-Rückkehrcodes	93
	(Optional) Unterstützung mehrerer RAM	93
	Beispiel:	94
	Angepasste Zuordnungs-Exec (optional).	95
	CARMA-Benutzerexit (optional)	96
	IRXJCL oder CRAXJCL (optional)	97
	CRAXJCL erstellen	97

Kapitel 4. SCLM Developer Toolkit (optional)	99
Voraussetzungen und Prüfliste	99
Voraussetzungen	100
Aktualisierung von ISPF.conf für SCLMDT	100
Aktualisierung von rsed.envvars für SCLMDT	101
Umsetzung langer/kurzer Namen (optional)	102
LSTRANS.FILE - VSAM-Datei für die Umsetzung langer/kurzer Namen - erstellen	102
Aktualisierung von rsed.envvars für die Umsetzung langer/kurzer Namen	104
Ant installieren und anpassen (optional)	104
SCLM-Updates für SCLMDT	105
Alte Dateien aus WORKAREA und /tmp entfernen	106

Kapitel 5. (Optional) Application Deployment Manager (veraltet)	107
Voraussetzungen und Prüfliste	107
CRD-Repository	108
CICS-Verwaltungsdienstprogramm	109
RESTful oder Web-Service	109
CRD-Server mit der RESTful-Schnittstelle	109
Primäre CICS-Verbindungsregion.	109
Nicht primäre CICS-Verbindungsregionen	110
CRD-Servertransaktions-IDs anpassen (optional)	110
CRD-Server mit der Web-Service-Schnittstelle.	111
Pipelinenachrichtenhandler	111
Primäre CICS-Verbindungsregion.	112
Nicht primäre CICS-Verbindungsregionen	113
Manifest-Repository (optional).	113

Kapitel 6. Hostbasierte Codeanalyse (optional)	115
Voraussetzungen und Prüfliste	115
Codeüberprüfung	115
Verarbeitung der Codeüberprüfung ändern	116
Codeabdeckung	116
Ausgabe der Codeabdeckung	116

Kapitel 7. Weitere Anpassungstasks (optional)	119
pushtoclient.properties - hostbasierte Clientsteuerung (optional)	119
ssl.properties - RSE-SSL-Verschlüsselung (optional)	122
rsecomm.properties - RSE-Trace (optional).	125
"include.conf" (optional): Erzwangene Includes für C/C++-Content-Assist	127
(Optional) z/OS UNIX-Unterprojekte	128
REXEC- oder SSH-Konfiguration	129
Unterstützung für Include-Vorprozessor (optional)	129
xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)	130
Enterprise Service Tools-Unterstützung (optional)	131
Unterstützung bidirektionaler Sprachen für CICS (optional).	131
IRZ-Diagnosenachrichten für generierten Code (optional)	132
Integrated Debugger (optional)	133

Konfigurationsparameter für Integrated Debugger	135
Parmlib-Aktualisierungen für Integrated Debugger	135
TCP/IP-Aktualisierungen für Integrated Debugger	135
Sicherheitsupdates für Integrated Debugger	136
CICS-Aktualisierungen für Integrated Debugger	136
Unterstützung für Tools zur Problembestimmung (optional).	136
Unterstützung für DB2- und IMS-Debug (optional)	137
Unterstützung für File Manager (optional).	138
Bereinigung von "WORKAREA" und "/tmp" (optional)	139

Kapitel 8. Installationsprüfung 141

Gestartete Tasks prüfen	141
JMON - JES Job Monitor	141
RSED - RSE-Dämon	141
DBGMR, Debug-Manager.	145
IVP-Bedienerbefehle	146
Wiederverwendbarkeit von PassTicket	146
RSE-Dämonverbindung	146
ISPF Client Gateway	146
Services prüfen.	147
Installationsprüfprogramm initialisieren	148
Portverfügbarkeit	149
TCP/IP konfigurieren	149
RSE-Dämonverbindung	151
JES Job Monitor-Verbindung	151
Verbindung zum TSO/ISPF-Client-Gateway von ISPF	152
CARMA-Verbindung (optional)	153
SCLMDT-Verbindung (optional)	154
Verbindung zu Integrated Debugger (optional)	156

Kapitel 9. Sicherheitsdefinitionen . . . 157

Voraussetzungen und Prüfliste	157
Sicherheitseinstellungen und -klassen aktivieren	158
OMVS-Segment für Benutzer von Developer for System z definieren	159
Gestartete Tasks für Developer for System z definieren	160
RSE als sicheren z/OS UNIX-Server definieren	161
Programmgesteuerte MVS-Bibliotheken für RSE definieren	162
PassTicket-Unterstützung für RSE definieren	163
Anwendungsschutz für RSE definieren	164
JES-Befehlssicherheit definieren	164
Dateiprofile definieren	166
Programmgesteuerte z/OS UNIX-Dateien für RSE definieren	170
Sicherheitseinstellungen prüfen	171

Kapitel 10. Leitfaden für die Migration 173

Aspekte der Migration	173
Zuvor konfigurierte Dateien sichern	173
Migrationshinweise für Version 9.0	174
IBM Rational Developer for System z - FMID HHOP900	174

IBM Rational Developer for System z Host Utilities, FMID HAKG900.	175
Migration von Version 8.5 auf Version 9.0	175
IBM Rational Developer for System z - FMID HHOP900	175
Konfigurierbare Dateien	177
IBM Rational Developer for System z Host Utilities, FMID HAKG900.	183
Konfigurationsdateien	183
Migrationshinweise für Version 8.5	184
Migration von Version 8.0.1 auf Version 8.5	185
IBM Rational Developer for System z - FMID HHOP850	185
Konfigurierbare Dateien	188

Kapitel 11. Bedienerbefehle 193

Start (S)	193
Integrated Debugger	193
JES Job Monitor	194
RSE-Dämon	194
Modify (F)	195
Integrated Debugger	195
JES Job Monitor	196
RSE-Dämon	199
Stop (P)	207
Hinweise zum Lesen eines Syntaxdiagramms.	207
Symbole	208
Operanden	208
Syntaxbeispiel	208
Nicht alphanumerische Zeichen und Leerzeichen	208
Mehrere Operanden auswählen	209
Mehrere Zeilen	209
Syntaxfragmente	209

Kapitel 12. Hostkonfigurationsreferenz 211

Wissenswertes zu Developer for System z	211
Sicherheitsaspekte	211
TCP/IP-Aspekte	211
Hinweise zu WLM	211
Optimierungsaspekte	211
Leistungsaspekte	212
Hinweise zu Push-to-Client.	212
CICSTS-Aspekte	212
Hinweise zum Benutzerexit	212
Anpassung der TSO-Umgebung	212
Ausführung mehrerer Instanzen	212
Konfigurationsprobleme lösen	212
SSL- und X.509-Authentifizierung konfigurieren	213
TCP/IP konfigurieren	213

Literaturübersicht 215

Referenzierte Veröffentlichungen	215
Veröffentlichungen mit weiteren Informationen	218

Bemerkungen zur Dokumentation für IBM Rational Developer for System z . 219

Abbildungen

1.	JMON: Gestartete Task von JES Job Monitor	27	22.	crastart*.conf: CARMA-Serverstart mit CRAFT	80
2.	DBGMGR: Gestartete Task des Debug-Managers	28	23.	CRASUB*: CARMA-Start mit Batchübergabe	83
3.	RSED: Gestartete Task für den RSE-Dämon	28	24.	CRACFG - Interaktion von CA Endeavor SCM-RAM mit dem SCM.	88
4.	RSED: Alternativer Start des RSE-Dämons	29	25.	CRASHOW - CA Endeavor SCM-RAM-Standardfilter	88
5.	rsed.stdin.sh: Alternativer Start des RSE-Dämons.	30	26.	CRATMAP: CA Endeavor SCM-RAM-Standardfilter	88
6.	RSED: Alternative TMPDIR-Verarbeitung	30	27.	CRABCFG: Konfiguration der Batch-Aktion für den CA Endeavor SCM-RAM	91
7.	rsed.stdev: Alternative TMPDIR-Verarbeitung	31	28.	CRABATCA: JCL für die Batch-Aktion des CA Endeavor SCM-RAM.	92
8.	FEJJCENFG - Konfigurationsdatei für JES Job Monitor	34	29.	CRABJOB: Jobkarte für Batch-Aktionen des CA Endeavor SCM-RAM	92
9.	rsed.envvars: RSE-Konfigurationsdatei	40	30.	Aktualisierung von ISPF.conf für SCLMDT	101
10.	rsed.envvars: RSE-Konfigurationsdatei (Fortsetzung)	41	31.	Aktualisierung von rsed.envvars für SCLMDT	102
11.	rsed.envvars: RSE-Konfigurationsdatei (Fortsetzung)	42	32.	FLM02LST: Konfigurations-JCL für Umsetzung langer/kurzer Namen.	103
12.	ISPF.conf: ISPF-Konfigurationsdatei	60	33.	pushtoclient.properties: Konfigurationsdatei für hostbasierte Clientsteuerung	120
13.	CRASRV.properties: CRAFT mit CA Endeavor SCM-RAM	66	34.	SSL-Konfigurationsdatei ssl.properties	124
14.	crastart.endevor.conf: CRAFT mit CA Endeavor SCM-RAM	67	35.	rsecomm.properties - Konfigurationsdatei für Protokollierung	126
15.	CRASRV.properties: CRAFT mit Muster-RAM.	69	36.	"include.conf": Erzwungene Includes für C/C++-Content-Assist	128
16.	crastart.conf: CRAFT mit Muster-RAM	69	37.	Bedienerbefehl 'START DBGMR'	193
17.	CRASRV.properties: Batchübergabe mit CA Endeavor SCM-RAM.	70	38.	Bedienerbefehl 'START JMON'	194
18.	CRASUBCA: Batchübergabe mit CA Endeavor SCM-RAM	72	39.	Bedienerbefehl 'START RSED'	194
19.	CRASRV.properties: Batchübergabe mit Muster-RAM	74	40.	Bedienerbefehl 'MODIFY DBGMR'	195
20.	CRASUBMT: Batchübergabe mit Muster-RAM	74	41.	Bedienerbefehl 'MODIFY JMON'	196
21.	CARMA-Konfigurationsdatei CRASRV.properties	75	42.	Bedienerbefehl 'MODIFY RSED'	199
			43.	Bedienerbefehl 'STOP'	207

Tabellen

1. Erforderliche Ressourcen	6	15. Standard-Transaktions-IDs des CRD-Servers	111
2. Optionale Ressourcen	6	16. Unterstützung der Push-to-Client-Gruppe	122
3. Administratoren für erforderliche Tasks	7	17. Mechanismen für den SSL-Zertifikatsspeicher	123
4. Administratoren für optionale Tasks.	8	18. Gültige Keystoretypen	125
5. Clientprüfliste: obligatorischer Teil	14	19. Installationsprüfprogramme für Services	148
6. Clientprüfliste: optionaler Teil	14	20. Variablen der Sicherheitskonfiguration	157
7. Zuordnung von Lademodulen zu Funktionen	23	21. Bedienerbefehle von JES2 Job Monitor	165
8. ELAXF*-Beispielprozeduren	31	22. Bedienerbefehle von JES3 Job Monitor	165
9. Prüfliste der übergeordneten Qualifikationsmerkmale in ELAXF*	32	23. Anpassungen in Version 9.0.	178
10. ELAXF*	32	24. Anpassungen in Host Utilities Version 9.0	183
11. Matrix der Befehlsberechtigungen für LIMIT- _COMMANDS	36	25. Anpassungen in Version 8.5.	188
12. CARMA-Rückkehrcodes	93	26. Fehlerstatus des Thread-Pools	201
13. Prüfliste für den SCLM-Administrator	105	27. Referenzierte Veröffentlichungen	215
14. Standard-Transaktions-IDs des CRD-Servers	110	28. Referenzierte Websites	217
		29. Veröffentlichungen mit weiteren Informationen	218

Zu diesem Handbuch

Dieses Handbuch beschäftigt sich mit der Konfiguration der Funktionen von IBM® Rational Developer for System z. Es enthält Konfigurationsanweisungen für IBM Rational Developer for System z Version 9.0.1 auf Ihrem z/OS-Hostsystem.

Im weiteren Verlauf dieses Handbuchs werden die folgenden Namen verwendet:

- *IBM Rational Developer for System z* wird als *Developer for System z* bezeichnet.
- *IBM Rational Developer for System z Integrated Debugger* wird als 'Integrated Debugger' bezeichnet.
- *Common Access Repository Manager* wird mit CARMA abgekürzt.
- Das *Software Configuration and Library Manager Developer Toolkit* wird als *SCLM Developer Toolkit* bezeichnet und mit SCLMDT abgekürzt.
- *IBM z/OS Automated Unit Testing Framework* wird als *zUnit* bezeichnet.
- *z/OS UNIX System Services* wird als *z/OS UNIX* bezeichnet.
- *Customer Information Control System Transaction Server* wird als *CICSTS* bezeichnet und mit *CICS* abgekürzt.

Dieses Dokument ist Teil einer Reihe von Dokumenten, in denen die Hostsystemkonfiguration für Developer for System z beschrieben wird. Jedes dieser Dokumente hat eine spezielle Zielgruppe. Zur Durchführung der Developer for System z-Konfiguration müssen Sie nicht alle Dokumente lesen.

- In *Rational Developer for System z Hostkonfiguration* (IBM Form SC12-4062) werden alle Planungstasks, Konfigurationstasks und Optionen (einschließlich der optionalen) ausführlich beschrieben und alternative Szenarios bereitgestellt.
- In *Rational Developer for System z Hostkonfigurationsreferenz* (IBM Form SC12-4489) wird das Design von Developer for System z beschrieben. Das Handbuch enthält außerdem Hintergrundinformationen für verschiedene Konfigurationstasks von Developer for System z, z/OS-Komponenten und weiteren Produkten (wie WLM und CICS) in Verbindung mit Developer for System z.
- Im *Rational Developer for System z Leitfaden für den Schnelleinstieg in die Hostkonfiguration* (IBM Form GI11-3191) wird eine Minimalkonfiguration von Developer for System z beschrieben.
- In *Rational Developer for System z Host Configuration Utility* (IBM Form SC12-4472) wird das Host Configuration Utility beschrieben, eine ISPF-Anzeigeapplikation, die Sie durch grundlegende und allgemeine optionale Anpassungsschritte für Developer for System z führt.

Die Informationen in diesem Dokument gelten für alle Pakete von IBM Rational Developer for System z Version 9.0.

Zielgruppe

Dieses Dokument ist für Systemprogrammierer gedacht, die IBM Rational Developer for System z Version 9.0.1 installieren und konfigurieren.

In diesem Dokument werden die Schritte, die für eine vollständige Einrichtung des Produkts erforderlich sind, sowie einige vom Standard abweichende Szenarios beschrieben. Hintergrundinformationen, die Sie bei der Planung und Ausführung der Konfiguration unterstützen, finden Sie in *IBM Rational Developer for System z Hostkonfigurationsreferenz* (IBM Form SC12-4489). Voraussetzung für die Verwendung dieses Handbuchs ist, dass Sie mit z/OS UNIX System Services und den MVS-Hostsystemen vertraut sind.

Zusammenfassung der Änderungen

In diesem Abschnitt werden die Änderungen für *IBM Rational Developer for System z Version 9.0.1 Hostkonfiguration* (IBM Formnummer SC12-4062-10) zusammengefasst (aktualisiert im Dezember 2013).

Technische Änderungen oder Zusätze zum Text und den Abbildungen sind durch eine vertikale Linie auf der linken Seite der Änderung angegeben.

Neue Informationen:

- Migrationsinformationen für Version 9.0.1. Lesen Sie hierzu „Migrationshinweise für Version 9.0“ auf Seite 174.
- Neue optionale Aktualisierungen von PARMLIB. Lesen Sie hierzu „PARMLIB-Änderungen“ auf Seite 19.
- Neue optionale gestartete Tasks. Siehe „PROCLIB-Änderungen“ auf Seite 26
- Neue Bedienerbefehle. Siehe Kapitel 11, „Bedienerbefehle“, auf Seite 193
- Neue und geänderte Anweisungen in `rsed.envvars`. Lesen Sie hierzu „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.
- Neue Anweisungen in `"CRASRV.properties"`. Lesen Sie hierzu „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 75.
- Neue Konfigurationsdatei für CARMA CA Endevor® SCM-RAM. Lesen Sie hierzu „CRACFG, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endevor® SCM-RAM“ auf Seite 87.
- Unterstützung eines beim CARMA-Start aufgerufenen Benutzerexits. Lesen Sie hierzu „CARMA-Benutzerexit (optional)“ auf Seite 96.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for System z Version 9.0 Hostkonfiguration* (IBM Formnummer SC12-4062-09) enthalten waren.

Neue Informationen:

- Migrationsinformationen für Version 9.0. Lesen Sie hierzu „Migration von Version 8.5 auf Version 9.0“ auf Seite 175.
- Neue und geänderte Anweisungen in `rsed.envvars`. Lesen Sie hierzu „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.
- Neue und entfernte Anweisungen in `FEJJCNFG`. Lesen Sie hierzu „FEJJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34.
- Neue ELAXF*-PROCLIB-Member. Lesen Sie hierzu „ELAXF*-Prozeduren für ferne Build-Erstellung“ auf Seite 31.
- Neue JMON- und RSED-Bedienerbefehle. Lesen Sie hierzu „Modify (F)“ auf Seite 195.
- Hinzugefügte Informationen zur hostbasierten Codeanalyse. Lesen Sie hierzu Kapitel 6, „Hostbasierte Codeanalyse (optional)“, auf Seite 115.

Entfernte Informationen:

- Die gestartete LOCKD-Task wird nicht mehr verwendet, sodass alle Informationen zum Sperrdämon entfernt wurden.
- Das Beispiel für die gespeicherte DB2-Prozedur wurde durch neue ELAXF*-Prozeduren für die Build-Erstellung ersetzt, sodass alle Informationen über die gespeicherte DB2-Prozedur entfernt wurden.
- Migrationsinformationen für nicht unterstützte Releases wurden entfernt.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for System z Version 8.5.1 Hostkonfiguration* (IBM Form SC12-4062-08) enthalten waren.

Neue Informationen:

- Migrationsinformationen für Version 8.5.1. Lesen Sie hierzu „Migrationshinweise für Version 8.5“ auf Seite 184.
- Neue und geänderte Anweisungen in `rsed.envvars`. Lesen Sie hierzu „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.
- Erweiterte Unterstützung für Batch-Aktionen für CA Endevor® SCM-RAM . Lesen Sie hierzu „Batch-Aktionen für CA Endevor® SCM-RAM“ auf Seite 89.
- Anforderungen für die ID des Installationsbenutzers. Lesen Sie hierzu „ID des Installationsbenutzers“ auf Seite 5.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for System z Version 8.5 Hostkonfiguration* (IBM Formnummer SC12-4062-07) enthalten waren.

Neue Informationen:

- Neue optionale Anweisungen in `FEJJCNFG`. Lesen Sie hierzu „FEJJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34.
- Neue optionale Anweisungen in `rsed.envvars`. Lesen Sie hierzu „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.
- Neue optionale Konfigurationsdatei. Lesen Sie hierzu „`include.conf`“ (optional): Erzwingene Includes für C/C++-Content-Assist“ auf Seite 127.
- Neue optionale Komponente. Lesen Sie hierzu „Unterstützung für Include-Vorprozessor (optional)“ auf Seite 129.
- Neue optionale Komponente. Lesen Sie hierzu „xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)“ auf Seite 130.

- Neue optionale Komponente. Lesen Sie hierzu „Unterstützung für DB2- und IMS-Debug (optional)“ auf Seite 137.
- Neue und erweiterte Bedienerbefehle. Lesen Sie hierzu Kapitel 11, „Bedienerbefehle“, auf Seite 193.
- Migrationsinformationen für Version 8.5. Lesen Sie hierzu „Migration von Version 8.0.1 auf Version 8.5“ auf Seite 185.

Entfernte Informationen:

- Die Unterstützung für die File Manager-Integration wurde geändert. Daher wurden die meisten Informationen im Abschnitt „Unterstützung für File Manager (optional)“ auf Seite 138 entfernt.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *Rational Developer for System z Version 8.0.3 Hostkonfiguration* (IBM Form SC12-4062-06) enthalten waren.

Neue Informationen:

- Neue Anweisungen in `rsed.envvars`. Lesen Sie hierzu „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.
- CA Endeavor® SCM-Hintergrundaktionen werden jetzt unterstützt. Lesen Sie hierzu „Batch-Aktionen für CA Endeavor® SCM-RAM“ auf Seite 89.
- CA Endeavor® SCM-Pakete werden jetzt unterstützt. Lesen Sie hierzu „CARMA-RAM (Repository Access Manager)“ auf Seite 85.
- Neue Anweisungen in `pushtoclient.properties`. Lesen Sie hierzu „pushtoclient.properties - hostbasierte Clientsteuerung (optional)“ auf Seite 119.
- Die File Manager-Integration ist veraltet. Lesen Sie hierzu „Unterstützung für File Manager (optional)“ auf Seite 138.
- Neue und erweiterte Bedienerbefehle. Lesen Sie hierzu Kapitel 11, „Bedienerbefehle“, auf Seite 193.
- Neue Veröffentlichung *IBM Rational Developer for System z Messages and Codes* (IBM Form SC14-7497).

Dieses Dokument enthält Informationen, die zuvor im Handbuch *Rational Developer for System z Version 8.0.1 Hostkonfiguration* (IBM Form SC12-4062-05) enthalten waren.

Neue Informationen:

- Neue Anweisungen in `FEJJCENFG`. Lesen Sie hierzu „FEJJCENFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34.
- Neue Anweisungen in `rsed.envvars`. Lesen Sie hierzu „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.
- Das Kapitel zu CARMA wurde reorganisiert und einige neue Informationen wurden hinzugefügt. Lesen Sie hierzu Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 63.
- Neue Konfigurationsdatei `pushtoclient.properties`. Lesen Sie hierzu „pushtoclient.properties - hostbasierte Clientsteuerung (optional)“ auf Seite 119.

Entfernte Informationen:

- Die Informationen, die zuvor im Handbuch *Rational Developer for System z Version 7.6.1 Hostkonfiguration* (IBM Form SC12-4062-04) enthalten waren, wurden auf zwei Dokumente aufgeteilt: *Rational Developer for System z Hostkonfiguration* (IBM Form SC12-4062) und *Rational Developer for System z Hostkonfigurationsreferenz* (IBM Form SC12-4489).
- Informationen zur APPC-Konfiguration wurden in das White Paper mit dem Titel *Using APPC to provide TSO command services* (IBM Form SC14-7291) verschoben.
- Informationen zu CARMA mit dem ISPF-Client-Gateway wurden in das White Paper mit dem Titel *Using ISPF Client Gateway to provide CARMA services* (IBM Form SC14-7292) verschoben.
- Abschnitt "Hostbasierte Eigenschaftsgruppen (optional)" in "Weitere Anpassungstasks (optional)" (Beschreibung von `propertiescfg.properties`).
- Abschnitt "Hostbasierte Projekte (optional)" in "Weitere Anpassungstasks (optional)" (Beschreibung von `projectcfg.properties`).
- Abschnitt "Nicht editierbare Zeichen (optional)" unter "Weitere Anpassungstasks (optional)" (Beschreibung von `"uchars.settings"`).
- Abschnitt "Version 7.6.1 Migrationshinweise" im "Migrationshandbuch".

Beschreibung des Dokumentinhalts

In diesem Abschnitt werden die in diesem Dokument enthaltenen Informationen zusammengefasst.

Planung

Nutzen Sie für die Planung der Installation und des Deployments von Developer for System z die Informationen in diesem Kapitel.

Basisanpassung

Die folgenden Anpassungsschritte beziehen sich auf eine Basiskonfiguration von Developer for System z:

- „Anpassungskonfiguration“ auf Seite 17
- „PARMLIB-Änderungen“ auf Seite 19
- „PROCLIB-Änderungen“ auf Seite 26
- „Sicherheitsdefinitionen“ auf Seite 33
- „FEJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34
- „RSE-Konfigurationsdatei rsed.envvars“ auf Seite 39
- „ISPF.conf - Konfigurationsdatei für TSO/ISPF Client Gateway“ auf Seite 59

Common Access Repository Manager (optional)

Common Access Repository Manager (CARMA) ist eine Serverplattform für Repository Access Manager (RAM). Ein RAM ist eine Anwendungsprogrammierschnittstelle (API) für einen Software Configuration Manager (SCM), der auf einem z/OS-System basiert. Indem die SCM-Funktionalität in einen RAM eingeschlossen wird, ist eine einzige API verfügbar, damit ein Client auf alle unterstützten SCM zugreifen kann.

Developer for System z stellt mehrere vordefinierte RAM und Quellcodemuster bereit, um einen eigenen RAM zu erstellen.

Die Schnittstelle für CA Endevor® Software Configuration Manager in IBM Rational Developer for System z gibt Developer for System z-Clients direkten Zugriff auf CA Endevor® SCM.

SCLM Developer Toolkit (optional)

Das SCLM Developer Toolkit stellt die Tools bereit, mit denen die Funktionalität von SCLM auch auf dem Client verfügbar gemacht werden kann. SCLM selbst ist ein hostbasierter Quellcodemanager, der in ISPF enthalten ist.

Das SCLM Developer Toolkit enthält ein Eclipse-basiertes Plug-in, das die Schnittstelle zu SCLM bildet. Es ermöglicht den Zugriff auf alle SCLM-Prozesse für die bisherige Codeentwicklung und stellt Unterstützung für die vollständige Java™- und J2EE-Entwicklung auf der Workstation mit Synchronisation mit SCLM auf dem Großrechner (Mainframe-Computer) bereit. Die Synchronisationsaktivitäten umfassen die Builderstellung, Assemblierung und Implementierung von J2EE-Code über den Großrechner.

Application Deployment Manager (optional, veraltet)

Developer for System z verwendet bestimmte Funktionen des Application Deployment Manager als allgemeine Deployment-Methode für verschiedene Komponenten. Durch eine optionale Anpassung können weitere Features von Application Deployment Manager aktiviert und die folgenden Services zu Developer for System z hinzugefügt werden:

- IBM CICS Explorer stellt eine Eclipse-basierte Infrastruktur für die Anzeige und Verwaltung von CICS-Ressourcen bereit und verbessert die Integration der verschiedenen CICS-Tools.
- Der CICS Resource Definition-Client (CRD-Client) und der CRD-Server stellen folgende Funktionen bereit:
 - Editor für CICS-Ressourcendefinitionen
 - CICS-Ressourcen, die von Anwendungsentwicklern in begrenzter, kontrollierter und sicherer Weise definiert werden
 - CICS-Entwicklungszugriff, der auf nicht autorisierte oder falsche VSAM-Dateien nicht verfügbar ist, durch Bereitstellen der CICS-Administratorsteuerung über das Namensattribut für physische Dateien in Dateidefinitionen
 - Sonstige Unterstützung für die CICS-Entwicklung
 - Sonstige Unterstützung für die Entwicklung von CICS-Web-Services

Hostbasierte Codeanalyse (optional)

Ähnlich wie der Developer for System z-Client unterstützt der Developer for System z-Host die Ausführung von Codeanalysetools, die als separates Produkt unter der Bezeichnung Rational Developer for System z Host Utilities bereitgestellt werden. Ein Vorteil der Ausführung der Codeanalyse auf dem Host besteht darin, dass sie in Ihre tägliche Stapelverarbeitung integriert werden kann.

Die folgenden Codeanalysetools sind auf dem Host verfügbar:

- Codeüberprüfung: Mithilfe von Regeln unterschiedlicher Fehlerkategorien prüft die Codeüberprüfung Quellcode und meldet Regelverstöße.
- Codeabdeckung: Analysiert ein aktives Programm und generiert einen Bericht zu den Zeilen, die ausgeführt werden, im Vergleich zur Gesamtzahl der ausführbaren Zeilen.

Weitere Anpassungstasks (optional)

In den folgenden Abschnitten ist eine Kombination optionaler Anpassungstasks beschrieben. Zur Konfiguration des erforderlichen Service führen Sie die Anweisungen im jeweiligen Abschnitt aus.

Anpassungen in Konfigurationsdateien von Developer for System z:

- `pushtoclient.properties`: Hostbasierte Clientsteuerung
- `ssl.properties`: RSE-SSL-Verschlüsselung
- `rsecomm.properties`: RSE-Trace
- `include.conf`: Erzwungene Includes für C/C++-Content-Assist

Anpassungen an oder für andere Produkte in Bezug auf Developer for System z:

- Gespeicherte DB2-Prozedur
- z/OS UNIX-Unterprojekte
- Unterstützung für Include-Vorprozessor
- xUnit-Unterstützung für Enterprise COBOL und PL/I
- Enterprise Service Tools-Unterstützung
- Unterstützung bidirektionaler Sprachen für CICS
- IRZ-Diagnosenachrichten für generierten Code
- Integrated Debugger
- Unterstützung für Tools zur Problembestimmung
- Unterstützung für DB2- und IMS-Debug
- File Manager-Unterstützung
- Bereinigung von "WORKAREA" und "/tmp"

Installationsprüfung

Nach Abschluss der Produkthanpassung können Sie die erfolgreiche Konfiguration der zentralen Produktkomponenten mit den in diesem Kapitel beschriebenen IVPs (Installation Verification Programs) überprüfen.

Sicherheitsdefinitionen

In diesem Abschnitt werden die erforderlichen und optionalen Sicherheitsdefinitionen mit RACF-Beispielbefehlen beschrieben.

Leitfaden für die Migration

In diesem Abschnitt werden die Installations- und Konfigurationsänderungen im Vergleich zu den früheren Produktreleases erläutert. Darüber hinaus finden Sie hier allgemeine Richtlinien für die Migration auf dieses Release.

Bedienerbefehle

In diesem Abschnitt erhalten Sie einen Überblick über die für Developer for System z verfügbaren Bedienerbefehle (oder Konsolenbefehle).

Hostkonfigurationsreferenz

In diesem Abschnitt werden die Informationen im Handbuch *IBM Rational Developer for System z Hostkonfigurationsreferenz* (IBM Form SC12-4489) zusammengefasst.

IBM Rational Developer for System z Hostkonfiguration

Kapitel 1. Planung

Nutzen Sie für die Planung der Installation und der Implementierung von Developer for System z die Informationen in diesem Kapitel und in *IBM Rational Developer for System z Prerequisites* (IBM Form SC23-7659). Die folgenden Themen werden behandelt:

- „Aspekte der Migration“
- „Hinweise zur Planung“
- „Hinweise zur Installationsvorbereitung“ auf Seite 4
- „Hinweise zur Konfigurationsvorbereitung“ auf Seite 10
- „Hinweise zur Implementierungsvorbereitung“ auf Seite 12
- „Clientprüfliste“ auf Seite 14

Aspekte der Migration

Kapitel 10, „Leitfaden für die Migration“, auf Seite 173 beschreibt die Installations- und Konfigurationsänderungen im Vergleich zu früheren Produktreleases. Nutzen Sie diese Informationen für die Planung Ihrer Migration auf das aktuelle Release von Developer for System z.

Anmerkung:

- Wenn Sie mit einer früheren Version von IBM Rational Developer for System z, IBM WebSphere Developer for System z, IBM WebSphere Developer for zSeries oder IBM WebSphere Studio Enterprise Developer arbeiten, speichern Sie die zugehörigen angepassten Dateien, bevor Sie IBM Rational Developer for System z Version 9.0 installieren. Eine Übersicht über die Dateien, die eine Anpassung erforderten, finden Sie im Kapitel 10, „Leitfaden für die Migration“, auf Seite 173.
- Wenn Sie planen, mehrere Instanzen von Developer for System z auszuführen, lesen Sie die Informationen zur "Ausführung mehrerer Instanzen" in der *Host-konfigurationsreferenz* (IBM Form SC12-4489-02).

Hinweise zur Planung

Produktübersicht

Developer for System z besteht aus einem Client, der auf dem Personal Computer des Benutzers installiert ist, und einem Server, der auf mindestens einem Hostsystem installiert ist. Diese Dokumentation enthält Informationen für ein z/OS-Hostsystem. Andere Betriebssysteme, wie AIX und Linux auf System z werden jedoch auch unterstützt.

Der Client stellt Entwicklern eine Eclipse-basierte Entwicklungsumgebung zur Verfügung, die eine einheitliche grafische Oberfläche für den Host ermöglicht. Unter anderem kann Arbeit vom Host auf den Client ausgelagert werden, wodurch Ressourcen auf dem Host gespart werden.

Die Hostkomponente besteht aus einigen ständig aktiven Tasks sowie Tasks, die ad hoc gestartet werden. Diese Tasks ermöglichen es dem Client, mit den verschiedenen Komponenten Ihres z/OS-Hostsystems zu arbeiten, wie zum Beispiel mit MVS-Dateien, TSO-Befehlen, z/OS UNIX-Dateien und -Befehlen sowie mit Jobübergabe und Jobausgabe.

Developer for System z kann auch mit Subsystemen und anderer Anwendungssoftware auf dem Hostsystem interagieren, wie beispielsweise CICS, IBM File Manager und Software Configuration Manager (SCM), wenn Developer for System z hierfür konfiguriert ist und wenn diese zusätzlich erforderlichen Produkte verfügbar sind.

Eine grundlegende Beschreibung des Designs von Developer for System z finden Sie in "Wissenswertes zu Developer for System z" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

Weitere Informationen zu den von Developer for System z bereitgestellten Funktionen finden Sie auf der Webseite für Developer for System z unter der Adresse <http://www-03.ibm.com/software/products/us/en/developerforsystemz/>. Alternativ können Sie sich auch an Ihren IBM Ansprechpartner wenden.

Erforderliche Qualifikationen

Für eine Hostinstallation von Developer for System z sind SMP/E-Kenntnisse erforderlich.

Die Konfiguration von Developer for System z erfordert mehr als die typischen Berechtigungen und Kenntnisse in Bezug auf die Systemprogrammierung. Daher wird wahrscheinlich weitere Unterstützung benötigt. In Tabelle 3 auf Seite 7 und Tabelle 4 auf Seite 8 sind die Administratoren aufgeführt, die für die erforderlichen und optionalen Anpassungstasks benötigt werden.

Zeitbedarf

Die für die Installation und Konfiguration der Hostsystemkomponenten von Developer for System z benötigte Zeit hängt von verschiedenen Faktoren ab. Dazu gehören:

- Die aktuelle z/OS UNIX- und TCP/IP-Konfiguration
- Die Verfügbarkeit von Softwarevoraussetzungen und Wartung
- Ob OMVS-Segmente für Developer for System z-Benutzer definiert sind
- Die Verfügbarkeit eines Benutzers, der den Client erfolgreich installiert hat, um die Installation zu testen und ggf. Probleme zu melden

Die Erfahrung hat gezeigt, dass der Installations- und Konfigurationsprozess für das Developer for System z-Hostsystem einen Zeitaufwand von einem bis zu vier Tagen erfordert. Diese Zeitangabe gilt für eine reibungslose Installation, die durch einen erfahrenen Systemprogrammierer durchgeführt wird. Falls Probleme auftreten oder das erforderliche Know-how fehlt, dauert die Installation länger.

Hinweise zur Installationsvorbereitung

Ausführliche Anweisungen zur SMP/E-Installation des Produkts finden Sie in der Veröffentlichung *Program Directory for IBM Rational Developer for System z* (GI11-8298).

Informationen zur Ausführung mehrerer Instanzen von Developer for System z finden Sie in "Ausführung mehrerer Instanzen" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

Das Dateisystem (HFS oder zFS), in dem Developer for System z installiert ist, muss mit gesetztem Berechtigungsbit SETUID angehängt werden. (Dies ist der Systemstandardwert.) Wenn Sie das Dateisystem mit dem Parameter NOSETUID anhängen, kann Developer for System z keine Sicherheitsumgebung für den Benutzer

erstellen, sodass die Verbindungsanforderungen des Clients zurückgewiesen werden. Dasselbe gilt für die Dateisysteme, die das Hosting für Java- und z/OS UNIX-Binärprogramme übernehmen.

ID des Installationsbenutzers

Die Benutzer-ID, die zur Installation von Developer for System z oder zur Installation einer Wartung verwendet wird, muss mindestens die folgenden Attribute haben:

- TSO-Zugriff (mit normaler Regionsgröße)
- Ein für das Sicherheitssystem (z. B. RACF) definiertes OMVS-Segment für die Benutzer-ID und für die zugehörige Standardgruppe
 - Das Feld HOME muss auf ein dem Benutzer zugeordnetes Ausgangsverzeichnis mit den Zugriffsrechten READ, WRITE und EXECUTE verweisen.
 - Das Feld PROGRAM im OMVS-Segment sollte auf /bin/sh oder eine andere gültige z/OS UNIX-Shell, z. B. /bin/tcsh, gesetzt sein.
 - Für die Standardgruppe, zu der die Benutzer-ID gehört, ist eine Gruppen-ID (GID) erforderlich.
- UID=0 oder READ-Berechtigung für das Profil BPX.SUPERUSER in der Klasse FACILITY.
- Wenn das Profil BPX.FILEATTR.APF oder das Profil BPX.FILEATTR.PROGCTL in der Klasse FACILITY definiert sind, READ-Zugriff auf diese Profile.
- Zugriffsrechte READ, WRITE und EXECUTE für das Verzeichnis /tmp (oder ein Verzeichnis, auf das in der Umgebungsvariablen TMPDIR verwiesen wird).

Vorausgesetzte Produkte

IBM Rational Developer for System z Prerequisites (IBM Form SC23-7659) enthält eine Liste der vorausgesetzten Software, die installiert und betriebsbereit sein muss, damit Developer for System z ordnungsgemäß funktionieren kann. Außerdem gibt es eine Liste zusätzlich erforderlicher Software zur Unterstützung bestimmter Features von Developer for System z. Zur Laufzeit muss diese zusätzlich erforderliche Software installiert und betriebsbereit sein, damit das entsprechende Feature ordnungsgemäß funktionieren kann.

Planen Sie vorausschauend, damit die vorausgesetzten Produkte rechtzeitig verfügbar sind. Dies kann je nach Geschäftspolitik an Ihrem Standort einige Zeit in Anspruch nehmen. Nachfolgend sind die wichtigsten Voraussetzungen für eine Basis-konfiguration aufgeführt:

- z/OS ab Version 1.8
- ISPF APAR OA43014 (TSO/ISPF-Client-Gateway)
- Java ab Version 6.0 (31- oder 64-Bit)

Erforderliche Ressourcen

Developer for System z erfordert die Reservierung der in Tabelle 1 auf Seite 6 aufgelisteten Systemressourcen. Die in Tabelle 2 auf Seite 6 aufgelisteten Ressourcen sind für Zusatzservices erforderlich. Planen Sie, diese Ressourcen bereitzustellen, weil die Beschaffung der Software je nach den Richtlinien Ihres Standorts einige Zeit in Anspruch nehmen kann.

Tabelle 1. Erforderliche Ressourcen

Ressource	Standardwert	Informationen
Datei mit APF-Berechtigung	FEK.SFEKAUTH	„APF-Berechtigungen in PROGxx“ auf Seite 22
Gestartete Task	JMON und RSED	„PROCLIB-Änderungen“ auf Seite 26
Port für die hostinterne Verwendung (JMON)	6715	„FEJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34
Port für Client-Host-Kommunikation (RSED)	4035	„RSE-Konfigurationsdatei rsed.envvars“ auf Seite 39
Portbereich für Client-Host-Kommunikation (RSED)	Jeder verfügbare Port kann verwendet werden.	„Verfügbaren Portbereich für den RSE-Server definieren“ auf Seite 49
Sicherheitsdefinition für den z/OS UNIX-Server	Berechtigung "UPDATE" für BPX.SERVER für die gestartete RSED-Task	„RSE als sicheren z/OS UNIX-Server definieren“ auf Seite 161
PassTicket-Sicherheitsdefinitionen	Kein Standard	„PassTicket-Unterstützung für RSE definieren“ auf Seite 163
MVS-Buildprozeduren	ELAXF*	„PROCLIB-Änderungen“ auf Seite 26

Tabelle 2. Optionale Ressourcen

Ressource	Standardwert	Informationen
Einleitendes Programmladen mit CLPA	Nicht anwendbar	„Integrated Debugger (optional)“ auf Seite 133
Gestartete Task	DBGMGR	„Integrated Debugger (optional)“ auf Seite 133
LINKLIST-Dateigruppe	FEK.SFEKAUTH und FEK.SFEKLOAD	<ul style="list-style-type: none"> • Kapitel 4, „SCLM Developer Toolkit (optional)“, auf Seite 99 • „Integrated Debugger (optional)“ auf Seite 133
LPA-Datei	FEK.SFEKLPA	Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 63
Portbereich für die hostinterne Verwendung	Jeder verfügbare Port kann verwendet werden.	<ul style="list-style-type: none"> • Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 63 • Kapitel 6, „Hostbasierte Codeanalyse (optional)“, auf Seite 115
Portbereich für die hostinterne Verwendung	5336	„Integrated Debugger (optional)“ auf Seite 133

Tabelle 2. Optionale Ressourcen (Forts.)

Ressource	Standardwert	Informationen
Port für die Kommunikation zwischen Client und Host	<ul style="list-style-type: none"> • 5129 für Web-Service oder 5130 für RESTful-Services • 5335 für Integrated Debugger 	<ul style="list-style-type: none"> • Kapitel 5, „(Optional) Application Deployment Manager (veraltet)“, auf Seite 107 • „Integrated Debugger (optional)“ auf Seite 133
Update für die CICS-Systemdefinition	Mehrere Werte	<ul style="list-style-type: none"> • Kapitel 5, „(Optional) Application Deployment Manager (veraltet)“, auf Seite 107 • „Integrated Debugger (optional)“ auf Seite 133
Update für CICS-JCL	<ul style="list-style-type: none"> • FEK.SFEKLOAD • FEK.SFEKAUTH 	<ul style="list-style-type: none"> • Kapitel 5, „(Optional) Application Deployment Manager (veraltet)“, auf Seite 107 • „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 131 • „Integrated Debugger (optional)“ auf Seite 133

Die Konfiguration von Developer for System z erfordert mehr als die typischen Berechtigungen und Kenntnisse in Bezug auf die Systemprogrammierung. Daher kann weitere Unterstützung erforderlich sein. In Tabelle 3 und Tabelle 4 auf Seite 8 sind die Administratoren aufgeführt, die für die erforderlichen und optionalen Anpassungstasks benötigt werden.

Tabelle 3. Administratoren für erforderliche Tasks

Administrator	Task	Informationen
Systemadministrator	Für alle Anpassungstasks sind typische Systemprogrammiereraktionen erforderlich.	Nicht zutreffend

Tabelle 3. Administratoren für erforderliche Tasks (Forts.)

Administrator	Task	Informationen
Sicherheitsadministrator	<ul style="list-style-type: none"> • OMVS-Segment für Benutzer von Developer for System z definieren • Dateiprofile definieren • Gestartete Tasks definieren • Befehlssicherheit für Bediener definieren • z/OS UNIX-Serverprofile definieren • Anwendungssicherheit definieren • PassTicket-Unterstützung definieren • Programmgesteuerte Dateien definieren • Programmgesteuerte z/OS UNIX-Dateien definieren 	"Sicherheitsaspekte" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02)
TCP/IP-Administrator	Neue TCP/IP-Ports definieren	"TCP/IP-Aspekte" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02)
WLM-Administrator	Ziele der gestarteten Tasks den Servern und deren untergeordneten Prozessen zuordnen	"Hinweise zu WLM" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02).

Tabelle 4. Administratoren für optionale Tasks

Administrator	Task	Informationen
Systemadministrator	Für alle Anpassungstasks sind typische Systemprogrammiereraktionen erforderlich.	Nicht zutreffend

Tabelle 4. Administratoren für optionale Tasks (Forts.)

Administrator	Task	Informationen
Sicherheitsadministrator	<ul style="list-style-type: none"> • Dateiprofile definieren • Programmgesteuerte Dateien definieren • Berechtigung für die Übergabe von xxx*-Jobs definieren • CICS-Transaktionssicherheit definieren • Zertifikat für SSL hinzufügen • Unterstützung des X.509-Clientzertifikats definieren • Gruppen und Profile für Push-to-Client definieren • Profile zum Ändern von Clientfunktionen definieren • Gestartete Tasks definieren • z/OS UNIX-Serverprofile definieren • Profile für autorisiertes Debugging definieren 	<ul style="list-style-type: none"> • "Sicherheitsaspekte" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02) • "CICSTS-Aspekte" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02) • "SSL- und X.509-Authentifizierung konfigurieren" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02)
TCP/IP-Administrator	Neue TCP/IP-Ports definieren	"TCP/IP-Ports" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02)
SCLM	<ul style="list-style-type: none"> • SCLM-Sprachumsetzer für Java EE-Unterstützung definieren • SCLM-Typen für Java EE-Unterstützung definieren 	Kapitel 4, „SCLM Developer Toolkit (optional)“, auf Seite 99
CICS-TS-Administrator (BIDI)	<ul style="list-style-type: none"> • JCL für die CICS-Region aktualisieren • CSD für die CICS-Region aktualisieren • CICS-Gruppe definieren • CICS-Transaktionsnamen definieren • Programm für CICS definieren • Debugger für CICS definieren 	<ul style="list-style-type: none"> • Kapitel 5, „(Optional) Application Deployment Manager (veraltet)“, auf Seite 107 • „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 131 • „Integrated Debugger (optional)“ auf Seite 133
WLM-Administrator	<ul style="list-style-type: none"> • Ziele für Developer for System z-Tasks zuweisen 	<ul style="list-style-type: none"> • "Hinweise zu WLM" in <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02)
LDAP	Gruppen für Push-to-Client definieren	Abschnitt "Hinweise zu Push-to-Client" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489)

Hinweise zur Konfigurationsvorbereitung

Informationen zu Developer for System z und zur Interaktion des Produkts mit Ihrem System sowie zu den vorausgesetzten und zusätzlich erforderlichen Produkten finden Sie in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02). Diese Informationen können Ihnen dabei helfen, eine Konfiguration zu erstellen, die Ihre aktuellen Anforderungen sowie ein zukünftiges Wachstum unterstützen.

Workload-Management

Im Gegensatz zu herkömmlichen z/OS-Anwendungen ist Developer for System z keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for System z besteht aus mehreren Komponenten, die interagieren, um dem Client den Zugriff auf die Hostsystemservices und -daten zu ermöglichen. Informationen zur Planung Ihrer WLM-Konfiguration finden Sie in "Hinweise zu WLM" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

Anmerkung: Developer for System z enthält mehrere Tasks, die miteinander und mit dem Client kommunizieren. Diese Tasks verwenden verschiedene Zeitgeber, um Kommunikationsunterbrechungen mit ihren Partnern festzustellen. Auf Systemen mit hoher CPU-Belastung oder bei falschen WLM-Einstellungen (Auslastungsverwaltung) für Developer for System z können Zeitlimitprobleme auftreten (weil während des Zeitlimitfensters zu wenig CPU-Zeit verfügbar ist).

Ressourcennutzung und Systembegrenzungen

Developer for System z verwendet eine variable Anzahl von Systemressourcen wie Adressräumen und z/OS UNIX-Prozesse und -Threads. Die Verfügbarkeit dieser Ressourcen ist durch verschiedene Systemdefinitionen begrenzt. Informationen zur Schätzung der Auslastung von wichtigen Ressourcen für die Planung Ihrer Systemkonfiguration finden Sie in "Optimierungsaspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02). Developer for System z kann entweder im 31-Bit- oder im 64-Bit-Modus ausgeführt, wodurch sich die Speicherressourcenbegrenzungen erheblich ändern.

Erforderliche Konfiguration für vorausgesetzte Produkte

Fragen Sie bei Ihrem MVS-Systemprogrammierer, beim Sicherheitsadministrator und beim TCP/IP-Administrator nach, ob die vorausgesetzten Produkte und die erforderliche Software installiert und getestet sind und funktionieren. Nachfolgend sind einige erforderliche Anpassungstasks aufgelistet, die leicht übersehen werden können:

- Alle Developer for System z-Benutzer müssen für die Java-Verzeichnisse die Zugriffsrechte READ und EXECUTE haben.
- Ferne (hostbasierte) Aktionen für z/OS UNIX-Unterprojekte erfordern, dass auf dem Hostsystem die z/OS UNIX-Version von REXEC oder SSH aktiv ist.

Hinweise zur Benutzer-ID

Die Benutzer-ID eines Developer for System z-Benutzers muss mindestens die folgenden Attribute haben:

- TSO-Zugriff (mit normaler Regionsgröße)

Anmerkung: Für die Benutzer-ID, die die Installationsprüfprogramme (Installation Verification Programs, IVPs) ausführt, ist eine große Regionsgröße erforder-

lich, weil speicherintensive Funktionen (beispielsweise Java) ausgeführt werden. Sie sollten die Regionsgröße auf 131.072 Kilobyte (128 Megabyte) oder mehr setzen.

- Ein für das Sicherheitssystem (z. B. RACF) definiertes OMVS-Segment für die Benutzer-ID und für die zugehörige Standardgruppe
 - Das Feld HOME muss auf ein dem Benutzer zugeordnetes Ausgangsverzeichnis (mit den Zugriffsrechten READ, WRITE und EXECUTE) verweisen.
 - Das Feld PROGRAM im OMVS-Segment sollte auf /bin/sh oder eine andere gültige z/OS UNIX-Shell, z. B. /bin/tcsh, gesetzt sein.
 - Das Feld ASSIZEMAX sollte nicht gesetzt sein, sodass Systemstandardwerte verwendet werden.
 - UID 0 ist für die Benutzer-ID nicht erforderlich.

Beispiel (Befehl **LISTUSER userid NORACF OMVS**):

USER=userid

```
OMVS INFORMATION
-----
UID= 0000003200
HOME= /u/userid
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMA= NONE
PROCSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
```

- Für die Standardgruppe, zu der die Benutzer-ID gehört, ist eine Gruppen-ID (GID) erforderlich.

Beispiel (Befehl **LISTGRP group NORACF OMVS**):

GROUP group

```
OMVS INFORMATION
-----
GID= 0000003243
```

- Zugriffsrechte READ und EXECUTE für das Installations- und Konfigurationsverzeichnis sowie die Installations- und Konfigurationsdateien von Developer for System z (standardmäßig /usr/lpp/rdz/*, /etc/rdz/* und /var/rdz/*)
- Zugriffsrechte READ, WRITE und EXECUTE für das Verzeichnis WORKAREA von Developer for System z (Standard: /var/rdz/WORKAREA) und für das Benutzerprotokollverzeichnis (Standard: /var/rdz/logs).
- Zugriffsrecht READ für die Installationsdateien von Developer for System z (standardmäßig FEK.SFEK*)
- Zugriffsrechte READ, WRITE und EXECUTE für das Verzeichnis /tmp oder ein Verzeichnis, auf das in der Umgebungsvariablen TMPDIR verwiesen wird.

Hinweise zum Server

Developer for System z besteht aus mehreren permanent aktiven Servern, bei denen es sich um gestartete Tasks oder Benutzerjobs handeln kann. Diese Server stellen selbst die erforderlichen Services bereit oder starten dazu andere Server (z. B. z/OS UNIX-Threads oder -Benutzerjobs). Es gibt keine bestimmte Startreihenfolge. Die einzige Bedingung ist, dass die Server betriebsbereit sein müssen, bevor der erste Benutzer versucht, eine Verbindung herzustellen. Die Dateien und Dateisysteme, in denen sich die von Developer for System z-Servern und -Services verwendeten Sicherheitsmechanismen befinden, müssen sicher sein. Dies bedeutet, dass

nur vertrauenswürdige Systemadministratoren berechtigt sein sollten, die Programmbibliotheken und Konfigurationsdateien zu aktualisieren.

- Der Debug-Manager (DBGMGR) stellt Services für das Debugging bereit.
- JMON (JES Job Monitor) stellt alle Services mit Bezug zum JES bereit.
- RSE (Remote Systems Explorer) stellt Kernservices wie den Verbindungsaufbau vom Client zum Hostsystem und das Starten anderer Server für bestimmte Services bereit. RSE umfasst zwei logische Einheiten:
 - RSE-Dämon (RSED) für die Verwaltung der Verbindungskonfiguration und die Ausführung im Einzelservermodus
 - RSE-Server für die einzelnen Clientanforderungen

Wie in "TCP/IP-Ports" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02) dokumentiert, muss der Client zu bestimmten Hostservices und somit zu den zugehörigen Ports eine Verbindung herstellen können. Diese Services und Ports müssen deshalb in der Firewall, die das Hostsystem schützt, definiert sein. An allen anderen von Developer for System z verwendeten Ports gibt es nur Hostdatenverkehr. Nachfolgend sind die Ports aufgelistet, die für die externe Kommunikation in einer Basiskonfiguration von Developer for System z erforderlich sind.

- RSE-Dämon für die Einrichtung der Client-Host-Kommunikation (über TCP). Standardport: 4035.
- RSE-Server für die Client-Host-Kommunikation (über TCP). Standardmäßig kann jeder verfügbare Port verwendet werden. Die verfügbaren Ports können jedoch auf einen bestimmten Portbereich eingeschränkt werden.

Konfigurationsmethode

Developer for System z stellt alternative Methoden zur Konfiguration der Hostsystemseite des Produkts bereit. Dabei handelt es sich um die folgenden Methoden:

- Verwendung der Dienstprogramms zur Hostkonfiguration. Diese ISPF-Anzeigeanwendung führt Sie durch die erforderlichen und die ausgewählten optionalen Anpassungsschritte. Weitere Informationen finden Sie in *Host Configuration Utility* (IBM Form SC12-4472).
- Verwendung des *Leitfadens für den Schnelleinstieg in die Hostkonfiguration*. Dieses Handbuch enthält eine Anleitung durch die erforderlichen Anpassungsschritte. Der Inhalt dieses Handbuchs ist auf eine Basiskonfiguration beschränkt.
- Verwendung des Handbuchs "*Hostkonfiguration*". Dieses Handbuch führt Sie durch die erforderlichen und alle optionalen Anpassungsschritte. In diesem Handbuch sind alle konfigurierbaren Optionen sowie einige vom Standard abweichende Szenarios beschrieben.

Hinweise zur Implementierungsvorbereitung

Developer for System z unterstützt das Klonen einer Installation auf einem anderen System, sodass Sie nicht auf jedem System eine SMP/E-Installation durchführen müssen.

Für die Implementierung (Deployment) auf anderen Systemen sind die nachfolgenden Dateien und Verzeichnisse obligatorisch. Falls Sie eine Datei an eine andere Position kopiert haben, muss die entsprechende Datei in den folgenden Listen durch Ihre angepasste Datei ersetzt werden.

Anmerkung: Die folgende Auflistung umfasst nicht die für die Implementierung vorausgesetzte und zusätzlich erforderliche Software.

Developer for System z

- FEK.SFEKAUTH(*)
- FEK.SFEKLOAD(*)
- FEK.SFEKPROC(*)
- FEK.#CUST.PARMLIB(*)
- FEK.#CUST.PROCLIB(*)
- /usr/lpp/rdz/*
- /etc/rdz/*
- /var/rdz/* (nur Verzeichnisstruktur)
- Optionale Komponenten:
 - FEK.SFEKLPA(*)
 - FEK.SFEKLMOD(*)
 - FEK.#CUST.CNTL(*)
 - Definitionen, Dateien und Verzeichnisse im Ergebnis der Anpassung von Jobs in FEK.#CUST.JCL

Developer for System z Host Utilities

- AKG.SAKGPROC(*)
- /usr/lpp/rdzutil/*

Anmerkung:

- FEK und /usr/lpp/rdz sind das während der Installation von Developer for System z verwendete übergeordnete Qualifikationsmerkmal und der Pfad. FEK.#CUST, /etc/rdz und /var/rdz sind die während der Anpassung des Produkts verwendeten Standardpositionen. (Weitere Informationen hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 17.)
- AKG und /usr/lpp/rdzutil sind das während der Installation von Developer for System z Host Utilities verwendete übergeordnete Qualifikationsmerkmal und der Pfad.
- Sie sollten Developer for System z in einem privaten Dateisystem (HFS oder zFS) installieren, um die Implementierung der z/OS UNIX-Produktkomponenten zu vereinfachen.
- Wenn Sie kein privates Dateisystem verwenden können, verwenden Sie für den Transport der z/OS UNIX-Verzeichnisse von einem System zu einem anderen ein Archivierungstool wie den z/OS UNIX-Befehl tar. Mit dieser Methode lassen sich die Attribute (z. B. für die Programmsteuerung) für die Dateien und Verzeichnisse von Developer for System z beibehalten.

Weitere Informationen zu den folgenden Beispielbefehlen für die Archivierung und Wiederherstellung des Installationsverzeichnisses von Developer for System z finden Sie in der Veröffentlichung *UNIX System Services Command Reference* (SA22-7802).

 - Archivierung: cd /SYS1/usr/lpp/rdz; tar -cSf /u/userid/rdz.tar
 - Wiederherstellung: cd /SYS2/usr/lpp/rdz; tar -xSf /u/userid/rdz.tar

Clientprüfliste

Benutzer der Clientkomponente von Developer for System z müssen das Ergebnis bestimmter Hostsystemanpassungen, z. B. der TCP/IP-Portnummern, kennen, damit der Client fehlerfrei funktioniert. Verwenden Sie diese Prüflisten für die erforderlichen Informationen.

Die Prüfliste in Tabelle 5 enthält die erforderlichen Ergebnisse obligatorischer Anpassungsschritte. In Tabelle 6 sind die erforderlichen Ergebnisse optionaler Anpassungsschritte aufgelistet.

Tabelle 5. Clientprüfliste: obligatorischer Teil

Anpassung	Wert
TCP/IP-Portnummer des RSE-Dämons. Die Standardeinstellung ist 4035. Lesen Sie hierzu „RSED, gestartete Task für den RSE-Dämon“ auf Seite 28.	

Tabelle 6. Clientprüfliste: optionaler Teil

Anpassung	Wert
Position der ELAXF*-Prozeduren, falls sie nicht in einer Systemprozedurbibliothek enthalten sind. Die Standardeinstellung ist FEK.#CUST.PROCLIB. Lesen Sie die Anmerkung zu JCLLIB in „ELAXF*-Prozeduren für ferne Build-Erstellung“ auf Seite 31.	
Namen der ELAXF*-Prozeduren oder der zugehörigen Prozedurschritte, sofern sie geändert wurden. Lesen Sie die Anmerkung zur Änderung der Namen im Abschnitt „ELAXF*-Prozeduren für ferne Build-Erstellung“ auf Seite 31.	
Position der AKGCR-Prozedur, sofern sie nicht in einer Systemprozedurbibliothek enthalten ist. Die Standardeinstellung ist AKG.#CUST.PROCLIB. Lesen Sie die Anmerkung zu JCLLIB im Abschnitt „Codeüberprüfung“ auf Seite 115.	
Position der AKGCC-Prozedur, sofern sie nicht in einer Systemprozedurbibliothek enthalten ist. Die Standardeinstellung ist AKG.#CUST.PROCLIB. Lesen Sie die Anmerkung zu JCLLIB im Abschnitt „Codeabdeckung“ auf Seite 116.	
Position der Exec FEKRNPLI des Include-Vorprozessors. Die Standardeinstellung ist FEK.#CUST.CNTL. Lesen Sie hierzu „Unterstützung für Include-Vorprozessor (optional)“ auf Seite 129.	
Position der Debugger-Lademodule, falls diese nicht in LINKLIST enthalten sind. Die Standardeinstellung ist FEK.SFEKAUTH. Siehe „Integrated Debugger (optional)“ auf Seite 133	

Tabelle 6. Clientprüfliste: optionaler Teil (Forts.)

Anpassung	Wert
<p>Position der Lademodule für den Komponententext, falls diese nicht in LINKLIST oder STEPLIB von rsed.envvars enthalten sind. Die Standardeinstellung ist FEK.SFEKLOAD.</p> <p>Lesen Sie hierzu „xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)“ auf Seite 130.</p>	
<p>Position der AZUZUNIT-Prozedur, sofern sie nicht in einer Systemprozedurbibliothek enthalten ist. Die Standardeinstellung ist FEK.#CUST.PROCLIB.</p> <p>Lesen Sie die Anmerkung zu JCLLIB im Abschnitt „xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)“ auf Seite 130.</p>	
<p>Position der Beispiel-XML-Dateien *.xsd und *.xsl, die für die Formatierung von Komponententestausgaben verwendet werden. Die Standardwerte sind /usr/lpp/rdz/samples/zunit/xsd und /usr/lpp/rdz/samples/zunit/xsl.</p> <p>Lesen Sie hierzu „xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)“ auf Seite 130.</p>	
<p>TN3270-Portnummer für den Host-Connect-Emulator (zusätzliche Voraussetzung). Die Standardeinstellung ist 23.</p> <p>Weitere Informationen finden Sie in "TCP/IP-Ports" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02).</p>	
<p>REXEC- oder SSH-Portnummer. Standardwert: 512 bzw. 22 (zusätzliche Voraussetzung).</p> <p>Lesen Sie hierzu „(Optional) z/OS UNIX-Unterprojekte“ auf Seite 128.</p>	
<p>Portnummer des Debug Tool-Servers (zusätzliche Voraussetzung; ohne Standardwert).</p> <p>Lesen Sie hierzu „Unterstützung für DB2- und IMS-Debug (optional)“ auf Seite 137.</p>	
<p>Portnummer für Application Deployment Manager (ADM). Standardwert: 5129 für Web-Service und 5130 für REST-Service.</p> <p>Weitere Informationen finden Sie in "TCP/IP-Ports" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02).</p>	
<p>Position der Beispielbibliothek SFEKSAMP für CARMA-RAM-Muster. Der Standardwert ist FEK.SFEKSAMP.</p> <p>Siehe <i>CARMA Developer's Guide</i> (IBM Form SC23-7660).</p>	
<p>Position der CRA#ASLM-JCL für das Zuordnen von CARMA-SCLM-RAM-Dateien. Der Standardwert ist FEK.#CUST.JCL.</p> <p>Lesen Sie die Anmerkung zu CRA#ASLM in „SCLM-RAM“ auf Seite 87.</p>	

Kapitel 2. Basisanpassung

Die folgenden Anpassungsschritte beziehen sich auf eine Basiskonfiguration von Developer for System z. Informationen zu den Voraussetzungen für die Anpassung optionaler Komponenten finden Sie in den Kapiteln zu diesen Komponenten.

Voraussetzungen und Prüfliste

Für diese Anpassungstask, für die die folgenden Ressourcen und speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines Sicherheitsadministrators und eines TCP/IP-Administrators:

- Datei mit APF-Berechtigung
- Verschiedene Aktualisierungen von PARMLIB
- Verschiedene Aktualisierungen der Sicherheitssoftware
- Unterschiedliche TCP/IP-Ports für interne und Client-Host-Kommunikation
- Einleitendes Programmladen zum Aktivieren eines optionalen SVC

Führen Sie zur Überprüfung der Installation und zur Verwendung von Developer for System z an Ihrem Standort die folgenden Tasks aus. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Erstellen Sie anpassbare Kopien der Beispiele sowie die Arbeitsumgebung für Developer for System z. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“.
2. Aktualisieren Sie die z/OS UNIX-Systembegrenzungen, starten Sie gestartete Tasks und definieren Sie über APF autorisierte Dateien und LINKLIST-Dateien sowie SVC- und LPA-Dateien (optional). Weitere Details enthält der Abschnitt „PARMLIB-Änderungen“ auf Seite 19.
3. Erstellen Sie gestartete Taskprozeduren sowie Kompilierungs- und Verknüpfungsprozeduren. Weitere Details enthält der Abschnitt „PROCLIB-Änderungen“ auf Seite 26.
4. Aktualisieren Sie die Sicherheitsdefinitionen. Weitere Details enthält der Abschnitt „Sicherheitsdefinitionen“ auf Seite 33. Zur Einrichtung der Threadsicherheit müssen Sie wissen, wie PassTickets verwendet werden. Weitere Informationen finden Sie in "PassTickets verwenden" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
5. Passen Sie die Konfigurationsdateien für Developer for Developer for System z an. Weitere Details enthalten die folgenden Abschnitte:
 - „FEJJCENFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34
 - „RSE-Konfigurationsdatei rsed.envvars“ auf Seite 39
 - „ISPF.conf - Konfigurationsdatei für TSO/ISPF Client Gateway“ auf Seite 59

Anpassungskonfiguration

In Developer for System z sind verschiedene Beispielkonfigurationsdateien und Beispiel-JCL enthalten. Um das Überschreiben Ihrer Anpassungen bei einer Wartung zu vermeiden, kopieren Sie alle diese Member und z/OS UNIX-Dateien an eine andere Position und ändern Sie die Kopie.

Einige Funktionen von Developer for System z erfordern das Vorhandensein bestimmter Verzeichnisse in z/OS UNIX. Sie müssen diese Verzeichnisse bei der Anpassung des Produkts erstellen. Zur Vereinfachung der Installation steht der Beispieljob FEKSETUP bereit, mit dem Sie die Kopien und die erforderlichen Verzeichnisse erstellen können.

Anmerkung: Im Handbuch *Rational Developer for System z Host Configuration Utility* (IBM Form SC12-4472) wird die Hostsystemkonfiguration mit Host Configuration Utility beschrieben. Der Job FEKSETUP und das Dienstprogramm führen einige gleiche Tasks aus, wobei nicht überprüft werden kann, ob diese Tasks bereits ausgeführt wurden. Deshalb ist es möglich, dass Änderungen rückgängig gemacht werden, die bereits vorgenommen wurden. Verwenden Sie aus diesem Grund nicht beide Methoden in derselben Installation.

Zum Erstellen anpassbarer Kopien von Konfigurationsdateien und der Konfigurations-JCL sowie zum Erstellen der erforderlichen z/OS UNIX-Verzeichnisse passen Sie das Beispielmembert FEKSETUP in der Datei FEK.SFEKSAMP an. Die notwendigen Anpassungsschritte sind innerhalb des Members beschrieben.

Dieser Job führt die folgenden Tasks aus:

- FEK.#CUST.PARMLIB erstellen und mit Beispielkonfigurationsdateien füllen
- FEK.#CUST.PROCLIB erstellen und mit SYS1.PROCLIB-Beispiel-Membert füllen
- FEK.#CUST.JCL erstellen und mit Beispielkonfigurations-JCL füllen
- FEK.#CUST.CNTL erstellen und mit Beispiel-Scripts für den Serverstart füllen
- FEK.#CUST.ASM erstellen und mit Assemblerbeispielquellcode füllen
- FEK.#CUST.COBOLE erstellen und mit COBOLE-Beispielquellcode füllen
- FEK.#CUST.SQL erstellen und mit SQL-Beispielbefehlsdateien füllen
- /etc/rdz/* erstellen und mit Beispielkonfigurationsdateien füllen
- /var/rdz/* als Arbeitsverzeichnisse für verschiedene Funktionen von Developer for System z erstellen und mit Beispieldateien füllen

Anmerkung:

- Für die Konfigurationsschritte in dieser Veröffentlichung werden die vom Job FEKSETUP erstellten Member-Positionen und Dateipositionen verwendet, sofern nichts anderes angegeben ist. Die Originalbeispiele, die nicht aktualisiert werden sollten, befinden sich in FEK.SFEKSAMP und /usr/lpp/rdz/samples/.
- Weitere Details dazu, welche Beispielmembert in welche Dateien kopiert werden, sowie weitere Details dazu, welche Verzeichnisse erstellt werden, ihre Berechtigungsbitmasken und wohin die verschiedenen Beispieldateien kopiert werden, finden Sie in den Kommentaren in FEK.SFEKSAMP (FEKSETUP).
- Zur Unterstützung der Migration einer vorhandenen Konfiguration dokumentieren die Kommentare in FEK.SFEKSAMP (FEKSETUP) auch die Änderungen zwischen den verschiedenen Versionen von Developer for System z.
- Falls Sie alle z/OS UNIX-Dateien von Developer for System z in demselben Dateisystem (HFS oder zFS) behalten wollen, die Konfigurationsdateien aber auch im Verzeichnis /etc/rdz enthalten sein sollen, können Sie das Problem mit symbolischen Links lösen. Die folgenden z/OS UNIX-Beispielbefehle erstellen im vorhandenen Dateisystem ein neues Verzeichnis (/usr/lpp/rdz/cust) und definieren eine symbolische Verbindung (/etc/rdz) zu diesem Verzeichnis:

```
mkdir /usr/lpp/rdz/cust
ln -s /usr/lpp/rdz/cust /etc/rdz
```

PARMLIB-Änderungen

In diesem Abschnitt werden die folgenden PARMLIB-Änderungen dokumentiert:

- „z/OS UNIX-Grenzwerte in BPXPRMxx festlegen“
- „Gestartete Tasks zu COMMNDxx hinzufügen“ auf Seite 20
- „SVC-Definitionen in IEASVCxx“ auf Seite 21
- „LPA-Definitionen in LPALSTxx“ auf Seite 21
- „APF-Berechtigungen in PROGxx“ auf Seite 22
- „LINKLIST-Definitionen in PROGxx“ auf Seite 23
- „Vorausgesetzte LINKLIST- und LPA-Definitionen“ auf Seite 24
- „LINKLIST-Definitionen für andere Produkte“ auf Seite 26

Weitere Informationen zu den in den nächsten Abschnitten aufgelisteten PARMLIB-Definitionen finden Sie in der Veröffentlichung *MVS Initialization and Tuning Reference* (SA22-7592). Weitere Informationen zu den Beispielkonsolenbefehlen finden Sie in der Veröffentlichung *MVS System Commands* (SA22-7627).

z/OS UNIX-Grenzwerte in BPXPRMxx festlegen

RSE (Remote Systems Explorer), mit dem Kernservices wie der Verbindungsaufbau vom Client zum Hostsystem bereitgestellt werden, ist ein auf z/OS UNIX basierender Prozess. Daher ist es wichtig, richtige Werte für die z/OS UNIX-Systemgrenzwerte in BPXPRMxx festzulegen. Diese basieren auf der Anzahl der gleichzeitig aktiven Benutzer in Developer for System z und auf ihrer durchschnittlichen Arbeitslast. Definieren Sie OMVS=xx im Parmlib-Member IEASYSxx, um festzulegen, welches Parmlib-Member BPXPRMxx beim einleitenden Programmieren verwendet werden soll.

Weitere Informationen zu verschiedenen definierten BPXPRMxx-Grenzwerten und ihrer Auswirkung auf Developer for System z finden Sie in "Optimierungsaspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

MAXASSIZE gibt die maximale Regionsgröße des Adressbereichs/Adressierungsprozesses an. Setzen Sie MAXASSIZE in SYS1.PARMLIB(BPXPRMxx) auf 2G. Dies ist der zulässige Maximalwert. Dieser Grenzwert gilt systemweit. Er ist daher für alle z/OS UNIX-Adressräume aktiv. Wenn Sie dies nicht wünschen, können Sie den Grenzwert in Ihrer Sicherheitssoftware auch nur für Developer for System z festlegen. Dies wird in „Gestartete Tasks für Developer for System z definieren“ auf Seite 160 beschrieben.

MAXTHREADS gibt die maximale Anzahl der aktiven Threads für einen einzelnen Prozess an. Setzen Sie MAXTHREADS in SYS1.PARMLIB(BPXPRMxx) auf mindestens 1500. Dieser Grenzwert gilt systemweit. Er ist daher für alle z/OS UNIX-Adressräume aktiv. Wenn Sie dies nicht wünschen, können Sie den Grenzwert in Ihrer Sicherheitssoftware auch nur für Developer for System z festlegen. Dies wird in „Gestartete Tasks für Developer for System z definieren“ auf Seite 160 beschrieben.

MAXTHREADTASKS gibt die maximale Anzahl der aktiven MVS-Tasks für einen einzelnen Prozess an. Setzen Sie MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) auf mindestens 1500. Dieser Grenzwert gilt systemweit. Er ist daher für alle z/OS UNIX-Adressräume aktiv. Wenn Sie dies nicht wünschen, können Sie den Grenzwert in Ihrer Sicherheitssoftware auch nur für Developer for System z festlegen. Dies wird in „Gestartete Tasks für Developer for System z definieren“ auf Seite 160 beschrieben.

MAXPROCUSER gibt die maximale Anzahl der Prozesse an, die für eine einzelne z/OS UNIX-Benutzer-ID gleichzeitig aktiv sein dürfen. Setzen Sie MAXPROCUSER in SYS1.PARMLIB(BPXPRMxx) auf mindestens 50. Diese Einstellung ist als systemweiter Grenzwert vorgesehen, weil er für alle Clients aktiv sein sollte, die Developer for System z verwenden.

Diese Werte können mit folgenden Konsolenbefehlen überprüft und dynamisch (bis zum nächsten einleitenden Programmladen) gesetzt werden:

- DISPLAY OMVS,0
- SETOMVS MAXASSIZE=2G
- SETOMVS MAXTHREADS=1500
- SETOMVS MAXTHREADTASKS=1500
- SETOMVS MAXPROCUSER=50

Anmerkung:

- Weitere Informationen zu anderen Positionen, an denen die Größe des Adressraums festgelegt oder begrenzt werden kann, finden Sie in "Größe des Adressraums" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Der hier gezeigte Wert für MAXPROCUSER basiert auf Benutzern, die eine eindeutige z/OS UNIX-Benutzer-ID (UID) haben. Erhöhen Sie diesen Wert, falls mehrere Benutzer eine UID gemeinsam verwenden.
- Stellen Sie sicher, dass andere BPXPRMxx-Werte – wie der für MAXPROCSYS und MAXUIDS – für die erwartete Anzahl gleichzeitig aktiver Benutzer von Developer for System z ausreichen. Weitere Details hierzu finden Sie in "Optimierungsaspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Bei der SMP/E-Installation von Developer for System z wurde empfohlen, den Code in einem separaten Dateisystem (zFS oder HFS) zu platzieren und BPXPRMxx zu aktualisieren, um dieses Dateisystem beim System-IPL anzuhängen. Nachfolgend wird noch einmal ein Beispielmountbefehl aufgeführt, falls diese Aktualisierung noch erfolgen muss:

```
MOUNT FILESYSTEM('#dsn')
  MOUNTPPOINT('-PathPrefix-usr/lpp/rdz')
  MODE(RDWR) /* kann MODE(READ) sein. */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, mit Erweiterungen */
/* TYPE(HFS) */ /* HFS, autom. Erweiterung */
```

- Bei der SMP/E-Installation von Developer for System z Host Utilities wurde empfohlen, den Code in einem separaten Dateisystem (zFS oder HFS) zu platzieren und BPXPRMxx zu aktualisieren, um dieses Dateisystem beim System-IPL anzuhängen. Nachfolgend wird noch einmal ein Beispielmountbefehl aufgeführt, falls diese Aktualisierung noch erfolgen muss:

```
MOUNT FILESYSTEM('#dsn')
  MOUNTPPOINT('-PathPrefix-usr/lpp/rdzutil')
  MODE(RDWR) /* kann MODE(READ) sein. */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, mit Erweiterungen */
/* TYPE(HFS) */ /* HFS, autom. Erweiterung */
```

Gestartete Tasks zu COMMNDxx hinzufügen

Fügen Sie Startbefehle für die RSED- und JMON-Server von Developer for System z zu SYS1.PARMLIB(COMMANDxx) hinzu, um sie beim nächsten einleitenden Programmladen des Systems automatisch zu starten. Definieren Sie CMD=xx im Parmlib-Member IEASYSxx, um anzugeben, welches Parmlib-Member COMMNDxx beim einleitenden Programmladen verwendet werden sollte.

Für den optionalen Integrated Debugger muss der DBGMR-Server von Developer for System z auf Ihrem System aktiv sein.

Nachdem die Server definiert und konfiguriert sind, können sie mit den folgenden Konsolenbefehlen dynamisch (bis zum nächsten einleitenden Programmladen) gestartet werden:

- S RSED
- S JMON
- S DBGMR

Anmerkung: Für die Server gibt es keine bestimmte Startreihenfolge. Die einzige Bedingung ist, dass die Server betriebsbereit sein müssen, bevor der erste Benutzer versucht, eine Verbindung herzustellen.

SVC-Definitionen in IEASVCxx

Der optionale Integrated Debugger kann Fehler in CICS-Transaktionen beheben, die in den Nur-Lese-Speicher geladen wurden. Hierzu muss ein Supervisoraufruf (SVC - Supervisor Call) für Developer for System z für Ihr System definiert werden.

Durch die Installation festgelegte SVCs sind in SYS1.PARMLIB(IEASVCxx) definiert. Für ihre Verwendung muss ein IPL aktiviert werden. Das zugehörige Lademodul muss beim einleitenden Programmladen in den Link-Pack-Bereich geladen werden. Definieren Sie SVC=xx im Parmlib-Member IEASYSxx, um anzugeben, welches Parmlib-Member IEASVCxx beim einleitenden Programmladen verwendet werden soll.

Geben Sie Folgendes in IEASVCxx an, um den SVC für Developer for System z zu definieren:

```
SVC Parm 251,REPLACE,TYPE(4),EPNAME(AQESVC01) /* RDz debug */
```

Wenn Sie nicht die standardmäßige SVC-Anzahl verwenden, ändern Sie den Wert 251 in der SVC Parm-Definition in die ausgewählte Anzahl und aktualisieren Sie den Startparameter SVC im JCL-Code für die gestartete DBGMR-Task.

LPA-Definitionen in LPALSTxx

Der optionale Service von Common Access Repository Manager (CARMA) unterstützt unterschiedliche Methoden für den Serverstart des CARMA-Servers. Bei der Startmethode "CRASTART" wird vorausgesetzt, dass sich das Modul CRASTART der Ladebibliothek FEK.SFEKLPA im Link-Pack-Bereich (LPA) befindet.

Der optionale Integrated Debugger kann Fehler in CICS-Transaktionen beheben, die in den Nur-Lese-Speicher geladen wurden. Hierzu muss sich das Lademodul AQESVC01 in der Ladebibliothek FEK.SFEKLPA beim einleitenden Programmladen im Link-Pack-Bereich (LPA) befinden.

LPA-Dateigruppen sind in SYS1.PARMLIB(LPALSTxx) definiert. Definieren Sie LPA=xx im Parmlib-Member IEASYSxx, um anzugeben, welches Parmlib-Member LPALSTxx beim einleitenden Programmladen verwendet werden soll.

LPA-Definitionen können mit folgendem Konsolenbefehl dynamisch (bis zum nächsten einleitenden Programmladen) gesetzt werden:

- SETPROG LPA,ADD,DSN=FEK.SFEKLPA

Anmerkung:

- In LPALSTxx aufgeführte Dateien müssen im Masterkatalog oder in einem Benutzerkatalog katalogisiert sein, der im LPALSTxx-Member angegeben ist.
- Für das Hinzufügen einer neuen Datei zu LPALSTxx muss ein IPL mit CLPA (Create Link Pack Area) aktiviert werden.
- Alle in LPA geladenen Bibliotheken werden automatisch als von APF autorisierte und programmgesteuerte Bibliotheken betrachtet. Stellen Sie sicher, dass Sie über die geeigneten Sicherheitsmaßnahmen für diese Bibliotheken verfügen.
- Wenn Sie keine für die LPA-Platzierung entwickelte Bibliothek in LPA speichern und stattdessen LINKLIST oder STEPLIB verwenden, müssen Sie sicherstellen, dass Sie die APF-Autorisierung und den Status der Programmsteuerung definieren.

APF-Berechtigungen in PROGxx

Damit JES Job Monitor auf JES-Spooldateien zugreifen kann, müssen das Modul FEJMON in der Ladebibliothek FEK.SFEKAUTH und die Language Environment-Laufzeitbibliotheken (LE-Laufzeitbibliotheken: CEE.SCEERUN*) eine APF-Berechtigung haben.

Damit mit dem optionalen Debug-Manager gearbeitet werden kann, muss das Modul AQEZPCM in der Ladebibliothek FEK.SFEKAUTH über eine APF-Berechtigung verfügen.

Damit der optionale Service von SCLM Developer Toolkit funktioniert, muss die REXX-Laufzeitbibliothek (REXX.*.SEAGLPA) über eine APF-Berechtigung verfügen.

Damit ISPF das TSO/ISPF-Client-Gateway erstellen kann, muss das Modul ISPZTS0 in SYS1.LINKLIB eine APF-Berechtigung haben. Das TSO/ISPF-Client-Gateway wird von TSO Commands Service und SCLM Developer Toolkit von Developer for System z verwendet.

APF-Berechtigungen werden standardmäßig in SYS1.PARMLIB(PROGxx) definiert. Definieren Sie PROG=xx im Parmlib-Member IEASYSxx, um anzugeben, welches Parmlib-Member PROGxx beim einleitenden Programm laden verwendet werden soll.

APF-Berechtigungen können mit den folgenden Konsolenbefehlen dynamisch (bis zum nächsten einleitenden Programm laden) festgelegt werden, wobei volser der Datenträger ist, auf dem sich die Datei (Dataset) befindet, sofern sie nicht von SMS verwaltet wird:

- SETPROG APF,ADD,DSN=FEK.SFEKAUTH,SMS
- SETPROG APF,ADD,DSN=CEE.SCEERUN,VOL=volser
- SETPROG APF,ADD,DSN=CEE.SCEERUN2,VOL=volser
- SETPROG APF,ADD,DSN=REXX.V1R4M0.SEAGLPA,VOL=volser
- SETPROG APF,ADD,DSN=SYS1.LINKLIB,VOL=volser

Anmerkung:

- Wenn Sie die Alternativbibliothek für das REXX-Produktpaket verwenden, lautet der Standardname der REXX-Laufzeitbibliothek REXX.*.SEAGALT und nicht REXX.*.SEAGLPA wie im obigen Beispiel.
- LPA-Bibliotheken, wie z. B. REXX.*.SEAGLPA, erhalten automatisch eine APF-Berechtigung, wenn sie sich im LPA befinden und somit keine expliziten Definitionen benötigen.

- Für einige der zusätzlich erforderlichen Produkte, wie beispielsweise IBM File Manager, ist ebenfalls eine APF-Berechtigung erforderlich. Weitere Informationen hierzu finden Sie im Anpassungshandbuch zum jeweiligen Produkt.

LINKLIST-Definitionen in PROGxx

LINKLIST-Definitionen für Developer for System z können in drei Kategorien gruppiert werden:

- Developer for System z-Ladebibliotheken, die für Developer for System z-Funktionen benötigt werden. Diese Definitionen werden in diesem Abschnitt beschrieben.
- Vorausgesetzte Ladebibliotheken, die für Developer for System z-Funktionen benötigt werden. Diese Definitionen werden im Abschnitt „Vorausgesetzte LINKLIST- und LPA-Definitionen“ auf Seite 24 beschrieben.
- Developer for System z-Ladebibliotheken, die von anderen Produkten benötigt werden. Diese Definitionen werden im Abschnitt „LINKLIST-Definitionen für andere Produkte“ auf Seite 26 beschrieben.

Tabelle 7. Zuordnung von Lademodulen zu Funktionen

Ladebibliothek	Lademodule	Verwendung	STEPLIB
FEK.SFEKAUTH	AQE* und CEE*	„Integrated Debugger (optional)“ auf Seite 133	ELAXFGO-Prozedur oder CICS
	FEJJ*	„PROCLIB-Änderungen“ auf Seite 26 (Gestartete Task von JES Job Monitor)	Prozedur der gestarteten Task
FEK.SFEKLMOD	IRZ* und IIRZ*	„IRZ-Diagnosenachrichten für generierten Code (optional)“ auf Seite 132	CICS-, IMS- oder MVS-Batch
FEK.SFEKLOAD	AND*	Kapitel 5, „(Optional) Application Deployment Manager (veraltet)“, auf Seite 107	CICS
	AZU* und IAZU*	„xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)“ auf Seite 130	rsed.envvars oder MVS-Batch
	BWB*	Kapitel 4, „SCLM Developer Toolkit (optional)“, auf Seite 99	rsed.envvars
	CRA*	Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 63	CRASUB* oder crastart*.conf
	ELAX*	„ELAXF*-Prozeduren für ferne Build-Erstellung“ auf Seite 31 (Fehlerrückmeldung und Include-Vorprozessor)	ELAXF*-Prozeduren
	FEJB*	„Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 131	CICS

Tabelle 7. Zuordnung von Lademodulen zu Funktionen (Forts.)

Ladebibliothek	Lademodule	Verwendung	STEPLIB
FEK.SFEKLPA	CRA*	Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 63	CRASRV.properties

Damit die aufgelisteten Developer for System z-Services funktionieren, müssen alle in Tabelle 7 auf Seite 23 dokumentierten Module, die sich auf den Service beziehen, mithilfe von STEPLIB oder LINKLIST (oder LPA) verfügbar gemacht werden. Beachten Sie, dass die Bibliothek SFEKLM0D nicht von Developer for System z selbst verwendet wird, sondern von Code, der von Developer for System z generiert wurde. In der Spalte STEPLIB (falls Sie STEPLIB auswählen) in Tabelle 7 auf Seite 23 können Sie erfahren, an welcher Position die Definition STEPLIB (bzw. DFHRPL für CICS) vorgenommen werden muss. Sie müssen jedoch die folgenden Punkte beachten:

- Die Verwendung von STEPLIB unter z/OS UNIX wirkt sich negativ auf die Leistung aus.
- Wenn eine STEPLIB-Bibliothek eine APF-Berechtigung hat, ist diese Berechtigung für alle Bibliotheken erforderlich. Bibliotheken verlieren ihre APF-Berechtigung, wenn sie mit STEPLIB-Bibliotheken ohne APF-Berechtigung gemischt werden.
- Bibliotheken, die der DD-Anweisung STEPLIB in einer JCL hinzugefügt wurden, werden nicht an die z/OS UNIX-Prozesse weitergegeben, die durch die JCL gestartet wurden.

Wenn Sie sich an Ihrem Standort nach den IBM Empfehlungen gerichtet haben, sind die LINKLIST-Dateien in SYS1.PARMLIB(PROGxx) definiert. Definieren Sie PROG=xx im Parmlib-Member IEASYSxx, um anzugeben, welches Parmlib-Member PROGxx beim einleitenden Programmladen verwendet werden soll.

Die erforderlichen Definitionen sehen wie folgt aus, wobei listname der Name der zu aktivierenden LINKLIST-Gruppe ist und volser für den Datenträger steht, auf dem sich die Datei befindet, sofern sie nicht im Masterkatalog katalogisiert ist:

- LNKLIST ADD NAME(listname) DSN=FEK.SFEKAUTH) VOLUME(volser)
- LNKLIST ADD NAME(listname) DSN=FEK.SFEKLOAD)

LINKLIST-Definitionen können mit den folgenden Gruppen- oder Konsolenbefehlen dynamisch (bis zum nächsten einleitenden Programmladen) erstellt werden. Dabei steht volser für den Datenträger, auf dem sich die Dateien befinden, wenn sie nicht im Masterkatalog katalogisiert sind:

1. LNKLIST DEFINE,NAME=LLTMP,COPYFROM=CURRENT
2. LNKLIST ADD NAME=LLTMP,DSN=FEK.SFEKAUTH,VOL=volser
3. LNKLIST ADD NAME=LLTMP,DSN=FEK.SFEKLOAD
4. LNKLIST ACTIVATE,NAME=LLTMP

Vorausgesetzte LINKLIST- und LPA-Definitionen

Remote Systems Explorer (RSE) ist ein z/OS UNIX-Prozess, der auf MVS-Ladebibliotheken zugreifen muss. Die JES Job Monitor- und Integrated Debugger-Server benötigen ebenfalls Zugriff auf das System sowie auf LE- und C-Bibliotheken (LE - Language Environment). Die folgenden vorausgesetzten Bibliotheken müssen über STEPLIB oder LINKLIST/LPALIB verfügbar sein:

- Systemladebibliothek

- SYS1.LINKLIB
- LE-Laufzeit (Language Environment)
 - CEE.SCEERUN
 - CEE.SCEERUN2
- DLL-Klassenbibliothek von C++'s
 - CBC.SCLBDLL
- TSO/ISPF-Client-Gateway von ISPF
 - ISP.SISPLoad
 - ISP.SISPLPA

Zur Unterstützung optionaler Services müssen die folgenden Bibliotheken über STEPLIB oder LINKLIST/LPALIB verfügbar sein. Diese Liste enthält keine Dateien, die spezifisch für ein Produkt sind, mit dem Developer for System z interagiert, wie beispielsweise IBM File Manager for z/OS:

- REXX-Laufzeitbibliothek (für das SCLM Developer Toolkit und RSE_DSICALL=TSO)
 - REXX.*.SEAGLPA
- Systemladebibliothek (für SSL-Verschlüsselung)
 - SYS1.SIEALNKE
- Systemladebibliothek (für Integrated Debugger unter z/OS ab Version 1.13)
 - SYS1.SIEAMIGE
- Systemladebibliothek (für Enterprise COBOL- und PL/I-Einheitentest)
 - SYS1.CSSLIB
 - SYS1.SIXMLOD1

Anmerkung:

- Wenn Sie die Alternativbibliothek für das REXX-Produktpaket verwenden, ist der Standardname der REXX-Laufzeitbibliothek REXX.*.SEAGALT anstelle von REXX.*.SEAGLPA im vorherigen Beispiel.
- Alle in LPA geladenen Bibliotheken werden automatisch als von APF autorisierte und programmgesteuerte Bibliotheken betrachtet. Stellen Sie sicher, dass Sie über die geeigneten Sicherheitsmaßnahmen für diese Bibliotheken verfügen.
- Bibliotheken, die in den Link-Pack-Bereich (LPA) gestellt werden müssen, wie z. B. REXX.*.SEAGLPA, erfordern unter Umständen zusätzliche Programmsteuerberechtigungen oder APF-Berechtigungen, wenn für den Zugriff auf diese Bibliotheken LINKLIST oder STEPLIB verwendet wird.
- Für einige der zusätzlich erforderlichen Produkte, wie beispielsweise IBM File Manager, sind auch STEPLIB- oder LINKLIST/LPALIB-Definitionen erforderlich. Weitere Informationen hierzu finden Sie im Anpassungshandbuch zum jeweiligen Produkt.

LINKLIST-Dateigruppen werden standardmäßig in SYS1.PARMLIB(PROGxx) definiert. LPA-Dateigruppen sind in SYS1.PARMLIB(LPALSTxx) definiert.

Wenn Sie sich für die Verwendung von STEPLIB entscheiden, müssen Sie die nicht über LINKLIST/LPALIB verfügbaren Bibliotheken in der Anweisung STEPLIB der RSE-Konfigurationsdatei rsed.envvars definieren. Beachten Sie jedoch die folgenden Punkte:

- Die Verwendung von STEPLIB unter z/OS UNIX wirkt sich negativ auf die Leistung aus.

- Wenn eine STEPLIB-Bibliothek eine APF-Berechtigung hat, ist diese Berechtigung auch für alle anderen STEPLIB-Bibliotheken erforderlich. Bibliotheken verlieren ihre APF-Berechtigung, wenn sie mit STEPLIB-Bibliotheken ohne APF-Berechtigung gemischt werden.
- Bibliotheken, die der DD-Anweisung STEPLIB in einer JCL hinzugefügt wurden, werden nicht an die z/OS UNIX-Prozesse weitergegeben, die durch die JCL gestartet wurden.

LINKLIST-Definitionen für andere Produkte

Der Developer for System z-Client verfügt über eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools'. Alle IRZM*- und IIRZ*-Module in der Ladebibliothek FEK.SFEKLMOD müssen mithilfe von STEPLIB oder LINKLIST verfügbar gemacht werden, damit der generierte Code Diagnosefehlernachrichten ausgeben kann.

LINKLIST-Dateigruppen werden standardmäßig in SYS1.PARMLIB(PROGxx) definiert.

Wenn Sie sich für die Verwendung von STEPLIB entscheiden, müssen Sie die nicht über LINKLIST verfügbaren Bibliotheken in der Anweisung STEPLIB der Task definieren, die den Code (IMS oder Batch-Job) ausführt. Wenn eine STEPLIB-Bibliothek jedoch eine APF-Berechtigung hat, ist diese Berechtigung für alle STEPLIB-Bibliotheken erforderlich. Bibliotheken verlieren ihre APF-Berechtigung, wenn sie mit STEPLIB-Bibliotheken ohne APF-Berechtigung gemischt werden.

PROCLIB-Änderungen

In diesem Abschnitt werden die folgenden PROCLIB-Änderungen dokumentiert:

- „JMON, gestartete Task von JES Job Monitor“
- „DBGMR, gestartete Task für den Debug-Manager“ auf Seite 27
- „RSED, gestartete Task für den RSE-Dämon“ auf Seite 28
- „ELAXF*-Prozeduren für ferne Build-Erstellung“ auf Seite 31

In den folgenden Unterabschnitten stehen weitere Informationen zur Verfügung:

- „Einschränkungen der JCL für die PARM-Variable“ auf Seite 29
- „TMPDIR-Verarbeitung“ auf Seite 30

Die gestartete Task und die Prozeduren für ferne Builds, die in den folgenden Abschnitten aufgelistet sind, müssen sich in einer für Ihr JES definierten Systemprozedurbibliothek befinden. In den Anweisungen der folgenden Abschnitte wird die IBM Standardprozedurbibliothek SYS1.PROCLIB verwendet.

JMON, gestartete Task von JES Job Monitor

Passen Sie das Beispielmember FEK.#CUST.PROCLIB(JMON) der gestarteten Task wie innerhalb des Members beschrieben an und kopieren Sie es in SYS1.PROCLIB. Geben Sie die folgenden Informationen wie im folgenden Codebeispiel an:

- Das übergeordnete Qualifikationsmerkmal der (autorisierten) Ladebibliothek (standardmäßig FEK)
- Die Konfigurationsdatei für JES Job Monitor (standardmäßig FEK.#CUST.PARMLIB(FEJJCNFG))

```

/*
/* JES JOB MONITOR
/*
//JMON      PROC PRM=,                * PRM='-TV' TO START TRACING
//          LEPRM='RPTOPTS(ON)',
//          HLQ=FEK,
//          CFG=FEK.#CUST.PARMLIB(FEJJCNFG)
/*
//JMON      EXEC PGM=FEJJMON,REGION=0M,TIME=NOLIMIT,
//          PARM=('&LEPRM,ENVAR("_CEE_ENVFILE_S=DD:ENVIRON")/&PRM')
//STEPLIB DD DISP=SHR,DSN=&HLQ..SFEKAUTH
//ENVIRON DD DISP=SHR,DSN=&CFG
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//          PEND
/*

```

Abbildung 1. JMON: Gestartete Task von JES Job Monitor

Anmerkung:

- Weitere Informationen zu den Startparametern finden Sie in Kapitel 11, „Bedienerbefehle“, auf Seite 193.
- Die Beispiel-JCL hat ursprünglich den Namen FEK.SFEKSAMP(FEJJJCL) und wird wie in „Anpassungskonfiguration“ auf Seite 17 beschrieben in FEK.#CUST.PROCLIB(JMON) umbenannt.
- Die Trace-Erstellung kann auch mit Konsolenbefehlen gesteuert werden. Eine diesbezügliche Beschreibung enthält Kapitel 11, „Bedienerbefehle“, auf Seite 193.
- Informationen zu den empfohlenen Workload Manager-Zielen (WLM) für diese Task finden Sie in "Hinweise zu WLM" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

DBGMGR, gestartete Task für den Debug-Manager

Passen Sie das Beispielmembere FEK.#CUST.PROCLIB(DBGMGR) der gestarteten Task wie innerhalb des Members beschrieben an und kopieren Sie es in SYS1.PROCLIB. Geben Sie die folgenden Informationen wie im folgenden Codebeispiel an:

- Die Zeitzonendifferenz (standardmäßig EST5DST)
- Den für die externe Kommunikation (Client-Host-Kommunikation) verwendeten Port (Standardwert 5335)
- Den für die interne, auf den Host begrenzte Kommunikation verwendeten Port (standardmäßig 5336)
- Die für die Fehlerbehebung bei CICS-Transaktionen im Nur-Lese-Speicher verwendete SVC-Nummer (standardmäßig 251)
- Das übergeordnete Qualifikationsmerkmal der Ladebibliothek (standardmäßig FEK)


```

/*
/* RDz Debug Manager
/*
/*DBGMGR  PROC PRM=,                * PRM=DEBUG TO START TRACING
/*          LEPRM='RPTOPTS(ON)',
/*          TZ='EST5EDT',
/*          CLIENT=5335,
/*          HOST=5336,
/*          SVC=251,
/*          HLQ=FEK
/*
/*DBGMGR  EXEC PGM=AQEZPCM,REGION=0M,TIME=NOLIMIT,
/*          PARM=('&LEPRM ENVAR("TZ=&TZ")/&HOST &CLIENT &SVC &PRM')
/*STEPLIB DD DISP=SHR,DSN=&HLQ..SFEKAUTH
/*SYSPRINT DD SYSOUT=*
/*SYSOUT DD SYSOUT=*
/*          PEND
/*
/*

```

Abbildung 2. DBGMGR: Gestartete Task des Debug-Managers

Anmerkung:

- Diese gestartete Task ist optional. Sie wird von der Funktion 'Integrated Debugger' von Developer for System z verwendet. Weitere Informationen finden Sie in „Integrated Debugger (optional)“ auf Seite 133.
- Die Beispiel-JCL hat ursprünglich den Namen FEK.SFEKSAMP(AQESTC) und wird in „Anpassungskonfiguration“ auf Seite 17 in FEK.#CUST.PROCLIB(DBGMGR) umbenannt.
- Informationen zu den empfohlenen Workload Manager-Zielen (WLM) für diese Task finden Sie in "Hinweise zu WLM" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

RSED, gestartete Task für den RSE-Dämon

Passen Sie das Beispielmember FEK.#CUST.PROCLIB(RSED) der gestarteten Task wie innerhalb des Members beschrieben an und kopieren Sie es in SYS1.PROCLIB. Geben Sie die folgenden Informationen wie im folgenden Codebeispiel an:

- Das Ausgangsverzeichnis, in dem Developer for System z installiert ist (standardmäßig /usr/lpp/rdz)
- Die Position der Konfigurationsdateien (standardmäßig /etc/rdz)

```

/*
/* RSE-DÄMON
/*
/*RSED    PROC IVP=,                * 'IVP' to do an IVP test
/*          PORT=,
/*          CNFG='/etc/rdz',
/*          HOME='/usr/lpp/rdz'
/*
/*RSED    EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
/*          PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG -P&PORT'
/*STDOUT  DD SYSOUT=*
/*STDERR  DD SYSOUT=*
/*          PEND
/*
/*

```

Abbildung 3. RSED: Gestartete Task für den RSE-Dämon

Anmerkung:

- Weitere Informationen zu den Startparametern finden Sie in Kapitel 11, „Bedienerbefehle“, auf Seite 193.

- Die Beispiel-JCL hat ursprünglich den Namen FEK.SFEKSAMP(FEKRSED) und wird wie in „Anpassungskonfiguration“ auf Seite 17 beschrieben in FEK.#CUST.PROCLIB(RSED) umbenannt.
- Begrenzen Sie den Jobnamen auf maximal 7 Zeichen. Wenn ein 8 Zeichen langer Name verwendet wird, schlagen die Bedienerbefehle **modify** und **stop** mit der Nachricht "IEE342I MODIFY REJECTED-TASK BUSY" fehl. Dieses Verhalten ist durch das z/OS UNIX-Design für untergeordnete Prozesse festgelegt.
- Informationen zu den empfohlenen Workload Manager-Zielen für diese Task und die von ihr erstellten untergeordneten Prozesse finden Sie in "Hinweise zu WLM" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02). Die untergeordneten Prozesse haben denselben Namen wie die übergeordnete Task (RSED), dem eine einstellige Zufallszahl angehängt ist, beispielsweise RSED8.

Einschränkungen der JCL für die PARM-Variable

Die Länge der PARM-Variablen liegt bei maximal 100 Zeichen. Dies kann zu Problemen führen, wenn Sie benutzerdefinierte Verzeichnisnamen verwenden. Zur Umgehung dieses Problems können Sie eine der folgenden Optionen verwenden:

- Verwenden Sie Standardwerte.

Das Startscript `rsed.sh` kann ohne Argumente gestartet werden. In diesem Fall werden die Standardargumentwerte verwendet.

- Verwenden Sie symbolische Links.

Für lange Verzeichnisnamen können symbolische Links als Kurzform verwendet werden. Der folgende z/OS UNIX-Beispielbefehl definiert einen symbolischen Link (`/usr/lpp/rdz`) zu einem anderen Verzeichnis (`/long/directory/name/usr/lpp/rdz`).

```
ln -s /long/directory/name/usr/lpp/rdz /usr/lpp/rdz
```

- Verwenden Sie STDIN.

Wenn das PARM-Feld leer ist, startet **BPXBATCH** eine z/OS UNIX-Shell und führt das von STDIN bereitgestellte Shell-Script aus. STDIN muss eine als ORDONLY zugeordnete z/OS UNIX-Datei sein. Durch die Verwendung von STDIN wird die Verwendung von PROC-Variablen wie TMPDIR inaktiviert. Die Shell führt die Shell-Anmeldescripts `/etc/profile` und `$HOME/.profile` aus.

Aktualisieren Sie zur Verwendung dieser Methode zunächst die Start-JCL so, dass sie in etwa wie im folgenden Beispiel aussieht:

```
//*
/* RSE-DÄMON - VERWENDUNG VON STDIN
/*
//RSED      PROC CNFG='/etc/rdz'
/*
//RSE       EXEC PGM=BPXBATCH,REGION=0M,TIME=NOLIMIT
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//STDIN     DD PATHOPTS=(ORDONLY),PATH='&CNFG./rsed.stdin.sh'
//          PEND
/*
```

Abbildung 4. RSED: Alternativer Start des RSE-Dämons

Erstellen Sie dann das Shell-Script (in diesem Beispiel `/etc/rdz/rsed.stdin.sh`), das den RSE-Dämon startet. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. Der Inhalt dieses Scripts sieht etwa wie das folgende Beispiel aus:

```

CNFG=/etc/rdz
PORT=
IVP=
/long/directory/name/usr/lpp/rdz/bin/rsed.sh $IVP -C$CNFG -P$PORT -T$TMPDIR

```

Abbildung 5. *rsed.stdin.sh: Alternativer Start des RSE-Dämons*

Anmerkung: Bei Verwendung dieser Methode ist der RSE-Dämon selbst nicht im RSED-Adressraum, sondern in einem RSEDx-Adressraum aktiv. Dies liegt daran, dass z/OS UNIX untergeordnete Prozesse (wie das Starten einer Shell) in separaten Adressräumen ausführt. Dieses Verhalten wird durch das Hinzufügen einer STDENV-DD mit einer Anweisung `_BPX_SHAREAS=YES` nicht geändert, weil die Anweisung zu spät interpretiert wird. Dieser Nebeneffekt erschwert die Verwendung von Developer for System z-Bedienerbefehlen.

TMPDIR-Verarbeitung

z/OS UNIX benötigt Schreibzugriff auf das Verzeichnis `/tmp` oder ein anderes Verzeichnis, auf das durch die Variable `TMPDIR` verwiesen wird, um bestimmte Befehle während des Starts von gestarteten Tasks verarbeiten zu können. Developer for System z legt `TMPDIR` während des Starts von gestarteten Tasks nach der folgenden Logik fest.

Beim Start von gestarteten Tasks überprüft Developer for System z, ob `TMPDIR` bereits festgelegt ist (DD `STDENV`). Ist dies der Fall, verwendet die gestartete Task den festgelegten Wert. Wenn `TMPDIR` nicht festgelegt ist, testet die gestartete Task, ob sie das Verzeichnis `/tmp` verwenden kann. Falls nicht, testet die gestartete Task, ob sie das Ausgangsverzeichnis verwenden kann, das der Benutzer-ID der gestarteten Task zugeordnet ist. Falls auch dieses Verzeichnis nicht verwendet werden kann, schlägt der Start fehl.

Wenn Sie das Ausgangsverzeichnis, das als Standardausweichverzeichnis für `/tmp` vorgesehen ist, nicht verwenden können, müssen Sie `TMPDIR` vorab mit der DD `STDENV` wie im folgenden Beispiel definieren:

```

/*
/* RSE-DÄMON
/*
//RSED      PROC IVP=,                * 'IVP' zum Ausführen eines IVP-Tests
//          PORT=,
//          CNFG='/etc/rdz',
//          HOME='/usr/lpp/rdz'
/*
//RSED      EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
// PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG -P&PORT'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//STDENV    DD PATHOPTS=(ORDONLY),PATH='&CNFG./rsed.stdenv'
//          PEND
/*

```

Abbildung 6. *RSED: Alternative TMPDIR-Verarbeitung*

Erstellen Sie anschließend die Datei (`/etc/rdz/rsed.stdenv` in diesem Beispiel), in der die Definition von `TMPDIR` enthalten sein soll. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. Der Inhalt dieser Datei sieht wie im folgenden Beispiel gezeigt aus:

TMPDIR=/tmp

Abbildung 7. *rsed.stdev*: Alternative TMPDIR-Verarbeitung

Beachten Sie, dass die Datei *rsed.envvars* zwar eine Variable *TMPDIR* enthält, die verwendet wird, sobald die gestartete Task die Datei *rsed.envvars* interpretieren kann, die Datei *rsed.envvars* jedoch *nicht* mit der DD *STDENV* verknüpft werden darf, da dies zu einem Startfehler führt.

ELAXF*-Prozeduren für ferne Build-Erstellung

Developer for System z stellt Beispiel-JCL-Prozeduren bereit, die für die JCL-Generierung, die Fernerstellung von Projektbuilds und die ferne Syntaxprüfung von CICS-BMS-Masken, IMS-MFS-Anzeigen sowie von COBOL-, PL/I-, Assembler- und C/C++-Programmen verwendet werden können. Diese Prozeduren ermöglichen Installationen, eigene Standards anzuwenden. Außerdem wird damit sichergestellt, dass die Entwickler dieselben Prozeduren mit denselben Compileroptionen und Compilerversionen verwenden.

Die Beispielprozeduren und ihre Funktionen sind in Tabelle 8 aufgelistet.

Tabelle 8. *ELAXF*-Beispielprozeduren*

Member	Zweck
ELAXFADT	Beispielprozedur für die Assemblierung und das Debugging von High-Level-Assembler-Programmen
ELAXFASM	Beispielprozedur für die Assemblierung von High-Level-Assembler-Programmen
ELAXFBMS	Beispielprozedur für die Erstellung eines CICS-BMS-Objekts und des entsprechenden Copy-, Dsect- oder Include-Members
ELAXFCOC	Beispielprozedur für COBOL-Kompilierung, integrierte CICS-Umsetzung und integrierte DB2-Umsetzung
ELAXFCOP	Beispielprozedur für die DB2-Vorverarbeitung von „EXEC SQL“-Anweisungen, die in COBOL-Programmen eingebettet sind
ELAXFCOT	Beispielprozedur für die CICS-Umsetzung von „EXEC CICS“-Anweisungen, die in COBOL-Programme eingebettet sind
ELAXFCPC	Beispielprozedur für C-Kompilierungen
ELAXFCPP	Beispielprozedur für C++-Kompilierungen
ELAXFCP1	Beispielprozedur für COBOL-Kompilierungen mit SCM-Vorprozessoranweisungen (-INC und ++INCLUDE)
ELAXFDCL	Beispielprozedur für die Ausführung eines Programms im TSO-Modus
ELAXFGO	Beispielprozedur für den GO-Schritt
ELAXFLNK	Beispielprozedur für die Verknüpfung von C/C++, COBOL-, PLI- und High-Level-Assembler-Programmen
ELAXFMFS	Beispielprozedur für die Erstellung von IMS-MFS-Anzeigen
ELAXFPLP	Beispielprozedur für die DB2-Vorverarbeitung von „EXEC SQL“-Anweisungen, die in PLI-Programme eingebettet sind
ELAXFPLT	Beispielprozedur für die CICS-Umsetzung von „EXEC-CICS“-Anweisungen, die in PLI-Programme eingebettet sind
ELAXFPL1	Beispielprozedur für PL/I-Kompilierung, integrierte CICS-Umsetzung und integrierte DB2-Umsetzung
ELAXFPP1	Beispielprozedur für PL/I-Kompilierungen mit SCM-Vorprozessoranweisungen (-INC und ++INCLUDE)

Tabelle 8. ELAXF*-Beispielprozeduren (Forts.)

Member	Zweck
ELAXFSP	Beispielprozedur für die Registrierung einer gespeicherten Prozedur in DB2
ELAXFSQL	Beispielprozedur für das Aufrufen von SQL
ELAXFTSO	Beispielprozedur für die Ausführung und das Debugging von generiertem DB2-Code im TSO-Modus
ELAXFUOP	Beispielprozedur für die Generierung des UOPT-Schritts beim Erstellen von Programmen, die in CICS- oder IMS-Subsystemen ausgeführt werden

Die Namen der Prozeduren und der einzelnen Prozedurschritte stimmen mit den Standardmerkmalen des Developer for System z-Clients überein. Wenn der Name einer Prozedur oder eines Prozedurschritts geändert wird, muss die entsprechende Eigenschaftendatei auf allen Clients aktualisiert werden. Namen von Prozeduren oder Prozedurschritten sollten nicht geändert werden.

Passen Sie die Member der Beispiel-Build-Prozeduren FEK.#CUST.PROCLIB(ELAXF*) wie in den Membern beschrieben an und kopieren Sie sie in SYS1.PROCLIB. Geben Sie die korrekten übergeordneten Qualifikationsmerkmale für die verschiedenen Produktbibliotheken an, wie in Tabelle 9 beschrieben.

Tabelle 9. Prüfliste der übergeordneten Qualifikationsmerkmale in ELAXF*

Produkt	Standard-HLQ	Wert
Developer for System z	FEK	
CICS	CICSTS42.CICS	
DB2	DSNA10	
IMS	IMS	
COBOL	IGY.V4R2M0	
PL/I	PLI.V4R2M0	
C/C++	CBC	
LE	CEE	
LINKLIB des Systems	SYS1	
MACLIB des Systems	SYS1	

Einige ELAXF*-Prozeduren verweisen auf Dateinamen, die keine festgelegten untergeordneten Qualifikationsmerkmale haben. Ein Beispiel ist die DB2-Ausführungsbibliothek, die DB2-Dienstprogramme enthält, die von Ihrem DB2-Administrator kompiliert wurden. Anhand der Informationen in Tabelle 10 können Sie die Standarddateinamen Namen zuordnen, die an Ihrem Standort verwendet werden.

Tabelle 10. ELAXF*. Prüfliste der vollständig qualifizierten Dateinamen

Produkt	Standard-DSN	Wert
Developer for System z - SQL-Beispiele	FEK.#CUST.SQL	
DB2-Ausführungsbibliothek	DSNA10.RUNLIB.LOAD	

Wenn die ELAXF*-Prozeduren nicht in eine Prozedurenbibliothek des Systems kopiert werden können, fordern Sie die Benutzer von Developer for System z auf, den Jobmerkmalen auf dem Client eine JCLLIB-Karte (direkt nach der JOB-Karte) hinzuzufügen.

```
//MYJOB      JOB <Jobparameters>  
//PROCS JCLLIB ORDER=(FEK.#CUST.PROCLIB)
```

Sicherheitsdefinitionen

Zum Erstellen der Sicherheitsdefinitionen für Developer for System z passen Sie das Beispielmember FEKRACF an und übergeben es. Der Benutzer, der diesen Job übergibt, muss die Zugriffsrechte eines Sicherheitsadministrators haben, z. B. RACF SPECIAL.

FEKRACF ist in FEK.#CUST.JCL enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Anmerkung:

- Bei den Sites, die CA ACF2™ for z/OS verwenden, rufen Sie die Produktseite auf der CA-Unterstützungssite (<https://support.ca.com>) auf und suchen nach dem entsprechenden Dokument (TEC492389) mit Informationen zu Developer for System z. Dieses Dokument enthält Details zu den Sicherheitsbefehlen, die für eine ordnungsgemäße Konfiguration von Developer for System z erforderlich sind.
- Bei den Sites, die CA Top Secret® for z/OS verwenden, rufen Sie die Produktseite auf der CA-Unterstützungssite (<https://support.ca.com>) auf und suchen nach dem entsprechenden Dokument (TEC492091) mit Informationen zu Developer for System z. Das Dokument enthält Details zu den erforderlichen Befehlen für Sicherheitsfunktionen, um Developer for System z richtig zu konfigurieren.

Die folgende Liste von Sicherheitsdefinitionen für Developer for System z ist in Kapitel 9, „Sicherheitsdefinitionen“, auf Seite 157 ausführlich erläutert.

- Sicherheitseinstellungen und -klassen aktivieren
- OMVS-Segment für Benutzer von Developer for System z definieren
- Dateiprofile definieren
- Gestartete Tasks JMON und RSED definieren
- JES-Befehlssicherheit definieren
- RSE als sicheren z/OS UNIX-Server definieren
- Programmgesteuerte MVS-Bibliotheken für RSE definieren
- Anwendungssicherheit für RSE definieren
- PassTicket-Unterstützung für RSE definieren
- Programmgesteuerte z/OS UNIX-Dateien für RSE definieren

Anmerkung: Der Beispieljob FEKRACF enthält nicht nur RACF-Befehle. Während des letzten Schritts für die Sicherheitsdefinitionen wird aus einer z/OS UNIX-Datei eine programmgesteuerte Datei gemacht. Diese Aufgabe könnte - je nach Geschäftspolitik an Ihrem Standort - von einem Systemprogrammierer und nicht vom Sicherheitsadministrator übernommen werden.

Achtung: Die Clientverbindungsanforderung schlägt fehl, wenn PassTickets nicht richtig konfiguriert sind.

FEJJCNFG - Konfigurationsdatei für JES Job Monitor

JMON (JES Job Monitor) stellt alle Services mit Bezug zum JES bereit. Das Verhalten von JES Job Monitor kann mit den Definitionen in FEJJCNFG gesteuert werden.

FEJJCNFG befindet sich in FEK.#CUST.PARMLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Passen Sie das Beispielkonfigurationsmember FEJJCNFG von JES Job Monitor wie im folgenden Beispiel an. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare in derselben Zeile sind nicht zulässig.

Anmerkung: Damit die Änderungen wirksam werden, muss die gestartete JMON-Task erneut gestartet werden.

```
SERV_PORT=6715
TZ=EST5EDT
#APPLID=FEKAPPL
#AUTHMETHOD=SAF
#CODEPAGE=UTF-8
#CONCHAR=$
#CONSOLE_NAME=JMON
#GEN_CONSOLE_NAME=OFF
#HOST_CODEPAGE=IBM-1047
#LIMIT_COMMANDS=NOLIMIT
#LIMIT_CONSOLE=LIMITED
#LIMIT_VIEW=USERID
#LISTEN_QUEUE_LENGTH=5
#LOOPBACK_ONLY=ON
#MAX_DATASETS=32
#MAX_THREADS=200
#TIMEOUT=3600
#TIMEOUT_INTERVAL=1200
#TRACE_STORAGE=OFF
#SEARCHALL=OFF
#SUBMIT_TIMEOUT=30
#SUBMITMETHOD=TSO
#TSO_TEMPLATE=FEK.#CUST.CNTL(FEJTSO)
```

Abbildung 8. FEJJCNFG - Konfigurationsdatei für JES Job Monitor

SERV_PORT

Die Portnummer für JES Job Monitor. Der Standardport ist 6715. Bei Bedarf kann der Port geändert werden.

Anmerkung:

- Dieser Wert muss der Portnummer entsprechen, die in der Konfigurationsdatei `rsed.envvars` für JES Job Monitor festgelegt wurde. Wenn unterschiedliche Werte verwendet werden, ist keine Verbindung zwischen Client und JES Job Monitor möglich. Informationen zur Definition der Variablen von RSE finden Sie in „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.
- Überprüfen Sie vor Auswahl eines Ports, ob der Port auf Ihrem System verfügbar ist. Verwenden Sie dazu die TSO-Befehle **NETSTAT** und **NETSTAT PORTL**.

TZ Zeitzonenselektor. Die Standardeinstellung ist EST5EDT. Die Standardzeitzo-

ne ist UTC + 5 Stunden (Eastern Standard Time mit Sommerzeit). Setzen Sie diesen Wert auf Ihre Zeitzone. Weitere Informationen hierzu finden Sie in der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802).

Folgende Definitionen sind optional. Wenn Sie diese Definitionen übergehen, werden die angegebenen Standardwerte verwendet:

APPLID

Gibt die Anwendungs-ID an, die für Ihre Sicherheitssoftware zur Erkennung von JES Job Monitor verwendet wird. Die Standardeinstellung ist FEKAPPL. Entfernen Sie das Kommentarzeichen und setzen Sie den Wert auf die erforderliche Anwendungs-ID.

Anmerkung: Dieser Wert muss der Anwendungs-ID entsprechen, die in der Konfigurationsdatei `rsed.envvars` für RSE festgelegt wurde. Wenn unterschiedliche Werte verwendet werden, ist keine Verbindung zwischen Client und JES Job Monitor möglich. Informationen zur Definition der Variablen von RSE finden Sie in „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.

AUTHMETHOD

Die Standardeinstellung ist SAF und bedeutet, dass die SAF-Sicherheitschnittstelle (System Authorization Facility) verwendet wird. Ändern Sie diese Einstellung nur auf Anweisung des IBM Support Center.

CODEPAGE

Die Codepage der Workstation. Die Standardeinstellung ist UTF-8. Die Workstation-Codepage ist auf UTF-8 gesetzt und sollte generell nicht geändert werden. Falls Sie Schwierigkeiten mit Zeichen von Nationalsprachen haben, z. B. mit dem Währungssymbol, müssen Sie möglicherweise das Kommentarzeichen vor der Anweisung entfernen und UTF-8 in die Codepage der Workstation ändern.

CONCHAR

Gibt das Befehlszeichen für die JES-Konsole an. Standardmäßig ist CONCHAR für JES2 auf `CONCHAR=$` und für JES3 auf `CONCHAR=*` gesetzt. Entfernen Sie das Kommentarzeichen und geben Sie das angeforderte Befehlszeichen an.

CONSOLE_NAME

Gibt den Namen der EMCS-Konsole an, über die Befehle für Jobs (Hold, Release, Cancel und Purge) abgesetzt werden. Der Standardname lautet JMON. Entfernen Sie das Kommentarzeichen und setzen Sie den Wert unter Beachtung der folgenden Richtlinien auf den erforderlichen Konsolennamen.

- Der `CONSOLE_NAME` muss entweder aus 2 bis 8 alphanumerischen Zeichen bestehen oder mit `'&SYSUID'` (ohne Anführungszeichen) angegeben sein.
- Wenn ein Konsolennamen angegeben ist, wird eine einzelne Konsole mit diesem Namen für alle Benutzer verwendet. Falls die Konsole mit diesem Namen bereits verwendet wird, scheitert der vom Client ausgegebene Befehl.
- Wenn `&SYSUID` angegeben ist, wird als Konsolename die Client-Benutzer-ID verwendet. Somit wird für jeden Benutzer eine andere Konsole verwendet. Falls die Konsole mit diesem Namen bereits verwendet wird (z. B. der Benutzer verwendet SDSF ULOG), scheitert der vom Client ausgegebene Befehl möglicherweise je nach Einstellung von `GEN_CONSOLE_NAME`.

Unabhängig vom verwendeten Konsolennamen wird die Benutzer-ID des Clients, der den Befehl anfordert, als logische Einheit der Konsole verwendet und in den syslog-Nachrichten IEA630I und IEA631I aufgezeichnet.

```
IEA630I OPERATOR console NOW ACTIVE,    SYSTEM=sysid, LU=id
IEA631I OPERATOR console NOW INACTIVE,  SYSTEM=sysid, LU=id
```

GEN_CONSOLE_NAME

Aktiviert bzw. inaktiviert das automatische Generieren von alternativen Konsolennamen. Die Standardeinstellung ist OFF. Zur Aktivierung alternativer Konsolennamen entfernen Sie das Kommentarzeichen und ändern den Wert in ON.

Diese Anweisung wird nur verwendet, wenn `CONSOLE_NAME` gleich `&SYSUID` ist und die Benutzer-ID nicht als Konsolename verfügbar ist.

Wenn `GEN_CONSOLE_NAME=ON` ist, wird ein alternativer Konsolename generiert, indem der Benutzer-ID eine einzelne Ziffer hinzugefügt wird. Dafür werden die Ziffern von 0 bis 9 versucht. Wenn keine verfügbare Konsole gefunden wird, scheitert der vom Client abgesetzte Befehl.

Wenn `GEN_CONSOLE_NAME=OFF` ist, scheitert der vom Client abgesetzte Befehl.

Anmerkung: Nur die Einstellungen ON und OFF sind gültig.

HOST_CODEPAGE

Die Codepage des Hostsystems. Die Standardeinstellung ist IBM-1047. Entfernen Sie das Kommentarzeichen und ändern Sie den Wert in die Codepage Ihres Hostsystems.

Beachten Sie, dass diese Codepage nicht für die Dateninterpretation, sondern nur für Serveroperationen und die Konfiguration von Clientverbindungen verwendet wird. Der Developer for System z-Client stellt die Codepage bereit, die für die Dateninterpretation zu verwenden ist (und die aus den Eigenschaften des Subsystems "MVS-Dateien" abgerufen wird).

LIMIT_COMMANDS

Definiert, für welche Jobs der Benutzer ausgewählte JES-Befehle absetzen kann (Show JCL, Hold, Release, Cancel und Purge). Die Standardeinstellung (`LIMIT_COMMANDS=USERID`) schränkt die Befehle auf die Jobs ein, deren Eigner der Benutzer ist. Wenn der Benutzer berechtigt sein soll, Befehle für alle Spooldateien abzusetzen (sofern dies von Ihrem Sicherheitsprodukt unterstützt wird), entfernen Sie das Kommentarzeichen vor dieser Anweisung und geben `LIMITED` oder `NOLIMIT` an.

Tabelle 11. Matrix der Befehlsberechtigungen für `LIMIT_COMMANDS`

LIMIT_COMMANDS	Jobeigner	
	Benutzer	Anderer Eigner
USERID (Standard)	Zulässig	Nicht zulässig
LIMITED	Zulässig	Zulässig, wenn die Berechtigung explizit in den Sicherheitsprofilen erteilt wird
NOLIMIT	Zulässig	Zulässig, wenn die Sicherheitsprofile die Berechtigung enthalten oder die JESSPOOL-Klasse nicht aktiv ist

Anmerkung: Nur die Einstellungen USERID, LIMITED und NOLIMIT sind gültig.

LIMIT_CONSOLE

Definiert den Berechtigungsumfang, der der zum Ausführen von unterstützten JES-Befehlen (Hold, Release, Cancel und Purge) verwendeten Konsole erteilt wird. Der Standardwert (LIMIT_CONSOLE=LIMITED) beschränkt die Berechtigung auf Befehle, die durch ein Sicherheitsprofil in der Klasse OPERCMDS geschützt sind. Wenn die Ausführung von unterstützten JES-Befehlen, die nicht durch ein Sicherheitsprofil geschützt sind, zugelassen werden soll, entfernen Sie das Kommentarzeichen von dieser Anweisung und geben den Wert NOLIMIT an.

Wenn ein Sicherheitsprofil für einen Befehl vorhanden ist, muss der Benutzer über ausreichende Berechtigungen für die Ausführung des Befehls verfügen, unabhängig von der Einstellung für LIMIT_CONSOLE. Nur die Einstellungen LIMITED und NOLIMIT sind gültig.

LIMIT_VIEW

Diese Einstellung definiert, welche Ausgaben der Benutzer anzeigen kann. Wenn die Standardeinstellung (LIMIT_VIEW=NOLIMIT) verwendet wird, kann der Benutzer alle JES-Ausgaben anzeigen, sofern Ihr Sicherheitsprodukt dies zulässt. Zum Einschränken der Anzeige auf Ausgaben, deren Eigner der Benutzer ist, entfernen Sie das Kommentarzeichen vor dieser Anweisung und geben USERID an.

Anmerkung: Nur die Einstellungen USERID und NOLIMIT sind gültig.

LISTEN_QUEUE_LENGTH

Die Länge der TCP/IP-Warteschlange für eingehende Verbindungen. Die Standardeinstellung ist 5. Ändern Sie diese Einstellung nur auf Anweisung des IBM Support Center.

LOOPBACK_ONLY

Definiert, ob JES Job Monitor nur an die Loopback-Adresse oder an jeden verfügbaren TCP/IP-Stack gebunden wird. Die Bindung an die Loopback-Adresse ist sicherer, weil nur Tasks, die für dieses z/OS-System lokal sind, auf JES Job Monitor zugreifen können. Die Standardeinstellung ist ON. Entfernen Sie das Kommentarzeichen von dieser Anweisung und geben Sie OFF an, wenn JES Job Monitor an alle TCP/IP-Stacks gebunden werden soll.

MAX_DATASETS

Maximale Anzahl von Spool-Ausgabedateigruppen, die JES Job Monitor an den Client zurückgibt (z. B. SYSOUT, SYSPRINT, SYS00001 usw.). Die Standardeinstellung ist 32. Der Maximalwert ist 2147483647.

MAX_THREADS

Dies ist die maximale Anzahl Benutzer, die JES Job Monitor gleichzeitig benutzen können. Die Standardeinstellung ist 200. Der Maximalwert ist 2147483647. Wenn Sie diese Anzahl erhöhen, müssen Sie unter Umständen auch den Adressbereich von JES Job Monitor vergrößern.

TIMEOUT

Dieser Parameter gibt die Zeitspanne (in Sekunden) an, nach der ein Thread bei fehlender Interaktion mit dem Client (mit kill) beendet wird. Die Standardeinstellung ist 3600 (1 Stunde). Der Maximalwert ist 2147483647. Mit TIMEOUT=0 wird die Funktion inaktiviert.

TIMEOUT_INTERVAL

Die Zeit zwischen den Überprüfungen auf eine Zeitlimitüberschreitung in Sekunden. Die Standardeinstellung ist 1200. Der Maximalwert ist 2147483647.

TRACE_STORAGE

Aktiviert den Speichertrace. Die Standardeinstellung ist OFF. Nur die Werte ON und OFF sind gültig. Verwenden Sie diese Option nur auf Anweisung des IBM Support Center. Wenn nach jedem Befehl ein Speicherbericht an DD SYSOUT geschrieben werden soll, entfernen Sie die Kommentarzeichen von dieser Anweisung und geben ON an.

SEARCHALL

Erfasst die APPC- und z/OS UNIX-Ausgabe, die dem JES Job Monitor-Filter entspricht, beispielsweise die Ausgabe, die von einem CARMA-Server von Developer for System z, der mithilfe der Methode CRASTART gestartet wurde, in SYSOUT geschrieben wurde. Die Standardeinstellung ist OFF. Nur die Werte ON und OFF sind gültig. Zur Erfassung der zusätzlichen Spooldateien entfernen Sie das Kommentarzeichen von dieser Anweisung und geben ON an.

SUBMIT_TIMEOUT

Die Anzahl von Sekunden, die Developer for System z auf den Abschluss des Jobs TSO_TEMPLATE wartet. Die Standardeinstellung ist 30. Der Maximalwert ist 2147483647. Hinweis: SUBMIT_TIMEOUT hat nur eine Wirkung, wenn auch SUBMITMETHOD=TSO angegeben ist.

SUBMITMETHOD=TSO

Jobübergabe mit Hilfe von TSO. Bei Verwendung der Standardeinstellung (SUBMITMETHOD=JES) werden Jobs direkt an JES übergeben. Zur Übergabe des Jobs durch den TSO-Befehl **SUBMIT** entfernen Sie das Kommentarzeichen von dieser Anweisung und geben TSO an. Bei dieser Methode können TSO-Exits aufgerufen werden. Diese Methode bringt jedoch einen Leistungsnachteil mit sich.

Anmerkung:

- Nur die Einstellungen TSO und JES sind gültig.
- Wenn SUBMITMETHOD=TSO angegeben ist, muss auch TSO_TEMPLATE definiert sein.

TSO_TEMPLATE

Wrapper-JCL für die Übergabe von Jobs mithilfe von TSO. Der Standardwert ist FEK.#CUST.CNTL(FEJTSO). Diese Anweisung bezieht sich auf den vollständig qualifizierten Namen der JCL, die als Wrapper für den TSO-Befehl **SUBMIT** verwendet werden soll. Weitere Informationen finden Sie unter der Anweisung SUBMITMETHOD.

Anmerkung:

- Ein Beispiel-Wrapper-Job ist in FEK.#CUST.CNTL(FEJTSO) enthalten. Dieses Member stellt weitere Informationen zur erforderlichen Anpassung bereit.
- TSO_TEMPLATE hat nur eine Wirkung, wenn auch SUBMITMETHOD=TSO angegeben ist.

RSE-Konfigurationsdatei `rsed.envvars`

Die RSE-Serverprozesse (RSE-Dämon, RSE-Thread-Pool und RSE-Server) verwenden die Definitionen in `rsed.envvars`. Developer for System z und Services anderer Anbieter können in dieser Konfigurationsdatei optional auch Umgebungsvariablen zur eigenen Verwendung definieren.

RSE (Remote Systems Explorer) stellt Kernservices wie den Verbindungsaufbau vom Client zum Hostsystem und das Starten anderer Server für bestimmte Services bereit.

Die Datei `rsed.envvars` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Die folgende Beispieldatei `rsed.envvars` muss an Ihre Systemumgebung angepasst werden. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit dem Nummernzeichen (`#`). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert enthalten. Kommentare sind in derselben Zeile nicht zulässig. Zeilenfortsetzungen und Leerzeichen vor und nach dem Gleichheitszeichen (`=`) werden nicht unterstützt.

Anmerkung: Damit die Änderungen wirksam werden, muss die gestartete RSED-Task erneut gestartet werden.

```

#####
# (1) erforderliche Definitionen
JAVA_HOME=/usr/lpp/java/J6.0
RSE_HOME=/usr/lpp/rdz
_RSE_RSED_PORT=4035
_RSE_JMON_PORT=6715
RSE_HLQ=FEK
_RSE_HOST_CODEPAGE=IBM-1047
TZ=EST5EDT LANG=C
PATH=/bin:/usr/sbin
_CEE_DMPTARG=/tmp
STEPLIB=NONE
#STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
_RSE_JAVAOPTS=""
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Xms128m -Xmx512m"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Duser.log=/var/rdz/logs"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY="
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlog.retention.period=5"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.clients=30"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.threads=520"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dminimum.threadpool.process=1"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.threadpool.process=100"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dipv6=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Ddisplay.users=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dkeep.all.logs=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dkeep.last.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.standard.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.port.of.entry=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.certificate.mapping=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.automount=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.audit.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.cycle=30"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.retention.period=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.log.mode=RW.R."
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.action=<user_exit>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.action.id=<userid>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlogon.action=<user_exit>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlogon.action.id=<userid>"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Ddeny.nonzero.port=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsingle.logon=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dprocess.cleanup.interval=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dreject.logon.threshold=1000000"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dinclude.c=/etc/rdz/include.conf"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dinclude.cpp=/etc/rdz/include.conf"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DRIS_BUFFER=8"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DAPPLID=FEKAPPL"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DRSE_DSICALL=TSO"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.hits=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.datasets=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.lines=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsearch.server.limit.timeout=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_TEXT_SEARCH=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DHIDE_ZOS_UNIX=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_REMOTE_INDEX_SEARCH=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDISABLE_DELETE_IN_SUBPROJECT=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=3600000"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_SSL_ALGORITHM=TLSv1.2"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_TCP_NO_DELAY=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"

```

Abbildung 9. rsed.envvars: RSE-Konfigurationsdatei

```

#=====
# (2) erforderliche Definitionen für TSO/ISPF Client Gateway
CGI_ISPHOME=/usr/lpp/ispf
CGI_ISPCONF=/etc/rdz
CGI_ISPWORK=/var/rdz
#STEPLIB=$STEPLIB:ISP.SISPLD:ISP.SISPLPA:SYS1.LINKLIB
_RSE_ISPF_OPTS=""
# _RSE_ISPF_OPTS="$_RSE_ISPF_OPTS&ISPPROF=&SYSUID..ISPPROF"
#CGI_ISPPREF=&SYSPREF..ISPF.VCMISPF"
#=====
# (3) erforderliche Definitionen für SCLM Developer Toolkit
_SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:$RSE_HLQ.SFEKAUTH:$RSE_HLQ.SFEKLOAD
# _SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1
#=====
# (4) optionale Definitionen
# _RSE_PORTRANGE=8108-8118
# _BPXK_SETIBMOPT_TRANSPORT=TCPIP
#TMPDIR=/tmp
# _RSE_FEK_SAF_CLASS=FACILITY
# _RSE_LDAP_SERVER=ldap_server_url
# _RSE_LDAP_PORT=389
# _RSE_LDAP_PTC_GROUP_SUFFIX="o=PTC,c=DeveloperForZ"
#GSK_CRL_SECURITY_LEVEL=HIGH
#GSK_LDAP_SERVER=ldap_server_url
#GSK_LDAP_PORT=ldap_server_port
#GSK_LDAP_USER=ldap_userid
#GSK_LDAP_PASSWORD=ldap_server_password
#STEPLIB=$RSE_HLQ.SFEKLOAD:SYS1.CSSLIB:SYS1.SIXMLOD1
#RSE_UBLD_DD=$CGI_ISPCONF/ISPF.conf
#RSE_UBLD_STEPLIB=$STEPLIB
#=====
# (5) nur auf Anweisung des IBM Support Center ändern
_RSE_SAF_CLASS=/usr/include/java_classes/IRRRacf.jar
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_BPX_SHAREAS=YES
_BPX_SPAWN_SCRIPT=YES
_EDC_ADD_ERRNO2=1
JAVA_PROPAGATE=NO
RSE_DSN_SFEKLOAD=$RSE_HLQ.SFEKLOAD
RSE_LIB=$RSE_HOME/lib
PATH=.:$JAVA_HOME/bin:$RSE_HOME/bin:$CGI_ISPHOME/bin:$PATH
LIBPATH=$JAVA_HOME/bin:$JAVA_HOME/bin/classic:$RSE_LIB:$RSE_LIB/icuc
LIBPATH=./usr/lib:$LIBPATH
CLASSPATH=$RSE_LIB:$RSE_LIB/dstore_core.jar:$RSE_LIB/clientserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_extra_server.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/zosserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_miners.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/universalminers.jar:$RSE_LIB/mvsminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/carma.jar:$RSE_LIB/luceneminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsluceneminer.jar:$RSE_LIB/cdzminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvscdzminer.jar:$RSE_LIB/jesminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsutil.jar:$RSE_LIB/jesutils.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/lucene-core-2.3.2.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/cdtparser.jar:$RSE_LIB/wdzBidi.jar
CLASSPATH=$CLASSPATH:$_RSE_SAF_CLASS
CLASSPATH=.:$CLASSPATH
_RSE_PTC=$_RSE_LDAP_PTC_GROUP_SUFFIX
_RSE_ISPF_OPTS="&SESSION=SPAWN$_RSE_ISPF_OPTS"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dldap.server.address=$_RSE_LDAP_SERVER"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dldap.server.port=$_RSE_LDAP_PORT"

```

Abbildung 10. rsed.envvars: RSE-Konfigurationsdatei (Fortsetzung)

```

_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dldap.ptc.group.name.suffix=$_RSE_PTC"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DISPF_OPTS='$_RSE_ISPF_OPTS'"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DA_PLUGIN_PATH=$RSE_LIB"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Xbootclasspath/p:$RSE_LIB/bidiTools.jar"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dfile.encoding=$_RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dconsole.encoding=$_RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_INITIAL_SIZE=0"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_MAX_FREE=0"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_SPIRIT_ON=false"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DSPIRIT_EXPIRY_TIME=90"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DSPIRIT_INTERVAL_TIME=6"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dcom.ibm.cacheLocalHost=true"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Duser.home=$HOME"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dclient.username=$RSE_USER_ID"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlow.heap.usage.ratio=15"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.heap.usage.ratio=40"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_KEEPA_LIVE_ENABLED=true"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_KEEPA_LIVE_RESPONSE_TIMEOUT=60000"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_IO_SOCKET_READ_TIMEOUT=180000"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DRSECOMM_LOGFILE_MAX=0"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Djob.monitor.port=$_RSE_JMON_PORT"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dlock.info.timeout=10000"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -showversion"
_RSE_SERVER_CLASS=org.eclipse.dstore.core.server.Server
_RSE_DAEMON_CLASS=com.ibm.etools.zos.server.RseDaemon
_RSE_POOL_SERVER_CLASS=com.ibm.etools.zos.server.ThreadPoolProcess
_RSE_SERVER_TIMEOUT=120000
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CGI_ISPHOME
_CGI_DTWORK=$_SCLMDT_WORK_HOME
_CMDSERV_BASE_HOME=$CGI_ISPHOME
_CMDSERV_CONF_HOME=$CGI_ISPCONF
_CMDSERV_WORK_HOME=$CGI_ISPWORK
# =====
# (6) zusätzliche Umgebungsvariablen

```

Abbildung 11. *rsed.envvars: RSE-Konfigurationsdatei (Fortsetzung)*

Anmerkung: Symbolische Link sind für die Angabe von Werten und Verzeichnissen in *rsed.envvars* zulässig, solange die Symbole in *rsed.envvars* definiert sind.

Folgende Definitionen sind erforderlich:

JAVA_HOME

Java-Home-Verzeichnis. Die Standardeinstellung ist `/usr/lpp/java/J6.0`. Passen Sie das Verzeichnis an Ihre Java-Installation an.

RSE_HOME

RSE-Ausgangsverzeichnis. Die Standardeinstellung ist `/usr/lpp/rdz`. Passen Sie das Verzeichnis an Ihre Installation von Developer for System z an.

_RSE_RSED_PORT

Portnummer des RSE-Dämons. Die Standardeinstellung ist 4035. Bei Bedarf können Sie diesen Wert ändern.

Anmerkung:

- Überprüfen Sie mit den TSO-Befehlen `NETSTAT` und `NETSTAT PORTL` vor der Auswahl eines Ports, ob der Port auf Ihrem System verfügbar ist.
- Dieser Port wird für die Client-Host-Kommunikation verwendet.
- Die gestartete RSED-Task kann die hier angegebene Portnummer außer Kraft setzen.

_RSE_JMON_PORT

Portnummer für JES Job Monitor. Die Standardeinstellung ist 6715. Bei Bedarf können Sie diesen Wert ändern.

Anmerkung:

- Dieser Wert muss der Portnummer entsprechen, die in der Konfigurationsdatei FEJJCNFG für JES Job Monitor festgelegt wurde. Wenn unterschiedliche Werte verwendet werden, ist keine Verbindung zwischen Client und JES Job Monitor möglich. Informationen zur Definition der Variablen von JES Job Monitor finden Sie in „FEJJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34.
- Überprüfen Sie mit den TSO-Befehlen **NETSTAT** und **NETSTAT PORTL** vor der Auswahl eines Ports, ob der Port auf Ihrem System verfügbar ist.
- Die gesamte Kommunikation über diesen Port ist auf Ihr z/OS-Hostsystem beschränkt.

RSE_HLQ

Für die Installation von Developer for System z verwendetes übergeordnetes Qualifikationsmerkmal. Die Standardeinstellung ist FEK. Passen Sie den Wert an die Position Ihrer Developer for System z-Dateien (Datasets) an.

_RSE_HOST_CODEPAGE

Die Codepage des Hostsystems. Die Standardeinstellung ist IBM-1047. Ändern Sie den Wert in die Codepage Ihres Hostsystems. Beachten Sie, dass diese Codepage nicht für die Dateninterpretation, sondern nur für Serveroperationen und die Konfiguration von Clientverbindungen verwendet wird. Der Developer for System z-Client stellt die Codepage bereit, die für die Dateninterpretation zu verwenden ist (und die aus den Eigenschaften des Subsystems "MVS-Dateien" abgerufen wird).

TZ Zeitzonenselektor. Die Standardeinstellung ist EST5EDT. Die Standardzeitzone ist UTC + 5 Stunden (Eastern Standard Time mit Sommerzeit). Passen Sie diesen Wert an Ihre Zeitzone an.

Weitere Informationen hierzu finden Sie in der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802).

LANG

Gibt den Namen der Standardländereinstellung an. Der Standardwert ist C. C gibt die POSIX-Ländereinstellung an, wobei Ja_JP beispielsweise für die japanische Ländereinstellung steht. Passen Sie den Wert an Ihre Ländereinstellung an.

PATH Befehlspfad. Die Standardeinstellung ist /bin:/usr/sbin:.. Bei Bedarf können Sie diesen Wert ändern.

_CEE_DMPTARG

Von der Java Virtual Machine (JVM) verwendete z/OS UNIX-Position für den LE-Speicherauszug (Language Environment). Die Standardposition ist /tmp.

STEPLIB

Zugriff auf MVS-Dateigruppen, die nicht in LINKLIST/LPALIB enthalten sind. Die Standardeinstellung ist NONE.

Sie können das Kommentarzeichen vor einer oder mehreren der folgenden STEPLIB-Anweisungen entfernen und die Anweisungen anpassen, wenn Sie die Notwendigkeit, erforderliche Bibliotheken in LINKLIST/LPALIB zu

haben, umgehen möchten. Weitere Informationen zur Verwendung der Bibliotheken in der folgenden Liste finden Sie in „PARMLIB-Änderungen“ auf Seite 19:

```
# RSE
STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
# ISPF
STEPLIB=$STEPLIB:ISP.SISPLOAD:ISP.SISPLPA:SYS1.LINKLIB
# SCLM Developer Toolkit
STEPLIB=$STEPLIB:$RSE_HLQ.SFEKAUTH:$RSE_HLQ.SFEKLOAD
# zUnit, xUnit support for Enterprise COBOL and PL/I
STEPLIB=$STEPLIB:$RSE_HLQ.SFEKLOAD:SYS1.CSSLIB:SYS1.SIXMLD1
```

Anmerkung:

- Die Verwendung von STEPLIB unter z/OS UNIX wirkt sich negativ auf die Leistung aus.
- Wenn eine STEPLIB-Bibliothek eine APF-Berechtigung hat, ist diese Berechtigung auch für alle anderen STEPLIB-Bibliotheken erforderlich. Bibliotheken verlieren ihre APF-Berechtigung, wenn sie mit STEPLIB-Bibliotheken ohne APF-Berechtigung gemischt werden.
- Bibliotheken, die in den Link-Pack-Bereich (LPA) gestellt werden müssen, erfordern unter Umständen zusätzliche Programmsteuerberechtigungen und APF-Berechtigungen, wenn für den Zugriff auf diese Bibliotheken LINKLIST oder STEPLIB verwendet wird.
- Die angeforderte STEPLIB-Verknüpfung kann nicht durch das Codieren einer DD-Anweisung STEPLIB in der Server-JCL gesetzt werden.

_RSE_JAVAOPTS

Zusätzliche RSE-spezifische Java-Optionen. Weitere Informationen zu dieser Definition finden Sie in „Zusätzliche Java-Startparameter mit _RSE_JAVAOPTS definieren“ auf Seite 51.

Die folgenden Definitionen sind erforderlich, wenn für TSO Commands Service oder SCLM Developer Toolkit das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

CGI_ISPHOME

Ausgangsverzeichnis für den ISPF-Code, der den TSO/ISPF-Client-Gateway-Service bereitstellt. Die Standardeinstellung ist /usr/lpp/ispf. Passen Sie das Verzeichnis an Ihre ISPF-Installation an. Diese Anweisung ist nur erforderlich, wenn das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

CGI_ISPCONF

Basiskonfigurationsverzeichnis für ISPF. Die Standardeinstellung ist /etc/rdz. Passen Sie das Verzeichnis an die Position der Anpassungsdatei ISPF.conf für das TSO/ISPF Client Gateway an. Diese Anweisung ist nur erforderlich, wenn das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

CGI_ISPWORK

Basisarbeitsverzeichnis für ISPF. Die Standardeinstellung ist /var/rdz. Passen Sie die Position an das vom TSO/ISPF Client Gateway verwendete Verzeichnis WORKAREA an. Diese Anweisung ist nur erforderlich, wenn das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

Anmerkung:

- Das TSO/ISPF-Client-Gateway fügt dem in CGI_ISPWORK angegebenen Pfad den Wert /WORKAREA hinzu. Fügen Sie die Angabe nicht selbst hinzu.
- Wenn Sie zur Erstellung der anpassbaren Umgebung nicht den Beispieljob SFEKSAMP(FEKSETUP) verwendet haben, überprüfen Sie, ob das Verzeichnis WORKAREA in dem Pfad vorhanden ist, der in CGI_ISPWORK angegeben wurde. Die Berechtigungsbits des Verzeichnisses müssen auf '777' gesetzt sein.

STEPLIB

STEPLIB wurde bereits im Abschnitt mit den erforderlichen Definitionen beschrieben.

_RSE_ISPF_OPTS

Zusätzliche, für das TSO/ISPF Client Gateway spezifische Java-Optionen. Die Standardeinstellung ist "". Weitere Informationen zu dieser Definition finden Sie in „Zusätzliche Java-Startparameter mit _RSE_ISPF_OPTS definieren“ auf Seite 59. Diese Anweisung ist nur erforderlich, wenn das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

CGI_ISPPREF

Übergeordnetes Qualifikationsmerkmal für die temporäre Datei, die vom TSO/ISPF-Client-Gateway erstellt wurde. Die Standardeinstellung ist "&SYSPREF..ISPF.VCMISPF". Entfernen Sie die Kommentarzeichen und ändern Sie den Wert so, dass er Ihren Namenskonventionen für Dateinamen entspricht. Diese Anweisung ist nur erforderlich, wenn das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

Im Dateinamen können die folgenden Variablen verwendet werden:

- &SYSUID. als Ersatz für die Benutzer-ID des Entwicklers
- &SYSPREF. als Ersatz für das TSO-Präfix des Entwicklers oder der Benutzer-ID, falls das TSO-Präfix nicht ermittelt werden kann
- &SYSNAME. als Ersatz für den Systemnamen, wie im Parmlib-Member IEASYMxx angegeben

Anmerkung: Für diese Anweisung ist ISPF APAR OA38740 erforderlich.

Wenn das SCLM Developer Toolkit verwendet wird, sind die folgenden Definitionen erforderlich.

_SCLMDT_CONF_HOME

Basiskonfigurationsverzeichnis des SCLM Developer Toolkit. Die Standardeinstellung ist /var/rdz/sclmdt. Passen Sie das Verzeichnis an die Position des Verzeichnisses an, das SCLMDT zum Speichern von SCLM-Projektinformationen verwendet. Diese Anweisung ist nur erforderlich, wenn das SCLMDT verwendet wird.

Anmerkung: SCLMDT fügt dem in SCLMDT_CNF_HOME angegebenen Pfad die Werte /CONFIG und /CONFIG/PROJECT hinzu. Fügen Sie die Angabe nicht selbst hinzu.

STEPLIB

STEPLIB wurde bereits im Abschnitt mit den erforderlichen Definitionen beschrieben.

_SCLMDT_TRANTABLE

Name der VSAM für die Umsetzung langer Namen in Kurznamen. Die

Standardeinstellung ist FEK.#CUST.LSTRANS.FILE. Entfernen Sie das Kommentarzeichen und ändern Sie den Namen in den Namen, der im SCLM-Beispieljob ISP.SISPSAMP (FLM02LST) verwendet wird. Diese Anweisung ist nur erforderlich, wenn die Umsetzung langer Namen in Kurznamen im SCLM Developer Toolkit verwendet wird.

ANT_HOME

Ausgangsverzeichnis für Ihre Ant-Installation. Die Standardeinstellung ist /usr/lpp/apache/Ant/apache-ant-1.7.1. Passen Sie das Verzeichnis an Ihre Ant-Installation an. Diese Anweisung ist nur erforderlich, wenn das SCLM Developer Toolkit mit der Java EE-Build-Unterstützung verwendet wird.

Folgende Definitionen sind optional. Wenn Sie diese Definitionen übergehen, werden Standardwerte verwendet:

_RSE_PORTRANGE

Gibt den Bereich der Ports an, die der RSE-Server für die Kommunikation mit einem Client öffnen kann. Standardmäßig kann jeder Port verwendet werden. Weitere Informationen zu dieser Definition finden Sie in „Verfügbaren Portbereich für den RSE-Server definieren“ auf Seite 49. Diese Anweisung ist optional.

_BPXK_SETIBMOPT_TRANSPORT

Gibt den Namen des zu verwendenden TCP/IP-Stack an. Der Standardname ist TCPIP. Entfernen Sie das Kommentarzeichen und geben Sie den angeforderten TCP/IP-Stacknamen an, wie er in der Anweisung TCPIPJOBNAME der zugehörigen TCPIP.DATA definiert ist. Diese Anweisung ist optional.

Anmerkung:

- Die angeforderte Stackaffinität kann nicht durch das Codieren einer DD-Anweisung SYSTCPD in der Server-JCL gesetzt werden.
- Wenn diese Anweisung nicht aktiv ist, bindet RSE an jeden verfügbaren Stack im System (BIND INADDRANY).

TMPDIR

Gibt den Pfad zum Speichern von temporären Dateien an. Die Standardposition ist /tmp. Entfernen Sie das Kommentarzeichen und ändern Sie den Wert in den angeforderten Pfad. Diese Anweisung ist optional.

_RSE_FEK_SAF_CLASS

Gibt die Sicherheitsklasse an, in der FEK.*-Profile definiert sind. Die Standardeinstellung ist FACILITY. Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

_RSE_LDAP_SERVER

Gibt den Namen des LDAP-Serverhosts an, der von der Push-to-Client-Funktion verwendet wird. Die Standardeinstellung ist der aktuelle z/OS-Hostname. Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

_RSE_LDAP_PORT

Gibt den Namen des LDAP-Serverports an, der von der Push-to-Client-Funktion verwendet wird. Der Standardport ist 389. Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

GSK_LDAP_PTC_GROUP_SUFFIX

Gibt das Suffix "O=<organization>,C=<country>" an, das für die Suche nach den Push-to-Client-Gruppen innerhalb des LDAP-Servers erforderlich ist. Die Standardeinstellung ist "O=PTC,C=DeveloperForZ". Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

GSK_CRL_SECURITY_LEVEL

Gibt die Sicherheitsstufe an, die von SSL beim Herstellen einer Verbindung zu LDAP-Servern während der Zertifikatsüberprüfung verwendet wird, um CRLs auf widerrufen Zertifikate zu überprüfen. Die Standardeinstellung ist MEDIUM. Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional. Folgende Werte sind gültig:

- LOW: Die Zertifikatsüberprüfung scheitert nicht, wenn keine Verbindung mit dem LDAP-Server hergestellt werden kann.
- MEDIUM: Die Zertifikatsüberprüfung erfordert die Erreichbarkeit des LDAP-Servers, jedoch keine Definition einer CRL. Dieser Wert ist der Standardwert.
- HIGH: Die Zertifikatsüberprüfung erfordert die Erreichbarkeit des LDAP-Servers, jedoch keine Definition einer CRL.

Anmerkung: Für diese Anweisung ist z/OS 1.9 oder höher erforderlich.

GSK_LDAP_SERVER

Gibt einen oder mehrere durch Leerzeichen getrennte Hostnamen der LDAP-Server an. Zum Erzwingen der Verwendung des angegebenen LDAP-Servers entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

Der Hostname kann eine TCP/IP-Adresse oder eine URL sein. Jeder Hostname kann eine optionale Portnummer enthalten, die durch einen Doppelpunkt (:) von diesem getrennt ist.

GSK_LDAP_PORT

Gibt den LDAP-Server-Port an. Der Standardport ist 389. Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

GSK_LDAP_USER

Gibt den definierten Namen an, der bei der Verbindung mit dem LDAP-Server verwendet wird. Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

GSK_LDAP_PASSWORD

Gibt das Kennwort an, das bei der Verbindung mit dem LDAP-Server verwendet wird. Zum Erzwingen der Verwendung des angegebenen Werts entfernen Sie das Kommentarzeichen und ändern den Wert. Diese Anweisung ist optional.

RSE_UBLD_DD

Gibt die Datendefinitionsanweisungen an, die beim Generieren von JCL für IBM Rational Team Concert-Benutzerbuilds über einen Developer for System z-Client verwendet werden, die TSO- oder ISPF-Befehle aufrufen. Developer for System z verwendet standardmäßig die Definitionen in der Datei ISPF.conf, die durch die Umgebungsvariable CGI_ISPCONF in rsed.envvars referenziert wird. Entfernen Sie die Kommentarzeichen und ändern Sie den Wert, um die DD-Definitionen in der angegebenen Datei zu

verwenden. Die Syntax der Datei muss den in *ISPF.conf*, der Konfigurationsdatei für das TSO/ISPF-Client-Gateway von ISPF, angegebenen Regeln entsprechen. Diese Anweisung ist optional.

RSE_UBLD_STEPLIB

Gibt die STEPLIB-Anweisung an, die beim Generieren von JCL für IBM Rational Team Concert-Benutzerbuilds über einen Developer for System z-Client verwendet werden, die TSO- oder ISPF-Befehle aufrufen. Developer for System z verwendet standardmäßig die STEPLIB-Definition in der Datei `rsed.envvars`. Entfernen Sie die Kommentarsymbole und ändern Sie den Wert, um die angegebene STEPLIB-Definition zu verwenden. Diese Anweisung ist optional.

Die folgenden Definitionen sind erforderlich und sollten nur auf Anweisung des IBM Support Center geändert werden:

_CEE_RUNOPTS

LE-Laufzeitoptionen (Language Environment). Die Standardeinstellung ist "ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)". Modifizieren Sie diese Einstellung nicht.

_BPX_SHAREAS

Ausführung von Vordergrundprozessen in demselben Adressbereich wie die Shell. Die Standardeinstellung ist YES. Modifizieren Sie diese Einstellung nicht.

_BPX_SPAWN_SCRIPT

Direkte Ausführung von Shell-Skripten von der Funktion `spawn()` aus. Die Standardeinstellung ist YES. Modifizieren Sie diese Einstellung nicht.

_EDC_ADD_ERRNO2

Zeigen den Ursachencode in z/OS UNIX-Fehlernachrichten an. Die Standardeinstellung ist 1 und darf nicht geändert werden.

JAVA_PROPAGATE

Gibt den Sicherheits- und Auslastungskontext während der Threaderstellung weiter (nur Java bis Version 1.4). Die Standardeinstellung ist NO und darf nicht geändert werden.

RSE_DSN_SFEKLOAD

Vollständig qualifizierter Dateiname der Ladebibliothek SFEKLOAD. Der Standardwert ist `$RSE_HLQ.SFEKLOAD`. Modifizieren Sie diese Einstellung nicht.

RSE_LIB

RSE-Bibliothekspfad. Die Standardeinstellung ist `$RSE_HOME/lib`. Modifizieren Sie diese Einstellung nicht.

PATH

Befehlspfad. Die Standardeinstellung ist `.:$JAVA_HOME/bin:$RSE_HOME/bin:$CGI_ISPHOME/bin:$PATH`. Modifizieren Sie diese Einstellung nicht.

LIBPATH

Bibliothekspfad. Die Standardeinstellung ist zu lang, um sie hier wiederzugeben. Modifizieren Sie diese Einstellung nicht.

CLASSPATH

Klassenpfad. Die Standardeinstellung ist zu lang, um sie hier wiederzugeben. Modifizieren Sie diese Einstellung nicht.

_RSE_ISPF_OPTS

Zusätzliche, für TSO Commands Service spezifische Java-Optionen. Die Standardeinstellung ist "&SESSION=SPAWN\$_RSE_ISPF_OPTS". Modifizieren Sie diese Einstellung nicht.

_RSE_JAVAOPTS

Zusätzliche RSE-spezifische Java-Optionen. Die Standardeinstellung ist zu lang, um sie hier wiederzugeben. Modifizieren Sie diese Einstellung nicht.

_RSE_SERVER_CLASS

Java-Klassen für den RSE-Server. Die Standardeinstellung ist `org.eclipse.dstore.core.server.Server`. Modifizieren Sie diese Einstellung nicht.

_RSE_DAEMON_CLASS

Java-Klassen für den RSE-Dämon. Die Standardeinstellung ist `com.ibm.etools.zos.server.RseDaemon`. Modifizieren Sie diese Einstellung nicht.

_RSE_POOL_SERVER_CLASS

Java-Klassen für den RSE-Thread-Pool. Die Standardeinstellung ist `com.ibm.etools.zos.server.ThreadPoolProcess`. Modifizieren Sie diese Einstellung nicht.

_RSE_SERVER_TIMEOUT

Zeitlimit für den RSE-Server (der auf den Client wartet) in Millisekunden. Die Standardeinstellung ist 120000 (2 Minuten). Modifizieren Sie diese Einstellung nicht.

SCLMDT_BASE_HOME

Ausgangsverzeichnis für den Code des SCLM Developer Toolkit. Die Standardeinstellung ist `$RSE_HOME`. Modifizieren Sie diese Einstellung nicht.

SCLMDT_WORK_HOME

Basisarbeitsverzeichnis des SCLM Developer Toolkit. Die Standardeinstellung ist `$CGI_ISPHOME`. Modifizieren Sie diese Einstellung nicht.

CGI_DTWORK

SCLM-Developer-Toolkit-Unterstützung für ältere Clients. Die Standardeinstellung ist `$_SCLMDT_WORK_HOME`. Modifizieren Sie diese Einstellung nicht.

_CMDSERV_BASE_HOME

Serviceunterstützung für das TSO/ISPF-Client-Gateway von ISPF. Die Standardeinstellung ist `$CGI_ISPHOME`. Modifizieren Sie diese Einstellung nicht.

_CMDSERV_CONF_HOME

Serviceunterstützung für das TSO/ISPF-Client-Gateway von ISPF. Die Standardeinstellung ist `$CGI_ISPCONF`. Modifizieren Sie diese Einstellung nicht.

_CMDSERV_WORK_HOME

Serviceunterstützung für das TSO/ISPF-Client-Gateway von ISPF. Die Standardeinstellung ist `$CGI_ISPWORK`. Modifizieren Sie diese Einstellung nicht.

Verfügbaren Portbereich für den RSE-Server definieren

Dieser Schritt gehört zur Anpassung der Datei `rsed.envvars`, die die Ports angibt, über die der RSE-Server mit dem Client kommunizieren kann. Dieser Portbereich steht nicht in Verbindung mit dem Port des RSE-Dämons.

Zum besseren Verständnis der Portverwendung wird der RSE-Verbindungsprozess nachfolgend kurz beschrieben:

1. Der Client stellt über den Hostsystemport 4035 eine Verbindung zum RSE-Dämon her.
2. Der RSE-Dämon erstellt einen RSE-Server-Thread.
3. Der RSE-Server öffnet einen Hostsystemport für die Clientverbindung. Die Auswahl dieses Ports kann durch die Definition `_RSE_PORTRANGE` in `rsed.envvars` konfiguriert werden.
4. Der RSE-Dämon gibt die Portnummer an den Client zurück.
5. Der Client stellt eine Verbindung zum Hostsystemport her.

Anmerkung:

- Dieser Prozess ist für die optionale alternative Verbindungsmethode mit REXEC/SSH ähnlich.
- Weitere Informationen finden Sie in "Wissenswertes zu Developer for System z" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

Wenn Sie den Portbereich für die Kommunikation des Clients mit z/OS angeben möchten, entfernen Sie das Kommentarzeichen aus der folgenden Zeile in `rsed.envvars` und passen die Zeile an:

```
#_RSE_PORTRANGE=8108-8118
```

Anmerkung: Überprüfen Sie vor Auswahl eines Portbereichs, ob der Bereich auf Ihrem System verfügbar ist. Verwenden Sie dazu die Befehle **NETSTAT** und **NETSTAT PORTL**.

Für `PORTRANGE` gilt das folgende Format: `_RSE_PORTRANGE=min-max`. Der Wert für `max` ist nicht einschließlich. Beispiel: Der Ausdruck `_RSE_PORTRANGE=8108-8118` bedeutet, dass die Portnummern von 8108 bis 8117 verwendbar sind. Die vom RSE-Server verwendete Portnummer wird wie folgt ermittelt:

1. Wenn in den Subsystemeigenschaften auf dem Client eine Portnummer ungleich null angegeben ist, wird diese Portnummer verwendet. Wenn der Port nicht verfügbar ist, schlägt der Verbindungsaufbau fehl. Diese Konfiguration wird nicht empfohlen.

Anmerkung: Das Hostsystem kann diesen Verbindungsanforderungstyp zurückweisen, wenn Sie die Anweisung `deny.nonzero.port=true` in `rsed.envvars` angeben. Weitere Informationen zu dieser Anweisung finden Sie in „Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren“ auf Seite 51.

2. Wenn die Portnummer in den Subsystemeigenschaften 0 ist und `_RSE_PORTRANGE` in `rsed.envvars` angegeben ist, wird der von `_RSE_PORTRANGE` angegebene Portbereich verwendet. Falls kein Port aus dem Bereich verfügbar ist, schlägt der Verbindungsaufbau fehl.

Der RSE-Server benötigt den Port nicht exklusiv für die Dauer der Clientverbindung. Es kann sich nur während der Zeitspanne, in der die Serververbindung an den Port und der Verbindungsaufbau des Clients erfolgen, kein anderer RSE-Server an den Port binden. Das bedeutet, dass die meisten Verbindungen den ersten Port im Bereich verwenden und der restliche Bereich als Puffer für den Fall mehrerer simultaner Anmeldungen genutzt wird.

3. Wenn die Portnummer in den Subsystemeigenschaften 0 ist und `_RSE_PORTRANGE` nicht in `rsed.envvars` enthalten ist, wird ein beliebiger verfügbarer Port verwendet.

Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren

Mit den verschiedenen `_RSE_*OPTS`-Anweisungen bietet die Datei `rsed.envvars` die Möglichkeit, zusätzliche Parameter für Java beim Start des RSE-Prozesses anzugeben. Die in `rsed.envvars` enthaltenen Beispieloptionen können durch Entfernen des Kommentarzeichens aktiviert werden.

`_RSE_JAVAOPTS` definiert RSE-spezifische Java-Optionen und Standard-Java-Optionen.

`_RSE_JAVAOPTS=""`

Variableninitialisierung. Modifizieren Sie diese Einstellung nicht.

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx512m"`

Festlegen der anfänglichen Heapspeichergröße (`Xms`) und der maximalen Heapspeichergröße (`Xmx`). Die Standardwerte sind 128M bzw. 512M. Ändern Sie diese, um die erforderlichen Werte der Heapgröße zu erzwingen. Wenn diese Anweisung auf Kommentar gesetzt ist, werden die Java-Standardwerte verwendet. Diese sind 4M und 512M (1M und 64M für Java 5.0).

Anmerkung: Weitere Informationen zum Ermitteln der optimalen Werte für diese Anweisung finden Sie in "Definitionen von wichtigen Ressourcen" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs"`

Das Verzeichnis, das den RSE-Dämon, die Serverprotokollierung und die RSE-Prüfdaten enthält. Die Standardeinstellung ist `/var/rdz/logs`. Ändern Sie diese, um die erforderliche Position zu erzwingen. Wenn diese Anweisung auf Kommentar gesetzt ist, wird das Ausgangsverzeichnis der Benutzer-ID verwendet, die dem RSE-Dämon zugeordnet ist. Das Ausgangsverzeichnis ist im Segment für die OMVS-Sicherheit der Benutzer-ID definiert.

Anmerkung: Wenn diese Anweisung oder das entsprechende Ausgangsverzeichnis keinen absoluten Pfad angibt (wobei der Pfad nicht mit einem Schrägstrich (/) beginnt), ist die aktuelle Position des Protokolls relativ zum Konfigurationsverzeichnis. Dies ist standardmäßig das Verzeichnis `/etc/rdz`.

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Duser.log=/var/rdz/logs"`

Verzeichnis für die benutzerspezifischen Protokolle. Die Standardeinstellung ist `/var/rdz/logs`. Ändern Sie diese, um die erforderliche Position zu erzwingen. Wenn diese Anweisung auf Kommentar gesetzt ist oder es sich bei dem Wert um eine leere Zeichenfolge handelt, wird das Ausgangsverzeichnis der Client-Benutzer-ID verwendet. Das Ausgangsverzeichnis ist im Segment für die OMVS-Sicherheit der Benutzer-ID definiert.

Anmerkung:

- Wenn diese Anweisung oder das entsprechende Ausgangsverzeichnis keinen absoluten Pfad angibt (der Pfad nicht mit einem Schrägstrich (/) beginnt), ist die aktuelle Position des Protokolls relativ zum Konfigurationsverzeichnis. Dies ist standardmäßig das Verzeichnis `/etc/rdz`.
- Der vollständige Pfad zu den Benutzerprotokollen ist `userlog/dstorelog/$LOGNAME/`. Dabei ist `userlog` der Wert der Anweisung `user.log`, `dstorelog` ist der Wert der Anweisung `DSTORE_LOG_DIRECTORY` und `$LOGNAME` ist die Benutzer-ID des Clients in Großbuchstaben.

- Stellen Sie sicher, dass die Berechtigungsbits für userlog/dstorelog so festgelegt sind, dass jeder Client \$LOGNAME erstellen kann.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY="

Dieses Verzeichnis wird an den in der Anweisung user.log angegebenen Pfad angehängt. Zusammen bilden sie den Pfad zu den benutzerspezifischen Protokollen. Die Standardeinstellung ist eine leere Zeichenfolge. Ändern Sie den Wert, damit das angegebene Verzeichnis verwendet wird. Wenn diese Anweisung auf Kommentar gesetzt ist, wird .eclipse/RSE/ verwendet.

Anmerkung:

- Der vollständige Pfad zu den Benutzerprotokollen ist userlog/dstorelog/\$LOGNAME/. Dabei ist userlog der Wert der Anweisung user.log, dstorelog ist der Wert der Anweisung DSTORE_LOG_DIRECTORY und \$LOGNAME ist die Benutzer-ID des Clients in Großbuchstaben.
- Das hier angegebene Verzeichnis ist eine relative Angabe zu dem Verzeichnis, das in user.log angegeben ist. Es darf daher nicht mit einem Schrägstrich (/) beginnen.
- Stellen Sie sicher, dass die Berechtigungsbits für userlog/dstorelog so festgelegt sind, dass jeder Client \$LOGNAME erstellen kann.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dlog.retention.period=5"

Anzahl von Tagen, für die Dämon- und Benutzerprotokolle gespeichert werden. Die Standardeinstellung ist 5. Passen Sie die Anweisung so an, dass die Protokolle nach einer bestimmten Anzahl von Tagen gelöscht werden. Geben Sie 0 an, um keine Begrenzung festzulegen. Der Maximalwert ist 365. Beachten Sie, dass das Dämonprotokoll bei der nächsten Aktion, für die eine Dämonaktivität erforderlich ist, bereinigt wird. Benutzerprotokolle werden bei der nächsten Anmeldung des Benutzers bereinigt.

Die folgenden Anweisungen sind standardmäßig auf Kommentar gesetzt.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.clients=30"

Maximale Anzahl der Clients, die ein Thread-Pool bedienen kann. Die Standardeinstellung ist 30. Zur Begrenzung der Anzahl von Clients pro Thread-Pool entfernen Sie das Kommentarzeichen und passen die Option an. Andere Grenzwerte können möglicherweise verhindern, dass RSE diese Begrenzung erreicht.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.threads=520"

Maximale Anzahl von aktiven Threads in einem Thread-Pool, um neue Clients zuzulassen. Die Standardeinstellung ist 520. Zur Begrenzung der Anzahl von Clients in jedem Thread-Pool entsprechend der Anzahl der verwendeten Threads entfernen Sie das Kommentarzeichen und passen die Option an. Jede Clientverbindung verwendet mehrere Threads (17 oder mehr) und andere Grenzwerte können verhindern, dass RSE diese Begrenzung erreicht.

Anmerkung: Dieser Wert muss kleiner als die Einstellungen MAXTHREADS und MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) sein.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dminimum.threadpool.process=1"

Die minimale Anzahl der aktiven Thread-Pools. Die Standardeinstellung ist 1. Zum Starten mindestens der aufgeführten Anzahl von Thread-Pool-Prozessen entfernen Sie das Kommentarzeichen und passen die Option an. Thread-Pool-Prozesse werden für die Lastverteilung der RSE-Server-Threads verwendet. Weitere neue Prozesse werden bei Bedarf gestartet. Wenn

die neuen Prozesse vorab gestartet werden, lassen sich Verzögerungen bei Verbindungen besser vermeiden. Das System verwendet in Leerlaufzeiten allerdings mehr Ressourcen.

Anmerkung: Wenn die Anweisung `single.logon` aktiv ist, werden mindestens zwei Thread-Pools gestartet, auch wenn `minimum.threadpool.process` auf 1 festgelegt ist. Die Standardeinstellung für `single.logon` in `rsed.envvars` ist aktiv.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.threadpool.process=100"

Maximale Anzahl aktiver Thread-Pools. Die Standardeinstellung ist 100. Zur Begrenzung der Anzahl von Thread-Pool-Prozessen entfernen Sie das Kommentarzeichen und passen die Option an. Thread-Pool-Prozesse werden für die Lastverteilung der RSE-Server-Threads verwendet. Eine Begrenzung ihrer Anzahl bedeutet demzufolge eine Beschränkung der Anzahl aktiver Clientverbindungen.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dipv6=true"

TCP/IP-Version. Die Standardeinstellung ist `false`, das heißt, es wird eine IPv4-Schnittstelle verwendet. Zur Verwendung einer IPv6-Schnittstelle entfernen Sie das Kommentarzeichen und geben den Wert `true` an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddisplay.users=true"

Automatisierte Anzeige von aktiven Benutzern. Die Standardeinstellung ist `false`. Entfernen Sie die Kommentarzeichen und geben Sie `true` an, um bei jeder An- und Abmeldung eines Benutzers eine automatisierte Anzeige von aktiven Benutzern in `rseserver.log` zu aktivieren.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dkeep.all.logs=false"

Verwenden Sie Dateinamen mit einer integrierten Zeitmarke für Dämon- und Benutzerprotokolle. Der Standardwert ist `true`. Dies bedeutet, dass die Protokolle erst durch die Einstellung `log.retention.period` gelöscht werden. Entfernen Sie die Kommentarzeichen und geben Sie `false` an, um festgelegte Protokolldateinamen zu verwenden, die bei jedem Start des Dämons oder bei jeder Anmeldung des Benutzers ersetzt werden.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dkeep.last.log=true"

Aufbewahren einer Kopie der Hostprotokolldateien, die zur vorherigen Sitzung gehören. Die Standardeinstellung ist `false`. Zum Umbenennen der vorherigen Protokolldateien in `*.last` während des Serverstarts und des Clientverbindungsaufbaus entfernen Sie das Kommentarzeichen und geben den Wert `true` an. Beachten Sie, dass die Benutzertracedateien `.dstore*` nicht automatisch entfernt werden, wenn der Client wieder verbunden wird, und auch nicht Teil der Verarbeitung für `keep.last.log` sind. Das Entfernen dieser Dateien ist eine manuelle Aktion. Die Anweisung `keep.all.logs` muss auf `false` gesetzt sein, damit `keep.last.log` wirksam wird.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.standard.log=true"

Die Datenströme `'stdout'` und `'stderr'` des Thread-Pools in eine Protokolldatei schreiben. Die Standardeinstellung ist `false`. Zum Speichern der Datenströme `'stdout'` und `'stderr'` entfernen Sie das Kommentarzeichen und geben den Wert `true` an. Die Protokolldateien befinden sich in dem Verzeichnis, auf das die Anweisung `daemon.log` verweist.

Anmerkung:

- Mit dem Bedienerbefehl **MODIFY RSESTANDARDLOG** können Sie die Aktualisierung der Datenstromprotokolldateien dynamisch stoppen oder starten.

- Wenn die Anweisung `enable.standard.log` aktiv ist, gibt es keine benutzerspezifischen Protokolldateien `stdout.log` und `stderr.log`. Die benutzerspezifischen Daten werden dann in den entsprechenden RSE-Thread-Pool-Datenstrom geschrieben.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.port.of.entry=true"

Option für die Überprüfung des Eingangsports. Die Standardeinstellung ist `false`. Zum Erzwingen der Überprüfung des Eingangsports für Clientverbindungen entfernen Sie das Kommentarzeichen und geben den Wert `true` an. Während der Überprüfung des Eingangsports wird die IP-Adresse des Clients von Ihrer Sicherheitssoftware einer Sicherheitszone für Netzzugriff zugeordnet. Die Clientbenutzer-ID muss berechtigt sein, das Profil zu verwenden, das die Sicherheitszone definiert.

Anmerkung:

- Die Überprüfung des Eingangsports muss auch in Ihrem Sicherheitsprodukt aktiviert sein.
- Durch Aktivieren der Überprüfung des Eingangsports wird das Produkt auch für andere z/OS UNIX-Service wie INETD aktiviert.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.certificate.mapping=false"

Verwenden Ihrer Sicherheitssoftware, um eine Anmeldung mit einem X.509-Zertifikat zu authentifizieren. Die Standardeinstellung ist `true`. Wenn der RSE-Dämon die Authentifizierung vom RSE-Dämon unabhängig von der X.509-Unterstützung Ihrer Sicherheitssoftware ausführen soll, entfernen Sie das Kommentarzeichen und geben den Wert `false` an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.automount=true"

Unterstützung für Ausgangsverzeichnisse, die vom z/OS UNIX-Automount erstellt wurden. Die Standardeinstellung ist `false`. Zum Sicherstellen, dass der z/OS UNIX-Automount die Client-Benutzer-ID als Eigentümer des Verzeichnisses verwendet, entfernen Sie das Kommentarzeichen und geben den Wert `true` an.

Anmerkung: Der z/OS UNIX-Automount verwendet die Benutzer-ID des Prozesses, der den Service beim Erstellen eines Dateisystems aufgerufen hat. Wenn diese Option inaktiviert ist, ist dieser Prozess der RSE-Thread-Pool-Server mit der Benutzer-ID STCRSE. Wenn diese Option aktiviert ist, wird mithilfe der Client-Benutzer-ID ein neuer, temporärer Prozess erstellt, bevor der Service aufgerufen wird.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Denable.audit.log=true"

Prüfprotokolloption. Die Standardeinstellung ist `false`. Wenn Sie die Prüfprotokollierung für Clientaktionen erzwingen möchten, entfernen Sie das Kommentarzeichen und geben den Wert `true` an. Prüfprotokolle werden an die Position für RSE-Dämonprotokolle geschrieben. Diese Position wird von der Option `daemon.log` der Variablen `_RSE_JAVAOPTS` angegeben.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.cycle=30"

Anzahl der in einer Prüfprotokolldatei gespeicherten Tage mit Prüfinformationen. Die Standardeinstellung ist 30. Wenn Sie steuern möchten, wie viele Prüfdaten in eine Prüfprotokolldatei geschrieben werden, entfernen Sie das Kommentarzeichen und passen die Option an. Der Maximalwert ist 365.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.retention.period=0"

Anzahl der geführten Prüfprotokolle. Der Standardwert ist 0 und bedeutet, dass keine Begrenzung angegeben ist. Wenn Protokolle nach einer angege-

benen Anzahl von Tagen gelöscht werden sollen, entfernen Sie das Kommentarzeichen und passen die Option an. Der Maximalwert ist 365.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.log.mode=RW.R."

Zugriffsberechtigungsmaske für Prüfprotokolle. Die Standardeinstellung ist RW.R., das heißt, der Eigentümer hat Lese- und Schreibzugriff. Die Standardgruppe des Eigentümers hat Lesezugriff und alle anderen Personen haben keinen Zugriff. Zum Festlegen der erforderlichen Zugriffsberechtigungen entfernen Sie das Kommentarzeichen und passen die Option an.

UNIX-Standards erfordern, dass Berechtigungen für drei Benutzertypen festgelegt werden können: Eigentümer, Gruppe und Sonstige. Die Felder in der Maske audit.log.mode stimmen mit dieser Reihenfolge überein und werden durch einen Punkt (.) getrennt. Alle Felder können leer sein oder den Wert R, W oder RW haben (R = Lesen und W = Schreiben).

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.action=<benutzerexit>"

Name eines Benutzerexits, der beim Schließen einer Prüfprotokolldatei aufgerufen wird. Es gibt keinen Standardwert, in der Datei /usr/lpp/rdz/samples/process_audit.rex wird jedoch ein Beispiexit bereitgestellt. Zum Aktivieren der Nachverarbeitung von Prüfprotokollen entfernen Sie das Kommentarzeichen und geben den vollständigen Pfadnamen des Benutzerexitprogramms an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Daudit.action.id=<benutzer-id>"

Benutzer-ID, die für die Ausführung des in der Variablen audit.action angegebenen Exits verwendet wird. Der Standardwert ist die Benutzer-ID, die dem RSE-Dämon zugeordnet ist. Zum Verwenden der angegebenen Benutzer-ID zur Ausführung des Exits für die Nachverarbeitung der Prüfung entfernen Sie das Kommentarzeichen und geben eine Benutzer-ID an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dlogon.action=<benutzerexit>"

Name eines Benutzerexits, der beim Anmelden eines Benutzers aufgerufen wird. Es gibt keinen Standardwert, in der Datei /usr/lpp/rdz/samples/process_logon.sh wird jedoch ein Beispiexit bereitgestellt. Zum Aktivieren der Nachverarbeitung einer Anmeldung entfernen Sie das Kommentarzeichen und geben den vollständigen Pfadnamen des Benutzerexitprogramms an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dlogon.action.id=<benutzer-id>"

Benutzer-ID, die für die Ausführung des in der Variablen logon.action angegebenen Exits verwendet wird. Der Standardwert ist die Benutzer-ID, die dem RSE-Dämon zugeordnet ist. Zum Verwenden der angegebenen Benutzer-ID zur Ausführung des Exits für die Nachverarbeitung der Anmeldung entfernen Sie das Kommentarzeichen und geben eine Benutzer-ID an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddeny.nonzero.port=true"

Nicht zulassen, dass der Client die Portnummer für die Kommunikation auswählt. Die Standardeinstellung ist false. Wenn Verbindungen zurückgewiesen werden sollen, bei denen der Client angibt, welcher Hostsystemport vom RSE-Server für die Verbindung verwendet werden soll, entfernen Sie das Kommentarzeichen und geben den Wert true an. Weitere Informationen finden Sie in „Verfügbaren Portbereich für den RSE-Server definieren“ auf Seite 49.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsingle.logon=false"

Nicht zulassen, dass sich ein Benutzer mit derselben Benutzer-ID mehrere Male anmeldet. Die Standardeinstellung ist true. Wenn sich eine Benutzer-ID mehrere Male bei einem RSE-Dämon anmelden können soll, entfernen Sie das Kommentarzeichen und geben Sie false an.

Anmerkung:

- Ein zweiter Anmeldeversuch hat zur Folge, dass die erste Anmeldung durch das Hostsystem abgebrochen wird, wenn diese Anweisung nicht aktiv bzw. nicht auf true gesetzt ist. Diese Abbruchaktion wird von der Konsolennachricht FEK210I begleitet.
- Wenn die Anweisung `single.logon` aktiv ist, werden mindestens zwei Thread-Pools gestartet, auch wenn `minimum.threadpool.process` auf 1 festgelegt ist. Die Standardeinstellung für `minimum.threadpool.process` in `rsd.envvars` ist 1.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dprocess.cleanup.interval=0"

RSE-Thread-Pools mit nicht behebbaren Fehlern automatisch entfernen. Standardmäßig werden fehlerhafte RSE-Thread-Pools nicht automatisch entfernt. Wenn fehlerhafte RSE-Thread-Pool-Server in bestimmten Intervallen, wobei die Intervalleinheit Sekunden ist, automatisch entfernt werden sollen, entfernen Sie das Kommentarzeichen und passen die Option an. Durch die Angabe von 0 wird kein Intervallzeitgeber gestartet, aber fehlerhafte RSE-Thread-Pool-Server werden entfernt, wenn der RSE-Dämon die RSE-Thread-Pools bei der Anmeldung eines neuen Clients oder dem Befehl `DISPLAY PROCESS` prüft.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dreject.logon.threshold=1000000"

Ein Thread-Pool, der eine Datei öffnet, die größer als die angegebene Größe ist, akzeptiert keine neuen Anmeldeanforderungen, bis die Datei geladen ist. Die Standarddateigröße ist 1000000 Byte. Zur Angabe der Dateigröße, ab der der Thread-Pool Anmeldeanforderungen ignorieren soll, wenn eine solche Datei geöffnet wird, entfernen Sie das Kommentarzeichen und passen die Option an. Andere Thread-Pools können weiterhin neue Anmeldeanforderungen akzeptieren.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dinclude.c=/etc/rdz/include.conf"

Dieses Variable verweist auf eine vollständig qualifizierte z/OS UNIX-Datei, die eine Liste von erzwungenen Includes für Content-Assist für C-Code enthält. Ein erzwungenes Include besteht aus einer Datei oder einem Verzeichnis, einem Dataset oder einem Dateimember, für das ein Parsing durchgeführt wird, wenn eine Content-Assist-Operation ausgeführt wird, unabhängig davon, ob diese Datei oder das Member mithilfe einer Vorprozessoranweisung in den Quellcode eingeschlossen wurde. Zur Angabe des Namens der Konfigurationsdatei entfernen Sie das Kommentarzeichen und passen die Option an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dinclude.cpp=/etc/rdz/include.conf"

Dieses Variable verweist auf eine vollständig qualifizierte z/OS UNIX-Datei, die eine Liste von erzwungenen Includes für Content-Assist für C++-Code enthält. Ein erzwungenes Include besteht aus einer Datei oder einem Verzeichnis, einem Dataset oder einem Dateimember, für das ein Parsing durchgeführt wird, wenn eine Content-Assist-Operation ausgeführt wird, unabhängig davon, ob diese Datei oder das Member mithilfe einer Vorprozessoranweisung in den Quellcode eingeschlossen wurde. Zur Angabe des Namens der Konfigurationsdatei entfernen Sie das Kommentarzeichen und passen die Option an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"

Bereinigungsintervall für nicht verwendete C/C++-Headerdateien in Millisekunden. Der Standardwert ist 60000, d. h. 1 Minute. Zum Ändern des Bereinigungsintervalls entfernen Sie das Kommentarzeichen und passen die Option an. Durch die Angabe des Werts 0 wird kein Zwischenspei-

chern von C/C++-Headerdateien vorgenommen, wodurch die Leistung der fernen Inhaltshilfe im Editor verringert wird.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DRIS_BUFFER=8"

Puffergröße in Megabyte, die während der Indexerstellung über Fernzugriff verwendet wird. Die Standardeinstellung ist 8 MB. Zum Ändern der Puffergröße entfernen Sie das Kommentarzeichen und passen die Option an. Gültige Werte sind ganze Zahlen zwischen 1 und 2000 (einschließlich). Durch einen größeren Puffer wird die Indexerstellung beschleunigt, aber auch ein größerer Teil des Java-Heapspeichers des Thread-Pools verwendet. Der Puffer wird automatisch in den Index geschrieben, wenn er vor der Beendigung der Indexerstellung voll ist.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DAPPLID=FEKAPPL"

ID der RSE-Serveranwendung. Die Standardeinstellung ist FEKAPPL. Zum Erzwingen der Verwendung der erforderlichen Anwendungs-ID entfernen Sie das Kommentarzeichen und passen diese Option an.

Anmerkung:

- Die Anwendungs-ID muss für Ihre Sicherheitssoftware definiert sein. Wenn dies nicht der Fall ist, wird die Clientanmeldung verhindert.
- Weitere Informationen zu den Sicherheitsauswirkungen bei Änderungen dieses Werts finden Sie in "PassTickets verwenden" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Dieser Wert muss der Anwendungs-ID entsprechen, die in der Konfigurationsdatei FEJJCNFG für JES Job Monitor festgelegt wurde. Wenn unterschiedliche Werte verwendet werden, ist keine Verbindung zwischen Client und JES Job Monitor möglich. Informationen zur Definition der Variablen von JES Job Monitor finden Sie in „FEJJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DRSE_DSICALL=TSO"

Methode, die zum Erfassen von Dateiinformatoren verwendet wird. Standardmäßig verwendet der Service die gleiche TSO/ISPF-Client-Gateway-Infrastruktur, die auch für TSO-Befehle, die vom Benutzer eingegeben werden, verwendet wird. Zur Verwendung einer alternativen, weniger ressourcenintensiven Methode zum Erfassen von Dateiinformatoren entfernen Sie das Kommentarzeichen und geben TSO an. Für diese alternative Methode müssen zusätzliche Voraussetzungen erfüllt sein:

- FEK.SFEKLOAD(FEKDSI) muss sich in LINKLIST oder STEPLIB befinden, d. h. in rsed.envvars definiert sein.
- Die REXX-Laufzeitbibliothek (REXX.*.SEAGLPA) muss sich in LPA, LINKLIST oder STEPLIB befinden, d. h. in rsed.envvars definiert sein.

Diese Variable wird nicht beachtet, wenn APPC anstelle des TSO/ISPF-Client-Gateways von ISPF zur Unterstützung der TSO-Befehlsservices verwendet wird.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.hits=0"

Begrenzen der Ressourcennutzung nicht indexierter Datei- und Textsuchen. Die Standardeinstellung ist 0 (keine Begrenzung). Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, um eine Suche nach der angegebenen Anzahl gefundener Ergebnisse zu stoppen.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.datasets=0"

Begrenzen der Ressourcennutzung nicht indexierter Datei- und Textsuchen. Die Standardeinstellung ist 0 (keine Begrenzung). Entfernen Sie das Kom-

mentarzeichen und passen Sie diese Anweisung an, um eine Suche nach der angegebenen Anzahl gescannter Dateien (Datasets) zu stoppen.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dsearch.server.limit.lines=0"

Begrenzen der Ressourcennutzung nicht indexierter Datei- und Textsuchen. Die Standardeinstellung ist 0 (keine Begrenzung). Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, um eine Suche nach der angegebenen Anzahl gescannter Zeilen zu stoppen.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDISABLE_TEXT_SEARCH=true"

Inaktivieren nicht indexierter Textsuchen. Die Standardeinstellung ist false. Wenn Benutzer keine Volltextsuche auf dem Host starten können sollen, entfernen Sie das Kommentarzeichen und geben den Wert true an.

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDISABLE_REMOTE_INDEX_SEARCH=true"**

Inaktivieren des Menüelements 'Ferne Indexsuche' auf dem Client. Die Standardeinstellung ist false. Wenn Benutzer keine fernen Indizes für Hostdateien erstellen können sollen, entfernen Sie das Kommentarzeichen und geben den Wert true an. Diese Option funktioniert nur bei Clients ab Version 8.5.1.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"

Option für die Kennwortspeicherung. Die Standardeinstellung ist false. Wenn Benutzer ihr Hostkennwort nicht auf dem Client speichern können sollen, entfernen Sie das Kommentarzeichen und geben den Wert true an. Bereits gespeicherte Kennwörter werden entfernt. Diese Option funktioniert nur bei Clients ab Version 7.1.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DHIDE_ZOS_UNIX=true"

Option zum Ausblenden von z/OS UNIX. Die Standardeinstellung ist false. Wenn Benutzern keine z/OS UNIX-Elemente, d. h. die Verzeichnisstruktur und die Befehlszeile, im Client angezeigt werden sollen, entfernen Sie das Kommentarzeichen und geben den Wert true an. Diese Option funktioniert nur bei Clients ab Version 7.6.

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDISABLE_DELETE_IN_SUBPROJECT=true"**

Option zum Inaktivieren des Menüpunkts "Löschen" im Kontextmenü von z/OS-Unterprojekten. Die Standardeinstellung ist false. Wenn der Menüpunkt "Löschen" im Kontextmenü von z/OS-Unterprojekten für Benutzer inaktiviert werden soll, entfernen Sie das Kommentarzeichen und geben den Wert true an. Diese Option funktioniert nur bei Clients ab Version 8.0.1.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=3600000"

Trennung der Verbindung inaktiver Clients. Standardmäßig wird die Verbindung inaktiver Clients nicht unterbrochen. Wenn Clients, die die angegebene Zahl von Millisekunden inaktiv sind, getrennt werden sollen, entfernen Sie das Kommentarzeichen und passen die Option an. 3600000 Millisekunden entsprechen 1 Stunde.

**#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS
-DDSTORE_SSL_ALGORITHM=TLSv1.2"**

Verwendung von TLS anstelle von SSL für die verschlüsselte Kommunikation mit dem Client. Die Standardeinstellung ist SSL. Zur Verwendung von TLS (Version 1.2) für die Developer for System z-Kommunikation zwischen Clients und Hostsystem entfernen Sie das Kommentarzeichen und geben TLSv1.2 an. Diese Option funktioniert nur mit Java ab Version 7.0 und Cli-

ents ab Version 9.0. Beachten Sie, dass der Client der Version 9.0 standardmäßig auch SSL verwendet. Auf dem Client muss -DDSTORE_SSL_ALGORITHM=TLSv1.2 in eclipse.ini angegeben werden, um die verschlüsselte TLS-Kommunikation zu aktivieren.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_TCP_NO_DELAY=true"

Inaktivieren der Funktion TCP/IP DELAY ACK. Die Standardeinstellung ist false. Zum Stoppen der Funktion DELAY ACK in TCP/IP für die Developer for System z-Client-Host-Kommunikation entfernen Sie das Kommentarzeichen und geben true an.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"

Starten des DSTORE-Trace. Verwenden Sie diese Option nur auf Anweisung des IBM Support Center. Die Ergebnisprotokolldatei .dstoreTrace wird in Unicode (ASCII), nicht in EBCDIC erstellt.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"

Starten des DSTORE-Speicher-Trace. Verwenden Sie diese Option nur auf Anweisung des IBM Support Center. Die Ergebnisprotokolldatei .dstoreMemLogging wird in Unicode (ASCII), nicht in EBCDIC erstellt.

Zusätzliche Java-Startparameter mit **_RSE_ISPF_OPTS** definieren

Mit den verschiedenen **_RSE_*OPTS**-Anweisungen bietet die Datei `rsed.envvars` die Möglichkeit, zusätzliche Parameter für Java beim Start des RSE-Prozesses anzugeben. Die in `rsed.envvars` enthaltenen Beispielloptionen können durch Entfernen des Kommentarzeichens aktiviert werden.

Die **_RSE_ISPF_OPTS**-Anweisungen sind RSE-spezifische Java-Optionen, die standardmäßig nur wirksam sind, wenn das TSO/ISPF-Client-Gateway von ISPF von Developer for System z verwendet wird.

_RSE_ISPF_OPTS=""

Variableninitialisierung. Modifizieren Sie diese Einstellung nicht.

_RSE_ISPF_OPTS="\$_RSE_ISPF_OPTS&ISPPROF=&SYSUID..ISPPROF"

Verwendung eines vorhandenen ISPF-Profiles für die ISPF-Initialisierung. Zur Verwendung des angegebenen ISPF-Profiles entfernen Sie das Kommentarzeichen und ändern den Dateinamen.

Im Dateigruppennamen können die folgenden Variablen verwendet werden:

- **&SYSUID.** als Ersatz für die Benutzer-ID des Entwicklers
- **&SYSPREF.** als Ersatz für das TSO-Präfix des Entwicklers oder der Benutzer-ID, falls das TSO-Präfix nicht ermittelt werden kann
- **&SYSNAME.** als Ersatz für den Systemnamen, wie im Parmlib-Member IEASYMxx angegeben

ISPF.conf - Konfigurationsdatei für TSO/ISPF Client Gateway

Das TSO/ISPF-Client-Gateway von ISPF erstellt anhand der Definitionen in `ISPF.conf` eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen. Developer for System z führt in dieser Umgebung einige MVS-basierte Services aus. Zu diesen Services gehören TSO Commands Service und SCLM Developer Toolkit.

Die Datei `ISPF.conf` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Definitionen müssen in Spalte 1 beginnen. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit einem Stern (*). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare in derselben Zeile sind nicht zulässig. Zeilenfortsetzungen werden nicht unterstützt. Wenn Sie Dateinamen verketteten, fügen Sie die Namen in derselben Zeile hinzu und trennen Sie die einzelnen Namen jeweils durch ein Komma (,).

Geben Sie nicht nur die korrekten Namen für die ISPF-Dateien an, sondern fügen Sie auch den Dateinamen für TSO Commands Service, `FEK.SFEKPROC`, zur Anweisung `SYSPROC` oder `SYSEXEC` hin, wie im folgenden Beispiel gezeigt.

```
* ERFORDERLICH:
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
ispmlib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispplib=ISP.SISPSLIB
ispllib=ISP.SISLOAD

* OPTIONAL:
*allocjob = ISP.SISPSAMP(ISPZISP2)
*ISPF_timeout = 900
```

Abbildung 12. *ISPF.conf: ISPF-Konfigurationsdatei*

Anmerkung:

- Sie können Ihre eigenen DD-Anweisungen und Dateiverkettungen hinzufügen, um die TSO-Umgebung anzupassen und so eine TSO-Anmeldeprozedur zu imitieren. Weitere Informationen finden Sie in "TSO-Umgebung anpassen" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Sie können eine `STEPLIB`-Anweisung nicht definieren. Verwenden Sie stattdessen die `STEPLIB`-Anweisung in der Datei `rsed.envvars`.
- Das TSO/ISPF-Client-Gateway funktioniert möglicherweise nicht ordnungsgemäß, wenn Sie ein Produkt eines Fremdanbieters verwenden, das ISPF-Befehle wie **ISPSTART** abfängt. Informationen dazu, wie dieses für Developer for System z inaktiviert werden kann, entnehmen Sie die der Dokumentation zu diesem Produkt. Wenn das Produkt die Zuordnung einer bestimmten DD-Anweisung zu `DUMMY` erfordert, können Sie dieses Verhalten in `ISPF.conf` simulieren, indem Sie die DD-Anweisung `nullfile` zuordnen.

Beispiel:

```
ISPTRACE=nullfile
```

- Wenn Sie die Anweisung `allocjob` verwenden, müssen Sie darauf achten, dass Sie nicht die DD-Definitionen aufheben, die Sie bereits in `ISPF.conf` festgelegt haben.
- Wenn der Parameter `JWT` im `Parmlib`-Member `SMFPRMxx` auf einen niedrigeren Wert als der Wert `ISPF_timeout` in `ISPF.conf` gesetzt ist, müssen Sie mit einem Systemabbruch 522 für Modul `ISPZTS0` rechnen. Dies wirkt sich nicht auf die Operationen in Developer for System z aus, weil das TSO/ISPF-Client-Gateway bei Bedarf automatisch neu gestartet wird.
- Die Änderungen sind für alle neuen Aufrufe aktiv. Es ist kein Serverneustart erforderlich.

Optionale Komponenten

In den folgenden Abschnitten ist eine Kombination optionaler Anpassungstasks beschrieben. Konfigurieren Sie den erforderlichen Service gemäß den Anweisungen im jeweiligen Abschnitt.

Anpassungen in eigenständigen Komponenten von Developer for System z:

- Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 63
- Kapitel 4, „SCLM Developer Toolkit (optional)“, auf Seite 99
- Kapitel 5, „(Optional) Application Deployment Manager (veraltet)“, auf Seite 107
- Kapitel 6, „Hostbasierte Codeanalyse (optional)“, auf Seite 115

Anpassungen in Konfigurationsdateien von Developer for System z:

- „pushtoclient.properties - hostbasierte Clientsteuerung (optional)“ auf Seite 119
- „ssl.properties - RSE-SSL-Verschlüsselung (optional)“ auf Seite 122
- „rsecomm.properties - RSE-Trace (optional)“ auf Seite 125
- „include.conf“ (optional): Erzwungene Includes für C/C++-Content-Assist“ auf Seite 127

Anpassungen an oder für andere Produkte in Bezug auf Developer for System z:

- „(Optional) z/OS UNIX-Unterprojekte“ auf Seite 128
- „Unterstützung für Include-Vorprozessor (optional)“ auf Seite 129
- „xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)“ auf Seite 130
- „Enterprise Service Tools-Unterstützung (optional)“ auf Seite 131
- „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 131
- „IRZ-Diagnosenachrichten für generierten Code (optional)“ auf Seite 132
- „Unterstützung für Tools zur Problembestimmung (optional)“ auf Seite 136
- „Unterstützung für DB2- und IMS-Debug (optional)“ auf Seite 137
- „Unterstützung für File Manager (optional)“ auf Seite 138
- „Bereinigung von "WORKAREA" und "/tmp" (optional)“ auf Seite 139

Installationsprüfung

Eine ausführliche Beschreibung der verschiedenen Installationsprüfprogramme (IVPs) finden Sie in Kapitel 8, „Installationsprüfung“, auf Seite 141, da einige IVPs für optionale Komponenten gelten.

Sie können die Basisfunktionen mit dem folgenden Szenario testen:

1. Starten Sie die gestartete Task bzw. den Benutzerjob JMON. Die Startinformationen in DD STDOUT sollten mit der folgenden Nachricht enden:

FEJ211I Server ready to accept connections.

Falls der Job mit dem Rückkehrcode 66 endet, ist FEK.SFEKAUTH nicht für APF berechtigt.

2. Starten Sie die gestartete RSED-Task bzw. den Benutzerjob mit dem Parameter IVP=IVP. Bei Verwendung dieses Parameters wird der Server nach Ausführung einiger Installationsprüftests beendet.

Überprüfen Sie die DD "STDOUT" auf Nachrichten, die angeben, dass die folgenden IVPs erfolgreich gestartet wurden:

- Java-Start
- JES Job Monitor-Verbindung
- TCP/IP konfigurieren

3. Starten Sie die gestartete RSED-Task bzw. den Benutzerjob ohne den Parameter "IVP". Der RSE-Dämon gibt nach einem erfolgreichen Start die folgende Konsolennachricht aus:

```
FEK002I RseDaemon started. (port=4035)
```

4. Setzen Sie die folgenden Bedienerbefehle ab und prüfen Sie in der Ergebniskonsolennachricht, ob die Tests erfolgreich ausgeführt wurden:

```
F RSED,APPL=IVP PASSTICKET,userid  
F RSED,APPL=IVP DAEMON,userid  
F RSED,APPL=IVP ISPF,userid
```

Ersetzen Sie userid durch eine gültige TSO-Benutzer-ID.

Kapitel 3. Common Access Repository Manager (optional)

Common Access Repository Manager (CARMA) ist eine Serverplattform für Repository Access Manager (RAM). Ein RAM ist eine Anwendungsprogrammierschnittstelle (API) für einen z/OS-basierten SCM (Software Configuration Manager). Indem die SCM-Funktionalität in einen RAM eingeschlossen wird, ist eine einzige API verfügbar, damit ein Client auf alle unterstützten SCM zugreifen kann.

Developer for System z stellt mehrere vordefinierte RAM und Quellcodemuster bereit, um einen eigenen RAM zu erstellen.

SCMs, die auf Hostsystemen basieren, benötigen Adressräume mit einzelnen Benutzern für den Zugriff auf ihre Services. Dazu muss CARMA einen CARMA-Server für jeden Benutzer starten. Es ist nicht möglich, nur einen Server zu erstellen, der mehrere Benutzer unterstützt.

Voraussetzungen und Prüfliste

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines Sicherheitsadministrators und eines TCP/IP-Administrators:

- TCP/IP-Port-Bereich für interne Kommunikation (optional)
- Sicherheitsregel, die Entwicklern die Aktualisierung von CARMA-VSAM-Dateien erlaubt (optional)
- Sicherheitsregel, die Benutzern die Übergabe von CRA*-Jobs erlaubt (optional)
- LPA-Update (optional)

Für die Verwendung von CARMA an Ihrem Standort müssen Sie die folgenden Tasks ausführen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Wählen Sie eine Startmethode für CARMA aus und wählen Sie aus, welche RAM aktiviert werden sollen. Es sind einige Kombinationen von RAM und Serverstartmethoden als vorkonfigurierte Installation verfügbar. Weitere Details enthält der Abschnitt „Serverstartmethode und aktiven RAM auswählen“ auf Seite 64.
2. Erstellen Sie CARMA-VSAM-Dateien. Weitere Details enthalten die Abschnitte „CARMA-VSAM-Dateien“ auf Seite 83 und „CARMA-RAM (Repository Access Manager)“ auf Seite 85.
3. Führen Sie eine erste Anpassung der RSE-Konfigurationsdateien durch, um eine Schnittstelle mit CARMA herzustellen. Die vollständige Anpassung hängt davon ab, welche Startmethode für CARMA ausgewählt wurde. Weitere Details enthält der Abschnitt „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 75.
4. Führen Sie abhängig von der ausgewählten CARMA-Startmethode und den ausgewählten RAM die erforderliche Anpassung der zugehörigen Konfigurationsdateien durch. Weitere Details enthalten die folgenden Abschnitte:
 - „crastart*.conf - CRASTART-Serverstart“ auf Seite 78
 - „CRASUB* - Serverstart mit Batchübergabe“ auf Seite 82
5. Optional: Passen Sie die CA Endevor® SCM-spezifischen Konfigurationsmember an. Details finden Sie in den Abschnitten „CRACFG, CRASHOW und CRAT-

MAP - Konfigurationsdateien für den CA Endevor® SCM-RAM" auf Seite 87 und „Batch-Aktionen für CA Endevor® SCM-RAM" auf Seite 89.

6. Optional: Aktualisieren Sie die Dateizuordnungs-Exec. Details finden Sie in den Abschnitten „CRANDVRA - Zuordnungs-Exec für den CA Endevor® SCM-RAM" auf Seite 88, „CRAALLOC - Angepasste Zuordnungs-Exec für den RAM" auf Seite 92 und „Angepasste Zuordnungs-Exec (optional)" auf Seite 95.
7. Optional: Benutzerexit für den Start erstellen. Details finden Sie im Abschnitt CARMA-Benutzerexit (optional).
8. Optional: Erstellen Sie CRAXJCL als Ersatz für IRXJCL. Weitere Details enthält der Abschnitt „IRXJCL oder CRAXJCL (optional)" auf Seite 97.

Anmerkung: Die Beispielmembers aus diesem Kapitel sind in FEK.#CUST.* und /etc/rdz enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration" auf Seite 17.

Serverstartmethode und aktiven RAM auswählen

Developer for System z unterstützt mehrere Methoden für den Start eines CARMA-Servers. Außerdem stellt Developer for System z mehrere Repository Access Manager (RAM) bereit, die in zwei Gruppen eingeteilt werden können: Produktions-RAM und Muster-RAM. In dieser Veröffentlichung werden einige mögliche Kombinationen von RAM und Serverstartmethoden beschrieben. Alle beschriebenen Konfigurationsszenarios sind als vorkonfigurierte Installationen verfügbar.

CARMA-Serverstart

Developer for System z unterstützt mehrere Methoden für den Start eines CARMA-Servers. Alle Methoden haben Vor- und Nachteile.

CRASTART

Die Methode "CRASTART" startet den CARMA-Server als Subtask innerhalb von RSE. Diese Methode stellt eine sehr flexible Konfiguration bereit, bei der eine gesonderte Konfigurationsdatei verwendet wird, die für den Start eines CARMA-Servers erforderliche Dateizuordnungen und Programmaufrufe definiert. Mit dieser Methode wird die beste Leistung erreicht. Sie nutzt am wenigsten Ressourcen, erfordert jedoch, dass sich das Modul CRASTART im LPA befindet.

Batchübergabe

Die Methode der Batchübergabe startet den CARMA-Server durch Übergabe eines Jobs. Dies ist die in den bereitgestellten Beispielformatdateien verwendete Standardmethode. Sie hat den Vorteil, dass in der Jobausgabe ohne großen Aufwand auf die CARMA-Protokolle zugegriffen werden kann. Bei dieser Methode kann jeder Entwickler auch eigene Server-JCL verwenden, die er selbst verwaltet. Allerdings wird bei dieser Methode für jeden Entwickler, der einen CARMA-Server startet, ein JES-Initiator verwendet.

TSO/ISPF-Client-Gateway (veraltet)

Bei der Methode "TSO/ISPF-Client-Gateway" wird mit dem TSO/ISPF-Client-Gateway von ISPF eine TSO- oder ISPF-Umgebung erstellt, in der der CARMA-Server gestartet wird. Diese Methode erlaubt flexible Dateizuordnungen durch die Möglichkeiten der Datei ISPF.conf. Sie ist jedoch nicht für den Zugriff auf SCMs geeignet, die mit normalen TSO- oder ISPF-Operationen in Konflikt geraten.

Anmerkung: Die Verbindungsaufbaumethode "TSO/ISPF Client Gateway" wurde als "veraltet" markiert. Obwohl diese Funktion noch unterstützt wird, wird sie nicht mehr erweitert. Die Dokumentation wurde in das White Paper *Using ISPF Client Gateway to provide CARMA services* (IBM Form SC14-7292) verschoben, das in der Bibliothek für Developer for System z unter <http://www-01.ibm.com/software/awdtools/rdz/library/> verfügbar ist.

Produktions-RAM

Produktions-RAM sind vordefinierte RAM mit vollem Funktionsumfang, die für den Zugriff auf einen SCM in einer Produktionsumgebung verwendet werden können.

CA Endevor® SCM-RAM

Die Schnittstelle für CA Endevor® Software Configuration Manager in IBM Rational Developer for System z gibt Developer for System z-Clients direkten Zugriff auf CA Endevor® SCM. Nachfolgend wird Schnittstelle für CA Endevor® SCM in IBM Rational Developer for System z mit CA Endevor® SCM-RAM abgekürzt.

RAM für CA Endevor® SCM-Pakete

RAM für CA Endevor® SCM-Pakete ermöglicht Clients von Developer for System z den direkten Zugriff auf CA Endevor® SCM-Pakete.

Muster-RAM

Es werden Muster-RAM zum Testen der Konfiguration Ihrer CARMA-Umgebung und als Beispiele für die Entwicklung eigener RAM bereitgestellt. Der Quellcode ist enthalten.

Achtung: Verwenden Sie die zur Verfügung gestellten Beispiel-RAM nicht in einer Produktionsumgebung.

PDS-RAM

Der PDS-RAM gibt eine Dateiliste ähnlich zu **MVS-Dateien** -> **Meine Dateien** in der Ansicht 'Ferne Systeme' zurück.

Skeleton-RAM

Der Skeleton-RAM gibt ein funktionales Gerüst zurück, das als Ausgangspunkt für die Entwicklung Ihres eigenen RAM verwendet werden kann.

SCLM-RAM

Der SCLM-RAM gibt einen Basiseintrag in SCLM (Software Configuration Manager von ISPF) zurück. Der SCLM-RAM ist standardmäßig nicht aktiviert.

Kombinationen aus vorkonfiguriertem RAM und Serverstart

Es sind einige Kombinationen von RAM und Serverstartmethoden als vorkonfigurierte Installation verfügbar. Für die aufgelisteten Szenarios sind nur geringe Anpassungen erforderlich, damit sie Ihrer Umgebung entsprechen.

- „CRASTART mit CA Endevor® SCM-RAM“ auf Seite 66
- „CRASTART mit Muster-RAM“ auf Seite 68
- „Batchübergabe mit CA Endevor® SCM-RAM“ auf Seite 70
- „Batchübergabe mit Muster-RAM“ auf Seite 73

Ausführliche Informationen zu den verschiedenen Schritten der Szenarios finden Sie in „CARMA-Konfigurationsdetails“ auf Seite 75.

Es ist möglich, jeder CARMA-Konfiguration sofort oder zu einem zukünftigen Zeitpunkt einen RAM hinzuzufügen. Weitere Informationen zum Hinzufügen eines RAM zu einer vorhandenen Konfiguration finden Sie in „(Optional) Unterstützung mehrerer RAM“ auf Seite 93.

CRASTART mit CA Endevor® SCM-RAM

In diesem Abschnitt ist die Konfiguration von CARMA mit den folgenden Spezifikationen beschrieben:

- Serverstart: Methode CRASTART. Für diese Methode muss sich CRASTART im LPA befinden.
- RAM: CA Endevor® SCM-RAM.

Wenn Sie eines der weiteren Szenarios mit anderen Spezifikationen verwenden möchten, können Sie diesen Anpassungsschritt übergehen.

CARMA-VSAM-Dateien erstellen

Zum Definieren und Füllen der VSAM-Dateien für CARMA passen Sie die folgenden JCL-Jobs an und übergeben sie. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Vorhandene VSAM-Dateien werden ersetzt.

Weitere Details zu diesem Schritt finden Sie in „CARMA-VSAM-Dateien“ auf Seite 83.

- FEK.#CUST.JCL(CRA\$VCAD)
- FEK.#CUST.JCL(CRA\$VCAS)
- FEK.#CUST.JCL(CRA\$VMSG)

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in `/etc/rdz/CRASRV.properties` zum Starten eines CARMA-Servers und für die Verbindung mit diesem Server. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten. Starten Sie die gestartete RSED-Task erneut, damit die Änderungen wirksam werden.

Wenn Sie die Standarddateipositionen verwenden, müssen Sie nur den Wert der Anweisung `clist.dsname` in `*CRASTART` und den Wert von `crastart.configuration.file` in `/etc/rdz/crastart.endevor.conf` ändern. Weitere Details zu den verschiedenen Anweisungen finden Sie in „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 75.

```
clist.dsname=*CRASTART
crastart.configuration.file=crastart.endevor.conf
```

Abbildung 13. CRASRV.properties: CRASTART mit CA Endevor® SCM-RAM

crastart.endevor.conf anpassen

CRASTART verwendet die Definitionen in `/etc/rdz/crastart.endevor.conf`, um eine gültige TSO/ISPF-Umgebung für das Starten eines CARMA-Servers zu erstellen. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten. Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

Anweisungen zur Anpassung finden Sie in der Dokumentation innerhalb der Datei. Weitere Informationen zur CRASTART-Startmethode finden Sie in „crastart*.conf - CRASTART-Serverstart“ auf Seite 78.

Anmerkung: Aufgrund von Seitenbreitenbegrenzungen wurden einige Zeilen im folgenden Beispiel in die nächste Zeile umgebrochen. Alle Zeilen, die mit einer Einrückung beginnen, sollten am Ende der vorherigen Zeile hinzugefügt werden.

```
* DD used by RAM
TYPEMAP = FEK.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEK.#CUST.PARMLIB(CRASHOW)
CRACFG = FEK.#CUST.PARMLIB(CRACFG)
* Kommentarteichen für CRABCFG und CRABSKEL für die Verwendung von Batch-Aktionen
entfernen
*CRABCFG = FEK.#CUST.PARMLIB(CRABCFG)
*CRABSKEL= FEK.#CUST.CNTL
CONLIB = CA.NDVR.CSIQLOAD
-COMMAND=ALLOC FI(JCLOUT) SYSOUT(A) WRITER(INTRDR) RECFM(F) LRECL(80)
  BLKSIZE(80)
-COMMAND=ALLOC FI(EXT1ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT2ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT1DEP) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
C1EXMSG= SYSOUT(H)
C1MSG1 = SYSOUT(H)
MSG3FILE= DUMMY

* DD used by CARMA server (CRASERV)
* pay attention to APF authorizations when using TASKLIB
TASKLIB = FEK.SFEKLOAD,CA.NDVR.CSIQAUTH,CA.NDVR.CSIQAUTU
CRADEF = FEK.#CUST.CRADEF
CRAMSG = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (via NDVRC1)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = FEK.SFEKPROC
* CRANDVRA
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1.
  &CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
  &CRAPRM8. )
```

Abbildung 14. crastart.endevor.conf: CRASTART mit CA Endevor® SCM-RAM

Zusätzliche Anpassung des CA Endeavor® SCM-RAM (optional)

Der CA Endeavor® SCM-RAM hat zusätzliche Komponenten, die bei Bedarf angepasst werden können.

- Der CARMA-Startprozess verfügt über einen optionalen Benutzerexit. Weitere Informationen finden Sie unter CARMA-Benutzerexit (optional).
- Der CA Endeavor® SCM-RAM verwendet mehrere Konfigurationsdateien, FEK.#CUST.PARMLIB(CRACFG), FEK.#CUST.PARMLIB(CRASHOW) und FEK.#CUST.PARMLIB(CRATMAP), die angepasst werden können. Weitere Informationen finden Sie in „CRACFG, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endeavor® SCM-RAM“ auf Seite 87.
- Der CA Endeavor® SCM-RAM verwendet eine Zuordnungs-Exec (FEK.SFEKPROC-(CRANDVRA)), die angepasst werden kann. Weitere Informationen finden Sie in „CRANDVRA - Zuordnungs-Exec für den CA Endeavor® SCM-RAM“ auf Seite 88.
- Der CA Endeavor® SCM-RAM unterstützt die Ausführung von CA Endeavor® SCM-Aktionen im Batch-Modus. Für Batch-Aktionen ist eine Konfigurationsdatei (FEK.#CUST.PARMLIB(CRABCFG)) und eine Skeleton-JCL (FEK.#CUST.CNTL(CRABATCA)) erforderlich, die angepasst werden müssen. Weitere Informationen finden Sie in „Batch-Aktionen für CA Endeavor® SCM-RAM“ auf Seite 89.

CRASTART mit Muster-RAM

In diesem Abschnitt ist die Konfiguration von CARMA mit den folgenden Spezifikationen beschrieben:

- Serverstart: Methode CRASTART. Für diese Methode muss sich CRASTART im LPA befinden.
- RAM: Muster-RAM, die nicht für Produktionszwecke zu verwenden sind.

Wenn Sie eines der weiteren Szenarios mit anderen Spezifikationen verwenden möchten, können Sie diesen Anpassungsschritt auslassen.

CARMA-VSAM-Dateien erstellen

Passen Sie die folgenden JCL-Jobs an und übergeben Sie sie, um die VSAM-Dateien für CARMA zu definieren und zu füllen. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Vorhandene VSAM-Dateien werden ersetzt.

Weitere Details zu diesem Schritt finden Sie in „CARMA-VSAM-Dateien“ auf Seite 83 und „CARMA-RAM (Repository Access Manager)“ auf Seite 85.

CARMA

- FEK.#CUST.JCL(CRA\$VDEF)
- FEK.#CUST.JCL(CRA\$VMSG)
- FEK.#CUST.JCL(CRA\$VSTR)

Muster-RAM

- FEK.#CUST.JCL(CRA#VPDS)

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in /etc/rdz/CRASRV.properties zum Starten eines CARMA-Servers und für die Verbindung mit diesem Server. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten. Damit die Änderungen wirksam werden, muss die gestartete RSED-Task erneut gestartet werden.

Wenn Sie die Standarddateipositionen verwenden, müssen Sie nur den Wert der Anweisung `clist.dsname` in `*CRASTART` ändern. Weitere Details zu den verschiedenen Anweisungen finden Sie in „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 75.

```
clist.dsname=*CRASTART
crastart.configuration.file=crastart.conf
```

Abbildung 15. CRASRV.properties: CRASTART mit Muster-RAM

crastart.conf anpassen

CRASTART verwendet die Definitionen in /etc/rdz/crastart.conf, um eine gültige TSO/ISPF-Umgebung für das Starten eines CARMA-Servers zu erstellen. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten. Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

Anweisungen zur Anpassung finden Sie in der Dokumentation innerhalb der Datei. Weitere Informationen zur CRASTART-Startmethode finden Sie in „crastart*.conf - CRASTART-Serverstart“ auf Seite 78.

```
* DD used by RAM
CRARAM1 = FEK.#CUST.CRARAM1                                * PDS RAM
* DD used by CARMA server (CRASERV)
TASKLIB = FEK.SFEKLOAD
CRADEF   = FEK.#CUST.CRADEF
CRAMSG   = FEK.#CUST.CRAMSG
CRASTRS  = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)
* DD used by ISPF (ISPSTART)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU
* DD used by TSO (IKJEFT01)
SYSPROC = #hlq.SFEKPROC                                      * CRAALLOC
SYSTSIN  = DUMMY
SYSTSPRT= SYSOUT(H)

PROGRAM=IKJEFT01 %CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1.
  &CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
  &CRAPRM8. )
```

Abbildung 16. crastart.conf: CRASTART mit Muster-RAM

Anmerkung: Aufgrund von Seitenbreitenbegrenzungen wurden einige Zeilen im Beispiel in die nächste Zeile umgebrochen. Alle Zeilen, die mit einer Einrückung beginnen, sollten am Ende der vorherigen Zeile hinzugefügt werden.

Zusätzliche benutzerdefinierte Anpassung des RAM (optional)

Die angepassten RAMs verfügen über zusätzliche Komponenten, die gegebenenfalls angepasst werden können.

- Der CARMA-Startprozess verfügt über einen optionalen Benutzerexit. Weitere Informationen finden Sie unter CARMA-Benutzerexit (Optional).
- Der angepasste RAM-Start verwendet eine Zuordnungs-Exec, FEK.SFEKPROC-(CRAALLOC), die angepasst werden kann. Weitere Informationen finden Sie im Abschnitt CRAALLOC - Angepasste Zuordnungs-Exec für den RAM.

Batchübergabe mit CA Endevor® SCM-RAM

In diesem Abschnitt ist die Konfiguration von CARMA mit den folgenden Spezifikationen beschrieben:

- Serverstart: Batchübergabemethode. Diese Methode erfordert JES-Initiatoren.
- RAM: CA Endevor® SCM-RAM.

Wenn Sie eines der weiteren Szenarios mit anderen Spezifikationen verwenden möchten, können Sie diesen Anpassungsschritt übergehen.

CARMA-VSAM-Dateien erstellen

Passen Sie die folgenden JCL an und übergeben Sie sie, um die VSAM-Dateien für CARMA zu definieren und zu füllen. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Vorhandene VSAM-Dateien werden ersetzt.

Weitere Details zu diesem Schritt finden Sie in „CARMA-VSAM-Dateien“ auf Seite 83.

- FEK.#CUST.JCL(CRA\$VCAD)
- FEK.#CUST.JCL(CRA\$VCAS)
- FEK.#CUST.JCL(CRA\$VMSG)

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in /etc/rdz/CRASRV.properties zum Starten eines CARMA-Servers und für die Verbindung mit diesem Server. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten. Damit die Änderungen wirksam werden, muss die gestartete RSED-Task erneut gestartet werden.

Wenn Sie die Standarddateipositionen verwenden, müssen Sie nur den Wert der Anweisung `clist.dsname` in `FEK.#CUST.CNTL(CRASUBCA)` ändern. Weitere Details zu den verschiedenen Anweisungen finden Sie in „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 75.

```
clist.dsname='FEK.#CUST.CNTL(CRASUBCA)'
```

Abbildung 17. CRASRV.properties: Batchübergabe mit CA Endevor® SCM-RAM

CRASUBCA anpassen

Mit der CLIST FEK.#CUST.CNTL(CRASUBCA) und der eingebetteten JCL wird ein CARMA-Server übergeben. Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Weitere Informationen zur Startmethode mit Batchübergabe finden Sie in „CRASUB* - Serverstart mit Batchübergabe“ auf Seite 82.

```

PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
/* DD used by RAM
//TYPEMAP DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRASHOW)
//CRACFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRACFG)
/* Kommentarzeichen für CRABCFG und CRABSKEL für die Verwendung von Batch-Aktionen
entfernen
/*CRABCFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRABCFG)
/*CRABSKEL DD DISP=SHR,DSN=FEK.#CUST.CNTL
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD
//JCLOUT DD SYSOUT=(A,INTRDR),DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
//EXT1ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT2ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT1DEP DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//C1MSGSG DD SYSOUT(H)
//C1EXMSGSG DD SYSOUT(H)
//MSG3FILE DD DUMMY
/*
/* DD used by CARMA server (CRASERV)
/* pay attention to APF authorizations when using STEPLIB
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
//CRADEF DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
/*
/* DD used by ISPF (via NDVRC1)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
/*
/* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=FEK.SFEKPROC * CRANDVRA
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &CRAPRM3 +
&CRAPRM4 &CRAPRM5 &CRAPRM6 &CRAPRM7 &CRAPRM8 )
$$
EXIT CODE(0)

```

Abbildung 18. CRASUBCA: Batchübergabe mit CA Endevor® SCM-RAM

Zusätzliche Anpassung des CA Endevor® SCM-RAM (optional)

Der CA Endevor® SCM-RAM hat zusätzliche Komponenten, die bei Bedarf angepasst werden können.

- Der CA Endeavor® SCM-RAM verwendet mehrere Konfigurationsdateien, FEK.#CUST.PARMLIB(CRACFG), FEK.#CUST.PARMLIB(CRASHOW) und FEK.#CUST.PARMLIB(CRATMAP), die angepasst werden können. Weitere Informationen finden Sie in „CRACFG, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endeavor® SCM-RAM“ auf Seite 87.
- Der CA Endeavor® SCM-RAM verwendet eine Zuordnungs-Exec (FEK.SFEKPROC-(CRANDVRA)), die angepasst werden kann. Weitere Informationen finden Sie in „CRANDVRA - Zuordnungs-Exec für den CA Endeavor® SCM-RAM“ auf Seite 88.
- Der CA Endeavor® SCM-RAM unterstützt die Ausführung von CA Endeavor® SCM-Aktionen im Batch-Modus. Für Batch-Aktionen ist eine Konfigurationsdatei (FEK.#CUST.PARMLIB(CRABCFG)), eine Skeleton-JCL FEK.#CUST.CNTL(CRABATCA) und eine optionale Standard-Jobkarte FEK.#CUST.CNTL(CRABJOBC), erforderlich, die angepasst werden müssen. Weitere Informationen finden Sie in „Batch-Aktionen für CA Endeavor® SCM-RAM“ auf Seite 89.

Batchübergabe mit Muster-RAM

In diesem Abschnitt ist die Konfiguration von CARMA mit den folgenden Spezifikationen beschrieben:

- Serverstart: Batchübergabemethode, für die JES-Initiatoren erforderlich sind
- RAM: Muster-RAM, die nicht für Produktionszwecke zu verwenden sind

Wenn Sie eines der weiteren Szenarios mit anderen Spezifikationen verwenden möchten, können Sie diesen Anpassungsschritt übergehen.

VSAM-Dateien erstellen

Passen Sie die folgenden JCL-Jobs an und übergeben Sie sie, um die VSAM-Dateien für CARMA zu definieren und zu füllen. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Vorhandene VSAM-Dateien werden ersetzt.

Weitere Details zu diesem Schritt finden Sie in „CARMA-VSAM-Dateien“ auf Seite 83 und „CARMA-RAM (Repository Access Manager)“ auf Seite 85.

CARMA

- FEK.#CUST.JCL(CRA\$VDEF)
- FEK.#CUST.JCL(CRA\$VMSG)
- FEK.#CUST.JCL(CRA\$VSTR)

Muster-RAM

- FEK.#CUST.JCL(CRA#VPDS)

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in /etc/rdz/CRASRV.properties zum Starten eines CARMA-Servers und für die Verbindung mit diesem Server. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten. Damit die Änderungen wirksam werden, muss die gestartete RSED-Task erneut gestartet werden.

Weil dies das Standardszenario für Developer for System z ist, sind keine Änderungen erforderlich, wenn Sie mit einer unveränderten Kopie der Datei beginnen. Weitere Details zu den verschiedenen Anweisungen finden Sie in „CRASRV.properties“.

ties: RSE-Schnittstelle zu CARMA" auf Seite 75.

```
clist.dsname='FEK.#CUST.CNTL(CRASUBMT)'
```

Abbildung 19. CRASRV.properties: Batchübergabe mit Muster-RAM

CRASUBMT anpassen

Mit der CLIST FEK.#CUST.CNTL(CRASUBMT) und der eingebetteten JCL wird ein CARMA-Server übergeben. Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Weitere Informationen zur Startmethode mit Batchübergabe finden Sie in „CRASUB* - Serverstart mit Batchübergabe“ auf Seite 82.

```
PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//CRARAM1 DD DISP=SHR,DSN=FEK.#CUST.CRARAM1          * PDS RAM
//*
//* DD used by CARMA server (CRASERV)
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//CRADEF  DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG  DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (ISPSTART)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=#h1q.SFEKPROC          * CRAALLOC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &CRAPRM3 +
                                     &CRAPRM4 &CRAPRM5 &CRAPRM6 &CRAPRM7 &CRAPRM8 )
$$
EXIT CODE(0)
```

Abbildung 20. CRASUBMT: Batchübergabe mit Muster-RAM

Zusätzliche benutzerdefinierte Anpassung des RAM (optional)

Die angepassten RAMs verfügen über zusätzliche Komponenten, die gegebenenfalls angepasst werden können.

- Der CARMA-Startprozess verfügt über einen optionalen Benutzerexit. Weitere Informationen finden Sie unter CARMA-Benutzerexit (Optional).

- Der angepasste RAM-Start verwendet eine Zuordnungs-Exec, FEK.SFEKPROC-(CRAALLOC), die angepasst werden kann. Weitere Informationen finden Sie im Abschnitt CRAALLOC - Angepasste Zuordnungs-Exec für den RAM.

CARMA-Konfigurationsdetails

Die verschiedenen Konfigurationsszenarios, die in dieser Veröffentlichung dokumentiert werden, verwenden viele CARMA-Konfigurationsdateien gemeinsam. Die Details dieser Konfigurationsdateien werden hier dokumentiert. In den verschiedenen Szenarios wird auf diese Dateien verwiesen.

CRASRV.properties: RSE-Schnittstelle zu CARMA

Der CARMA-Server stellt eine Standard-API für andere Produkte, die Hostsysteme verwenden, für den Zugriff auf einen oder mehrere Software Configuration Manager (SCM) bereit. CARMA stellt jedoch keine Methoden für eine direkte Kommunikation mit einem Client-Computer bereit. Für diese Kommunikation greift CARMA auf andere Produkte zurück, wie zum Beispiel den RSE-Server. Der RSE-Server verwendet die Einstellungen in CRASRV.properties zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server.

Die Datei CRASRV.properties befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

Anmerkung: Damit die Änderungen wirksam werden, muss die gestartete RSED-Task erneut gestartet werden.

```
# CRASRV.properties - CARMA-Konfigurationsoptionen
#
port.start=0
#port.range=100
#user.exit='FEK.SFEKSAMP(CRAEXIT)'
startup.script.name=carma.startup.rex
clist.dsname='FEK.#CUST.CNTL(CRASUBMT)'
crastart.configuration.file=crastart.conf
#crastart.stub=/usr/lpp/rdz/bin/CRASTART
#crastart.syslog=Partial
#crastart.timeout=420
#crastart.steplib=FEK.SFEKLPA
#crastart.tasklib=TASKLIB
```

Abbildung 21. CARMA-Konfigurationsdatei CRASRV.properties

port.start

Wenn port.start den Wert 0 (null) hat, verwendet CARMA für die Kommunikation zwischen CARMA und dem RSE-Server einen ephemeren Port. In diesem Szenario ordnet TCP/IP eine freie, zufällig gewählte Portnummer zu. Wenn port.start einen Wert ungleich null hat, wird dieser Wert als Anfangspunkt eines Portbereichs interpretiert, der für die Kommunikation zwischen CARMA und dem RSE-Server verwendet wird. In diesem Fall muss auch die Variable port.range definiert werden. Der Standardport lautet 0. Die Kommunikation über diesen Port ist auf Ihr Hostsystem beschränkt.

Anmerkung: Überprüfen Sie vor Auswahl eines Ports, ob der Port auf Ihrem System verfügbar ist. Verwenden Sie dazu die Befehle **NETSTAT** und

NETSTAT PORTL. Weitere Informationen finden Sie in "Reservierte TCP/IP-Ports" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

#port.range

Gibt den Portbereich an, der mit `port.start` beginnt und für die CARMA-Kommunikation verwendet wird, wenn `port.start` ungleich null ist. Die Standardeinstellung ist 100. Zur Angabe der Größe des Portbereichs entfernen Sie das Kommentarzeichen und passen die Option an. Beispiel: Wenn `port.start` den Wert '5227' und `port.range` den Wert '100' hat, können die Ports von 5227 bis 5326 (beide eingeschlossen) von CARMA verwendet werden. Jede CARMA-Verbindung verwendet einen Port exklusiv, sodass die Angabe eines Portbereichs die maximale Anzahl gleichzeitiger CARMA-Sitzungen eingrenzt.

#user.exit

Definiert benutzerdefinierten Code, der beim CARMA-Start ausgeführt wird. Entfernen Sie die Kommentarzeichen und geben Sie den Dateinamen des auszuführenden Code an.

Ist der Dateiname in Anführungszeichen (') gesetzt, handelt es sich um einen absoluten Verweis. Bei Angabe ohne Anführungszeichen (') wird dem Dateinamen die Clientbenutzer-ID und nicht das TSO-Präfix vorangestellt. In letzterem Fall müssen alle CARMA-Benutzer ihren eigenen Exit-Code führen.

Ein Beispielbenutzerexit ist als SFEKSAMP(CRAEXIT) zur Verfügung gestellt. In diesem Beispiel sind auch die an den Benutzerexit übergebenen Startargumente dokumentiert. Weitere Informationen finden Sie unter CARMA-Benutzerexit (optional).

startup.script.name

Definiert das CARMA-Start-Script. Die Standardeinstellung ist `carma.startup.rex`. Diese REXX-Exec löst den Start eines CARMA-Servers aus. Der Dateiname kann auf mehrere Arten angegeben werden:

- Nullzeichenfolge: Dies bedeutet, dass die Variable nicht angegeben ist. In diesem Fall wird der Standardwert verwendet.
- Nur ein Dateiname: Dies ist die Standardmethode. CARMA durchsucht die Verzeichnisse in der Umgebungsvariable PATH, um die Datei zu finden. Das Verzeichnis, in dem die ausführbaren Developer for System z-Dateien enthalten sind (standardmäßig `/usr/lpp/rdz/bin`), wird automatisch der Umgebungsvariablen PATH hinzugefügt.
- Relativer Pfad: Dies ist das Verzeichnis und der Dateiname ohne einen führenden Schrägstrich (/). CARMA fügt Ihr Konfigurationsverzeichnis (standardmäßig `/etc/rdz/`) dem angegebenen Pfad hinzu, um diesen zu einem absoluten Pfad zu machen.
- Absoluter Pfad: Dies ist das Verzeichnis und der Dateiname mit einem führenden Schrägstrich (/). CARMA verwendet die angegebene Dateiposition.

clist.dsname

Definiert die Startmethode für den CARMA-Server. Weitere Details zu den verschiedenen Startmethoden finden Sie in „Serverstartmethode und aktiven RAM auswählen“ auf Seite 64.

- *CRASTART gibt an, dass der CARMA-Server innerhalb von RSE mit CRA-START als Sub-Task gestartet werden soll. Wenn Sie *CRASTART angeben, müssen Sie auch die Anweisungen `crastart.*` angeben oder deren Standardwerte verwenden.

- *ISPF gibt an, dass der CARMA-Server mit dem TSO/ISPF-Client-Gateway von ISPF gestartet werden soll. Diese Startmethode ist veraltet.
- Alle anderen Werte definieren die Position der CLIST CRASUBMT mit Namenskonventionen wie in TSO. In einfachen Anführungszeichen (') ist der Dateiname ein absoluter Verweis. Bei Angabe ohne Anführungszeichen (') wird dem Dateinamen die Clientbenutzer-ID und nicht das TSO-Präfix vorangestellt. Für diesen letztgenannten Fall müssen alle CARMA-Benutzer eine eigene CLIST CRASUBMT führen.

Die Standardeinstellung ist 'FEK.#CUST.CNTL(CRASUBMT)'. Diese CLIST startet einen CARMA-Server, wenn eine Verbindung mit der Batch-Übergabemethode geöffnet wird.

crastart.configuration.file

Gibt den Namen der CRASTART-Konfigurationsdatei an. Die Standardeinstellung ist `crastart.conf`. Diese Datei gibt die zum Starten eines CARMA-Servers erforderlichen Dateizuordnungen und Programmaufrufe an. Diese Anweisung wird nur verwendet, wenn für die Anweisung `clist.dsname` der Wert *CRASTART angegeben ist. Der Dateiname kann auf mehrere Arten angegeben werden:

- Nullzeichenfolge: Dies bedeutet, dass die Variable nicht angegeben ist. Der Standardwert wird verwendet.
- Nur ein Dateiname: Dies ist die Standardmethode. CARMA durchsucht Ihr Konfigurationsverzeichnis (standardmäßig `/etc/rdz`), um die Datei zu finden.
- Relativer Pfad: Dies ist das Verzeichnis und der Dateiname ohne einen führenden Schrägstrich (/). CARMA fügt Ihr Konfigurationsverzeichnis (standardmäßig `/etc/rdz/`) dem angegebenen Pfad hinzu, um diesen zu einem absoluten Pfad zu machen.
- Absoluter Pfad: Dies ist das Verzeichnis und der Dateiname mit einem führenden Schrägstrich (/). CARMA verwendet die angegebene Dateiposition.

#crastart.stub

z/OS UNIX-Stub zum Aufrufen von CRASTART. Die Standardeinstellung ist CRASTART. Dieser Stub macht das MVS-basierte Lademodul CRASTART für z/OS UNIX-Prozesse verfügbar. Zur Angabe eines bestimmten Pfads entfernen Sie das Kommentarzeichen und passen die Option an. Diese Anweisung wird nur verwendet, wenn für die Anweisung `clist.dsname` der Wert *CRASTART angegeben ist. Der Dateiname kann auf mehrere Arten angegeben werden:

- Nullzeichenfolge: Dies bedeutet, dass die Variable nicht angegeben ist. Der Standardwert wird verwendet.
- Nur ein Dateiname: Dies ist die Standardmethode. CARMA durchsucht die Verzeichnisse in der Umgebungsvariable `PATH`, um die Datei zu finden. Das Verzeichnis, in dem die ausführbaren Developer for System z-Dateien enthalten sind (standardmäßig `/usr/lpp/rdz/bin`), wird automatisch der Umgebungsvariablen `PATH` hinzugefügt.
- Relativer Pfad: Dies ist das Verzeichnis und der Dateiname ohne einen führenden Schrägstrich (/). CARMA fügt Ihr Konfigurationsverzeichnis (standardmäßig `/etc/rdz/`) dem angegebenen Pfad hinzu, um diesen zu einem absoluten Pfad zu machen.
- Absoluter Pfad: Dies ist das Verzeichnis und der Dateiname mit einem führenden Schrägstrich (/). CARMA verwendet die angegebene Dateiposition.

#crastart.syslog

Gibt an, wie viele Informationen beim Starten eines CARMA-Servers mit CRASTART in das Systemprotokoll geschrieben werden. Die Standardeinstellung ist Partial. Gültige Werte sind in der folgenden Tabelle aufgeführt.

A (All)	Alle Traceinformationen werden im SYSLOG ausgegeben.
P (Partial)	Im SYSLOG werden nur Informationen zum Aufbau und zur Trennung von Verbindungen sowie Fehlerinformationen ausgegeben.
Alle anderen Werte	Im SYSLOG werden nur Fehlerbedingungen ausgegeben.

Zur Angabe des erforderlichen Detaillierungsgrads für Ihre Systemprotokollnachrichten entfernen Sie das Kommentarzeichen und passen den Wert an. Diese Anweisung wird nur verwendet, wenn für die Anweisung `clist.dsname` der Wert `*CRASTART` angegeben ist.

#crastart.timeout

Dieser Parameter gibt die Zeitspanne (in Sekunden) an, nach der ein CARMA-Server bei fehlender Aktivität beendet wird. Die Standardeinstellung ist 420 (7 Minuten). Zur Angabe des erforderlichen Zeitlimitwerts entfernen Sie das Kommentarzeichen und passen die Option an. Diese Anweisung wird nur verwendet, wenn für die Anweisung `clist.dsname` der Wert `*CRASTART` angegeben ist.

Anmerkung: Für Modul CRASERV tritt ein Systemabbruch 522 auf, wenn der Wert für den Parameter JWT im PARMLIB-Member SMFPRMxx niedriger als der Wert `crastart.timeout` in CRASRV.properties ist. Dieser Vorfall hat keine Auswirkungen auf CARMA-Operationen, da der Server automatisch erneut gestartet wird, falls dies erforderlich ist.

#crastart.steplib

Die Position des Moduls CRASTART, wenn der Zugriff über die Anweisung STEPLIB in `rsed.envvars` erfolgt. Die Standardeinstellung ist FEK.S-FEKLPA. Wenn das Modul CRASTART keine Komponente von LPA oder LINKLIST sein kann, entfernen Sie das Kommentarzeichen und passen diese Anweisung an. Wenn sich das Modul CRASTART nicht im LPA befindet, können Probleme mit der Programmsteuerung und mit APF auftreten. Diese Anweisung wird nur verwendet, wenn für die Anweisung `clist.dsname` der Wert `*CRASTART` angegeben ist.

#crastart.tasklib

Alternativer Name für den DD-Namen TASKLIB in `crastart.conf`. Die Standardeinstellung ist TASKLIB. Wenn der DD-Name TASKLIB für Ihren SCM oder RAM eine spezielle Bedeutung hat und nicht als Ersatz für STEPLIB verwendet werden kann, entfernen Sie das Kommentarzeichen und passen diese Anweisung an. Diese Anweisung wird nur verwendet, wenn für die Anweisung `clist.dsname` der Wert `*CRASTART` angegeben ist.

crastart*.conf - CRASTART-Serverstart

RSE startet das Lademodul CRASTART, das entsprechend den Definitionen in `crastart*.conf` eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen erstellt. Developer for System z führt in dieser Umgebung den CARMA-Server (CRASERV) aus.

Die Datei `crastart*.conf` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Anmerkung: Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

Developer for System z stellt mehrere Konfigurationsdateien `crastart*.conf` bereit. Jede dieser Musterdateien ist für ein bestimmtes Anpassungsszenario vorkonfiguriert:

- `crastart.endevor.conf` ist für den CRASTART-Start mit dem CA Endevor® SCM-RAM konfiguriert.
- `crastart.conf` ist für den CRASTART-Start mit Muster-RAM konfiguriert.

Die Funktion der Datei `crastart*.conf` ist dem Konzept eines JCL-Jobstroms ähnlich, unterliegt aber größeren Einschränkungen.

- Die folgenden Beispiele zeigen gültige Zeilenformate:
 - `* comment`
 - `ddname=dsn1,dsn2,dsn3` `* comment`
 - `ddname=SYSOUT(c)` `* comment`
 - `ddname=DUMMY` `* comment`
 - `-COMMAND=<beliebiger BPXWDYN-Befehl>` `* comment`
 - `PROGRAM = progname parms` `* comment`

Anmerkung: Der Befehl **BPXWDYN** ist in *Using REXX and z/OS UNIX System Services* (IBM Form SA22-7806) dokumentiert und ermöglicht komplexe Zuordnungsstrukturen.

- Alle Eingaben werden in Großschreibung geändert.
- Zeilenfortsetzungen werden nicht unterstützt.
- Es gibt keine Einschränkung für die Zeilenlänge.
- Vor und nach dem Gleichheitszeichen (=) sind ein oder mehrere Leerzeichen zulässig.
- DD-Zuordnungen müssen der zugehörigen Anweisung `PROGRAM` vorangehen.
- Hier zugeordnete DD-Namen werden am Ende der Programmausführung freigegeben. Sie werden nicht zusammengefasst.
- Von den aufgerufenen Programmen zugeordnete DD-Namen werden nicht freigegeben.
- Mehrere Dateien können zu einem DD-Namen verkettet werden. Die Dateinamen müssen durch ein Komma (,) getrennt werden und die Verkettung wird in der aufgelisteten Reihenfolge durchsucht.
- Alle Dateizuordnungen werden mit `DISP=SHR` definiert, mit Ausnahme von Zuordnungen mithilfe von `-COMMAND`.
- Integrierte Daten werden nicht unterstützt. Alle Daten müssen in Katalogdateien gespeichert sein.
- Variablen können nur rechts vom Gleichheitszeichen (=) verwendet werden.
- Die folgenden Variablen werden unterstützt:

<code>&CRAUSER.</code>	Client-Benutzer-ID
<code>&CRADATE.</code>	Aktuelles Datum im (siebenstelligen julianischen) Format <code>Djjjjttt</code>

&CRATIME.	Aktuelle Uhrzeit im Format Thhmmss (Stunden, Minuten, Sekunden)
&CRAPRM1.	Portnummer
&CRAPRM2.	Zeitlimitwert
Systemsymbol	Ein beliebiges SYS1.PARMLIB(IEASYMxx)-Systemsymbol
-<DD-Name>	Ein Gedankenstrich (-), auf den ein vorher definierter DD-Name folgt, wird als Rückbezug (*.ddname) in JCL verwendet. Die ursprüngliche DD muss mit der Anweisung –COMMAND zugeordnet werden.

Anmerkung: Für das TSO-Präfix gibt es keine Variable, weil TSO während der Interpretation der Konfigurationsdatei nicht aktiv ist. Wenn Sie das TSO-Präfix oder weitere Variablen benötigen, die nicht verfügbar sind, finden Sie weitere Informationen in „Angepasste Zuordnungs-Exec (optional)“ auf Seite 95.

Abb. 22 zeigt ein grundlegendes crastart*.conf-Skeleton, das ISPF-Services einschließt.

```
* DD used by RAM

* DD used by CARMA server (CRASERV)
TASKLIB = FEK.SFEKLOAD
CRADEF  = FEK.#CUST.CRADEF
CRAMSG  = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS
CARMALOG= SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (ISPSTART)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = #hlq.SFEKPROC
SYSTSIN = DUMMY
SYSPRINT= SYSOUT(H)

PROGRAM=IKJEFT01 %CRAALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1.
&CRAPRM2. &CRAPRM3. &CRAPRM4. &CRAPRM5. &CRAPRM6. &CRAPRM7.
&CRAPRM8. )
```

Abbildung 22. crastart*.conf: CARMA-Serverstart mit CRASTART

Anmerkung:

- Aufgrund von Seitenbreitenbegrenzungen wurden einige Zeilen im Beispiel in die nächste Zeile umgebrochen. Alle Zeilen, die mit einer Einrückung beginnen, sollten am Ende der vorherigen Zeile hinzugefügt werden.
- Wenn Sie die Zeile PROGRAM ändern, stellen Sie sicher, dass mindestens ein Leerzeichen vor der schließenden runden Klammer (") der PARM()-Anweisung steht, um die Verarbeitung der Zeichenfolge zu vereinfachen.
- Sie können Ihre eigenen DD-Anweisungen und Dateiverkettungen hinzufügen, um die CARMA-TSO-Umgebung anzupassen und so eine TSO-Anmeldeprozedur zu imitieren.
- Der DD-Name TASKLIB verhält sich wie STEPLIB in JCL. Der entsprechende DD-Name muss mit dem Wert übereinstimmen, der für crastart.tasklib in CRASRV.properties angegeben wurde. Dies wird in „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 75 beschrieben.
- Für TASKLIB-Zuordnungen gelten normale APF-Regeln. Bibliotheken verlieren ihre APF-Berechtigung, wenn eine nicht über APF berechtigte Bibliothek Teil der Verkettung ist.
- Für Modul CRASERV tritt ein Systemabbruch 522 auf, wenn der Wert für den Parameter JWT im PARMLIB-Member SMFPRMxx niedriger als der Wert crastart.timeout in CRASRV.properties ist. Dieser Systemabbruch hat keine Auswirkungen auf CARMA-Operationen, da der Server automatisch erneut gestartet wird, falls dies erforderlich ist.
- Details des CARMA-Serverstarts werden in der Datei rsecomm.log angezeigt, wenn der Server beendet wird. Weitere Informationen zum Festlegen des Detaillierungsgrades von rsecomm.log finden Sie in „rsecomm.properties - RSE-Trace (optional)“ auf Seite 125.

CRASTART-Protokolldateien erfassen

CRASTART erstellt eine TSO-Umgebung als untergeordneten Prozess von RSE, der in einem separaten Adressraum ausgeführt wird. Es sind möglicherweise nicht-triviale Aktionen erforderlich, damit die CARMA-Ausgabe gespeichert wird, die nach SYSOUT(*) gesendet wird. Dadurch wird das Erfassen der Protokolldateien erschwert. Sie können dieses Problem lösen, indem Sie die Protokolldateien in eine benutzerspezifische Datei schreiben, wie in der folgenden Musterzuordnung dargestellt:

```
-COMMAND=ALLOC FI(CARMALOG) MOD CATALOG DSORG(PS) RECFM(F,B) LRECL(133)
          BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
          DA(&CRAUSER..&SYSNAME..CRA.CARMALOG)
```

Anmerkung:

- Aufgrund von Seitenbreitenbegrenzungen wurden einige Zeilen im Beispiel in die nächste Zeile umgebrochen. Alle Zeilen, die mit einer Einrückung beginnen, sollten am Ende der vorherigen Zeile hinzugefügt werden.
- Damit benutzerspezifische Protokolldateien erstellt werden können, muss diese Protokolldatei mit der Anweisung -COMMAND zugeordnet werden.
- Sie können die Protokolldateien auch in einer Zuordnungs-Exec zuordnen, wenn Sie mehr Flexibilität benötigen, z. B. um das Protokoll nur für bestimmte Benutzer in eine Datei zu senden. Weitere Informationen zu Zuordnungs-Execs finden Sie in „Angepasste Zuordnungs-Exec (optional)“ auf Seite 95.

Wenn Sie Protokolldateien an SYSOUT schreiben, beachten Sie, dass die SYSOUT-Zuordnung von z/OS UNIX-Prozessen in JES als spezielle Ausgabe behandelt wird. Dies ist ähnlich wie die SYSOUT-Zuordnung von APPC-Transaktionen.

- Solange der CARMA-Server noch aktiv ist, kann die Ausgabe mit dem Befehl **DA** in SDSF angezeigt werden. Der Job verwendet die Benutzer-ID des Benutzers gefolgt von einer wahlfreien, einstelligen Zahl als Jobnamen und eine STC-Job-ID. Der Benutzer ist der Jobeigner.
- Wenn die Ausgabe in eine HOLD-Ausgabeklasse geschrieben wurde, kann die Ausgabe, wenn der CARMA-Server aufgrund von Inaktivität oder durch das Trennen der Verbindung durch den Benutzer beendet wird, mit den Befehlen **APPC ON** und **H ALL** in SDSF angezeigt werden. Der Jobname, die Job-ID und der Jobeigner werden nicht geändert. Jede DD wird als separate Spooldatei ohne Angabe der eigentlichen DD angezeigt.
- JES Job Monitor kann ebenfalls die Ausgabe anzeigen, wenn SEARCHALL=ON in FEJCNFG aktiv ist und sich die Ausgabe im Spool in einer HOLD-Ausgabeklasse befindet. Weitere Informationen zur Anweisung SEARCHALL finden Sie in „FEJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34.

CRASUB* - Serverstart mit Batchübergabe

RSE startet die CLIST CRASUB*, die wiederum eine eingebettete JCL übergibt, um eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen zu erstellen. Developer for System z führt in dieser Umgebung den CARMA-Server (CRASERV) aus.

CRASUB* befindet sich in FEK.#CUST.CNTL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Anmerkung: Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

Developer for System z stellt mehrere CRASUB*-JCL-Jobs bereit. Jede dieser Musterdateien ist für ein bestimmtes Anpassungsszenario vorkonfiguriert:

- CRASUBCA ist für den Batchstart mit dem CA Endevor® SCM-RAM konfiguriert.
- CRASUBMT ist für den Batchstart mit Muster-RAM konfiguriert.

Abb. 23 auf Seite 83 zeigt ein grundlegendes "CRASUB*" -Skeleton, das ISPF-Services einschließt.


```

PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
/* SET CRAPRM2=420
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//*
//* DD used by CARMA server (CRASERV)
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//CRADEF  DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG  DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (ISPSTART)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPLIB  DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=#hlq.SFEKPROC          * CRAALLOC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRALLOC ISPSTART PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &CRAPRM3 +
                                     &CRAPRM4 &CRAPRM5 &CRAPRM6 &CRAPRM7 &CRAPRM8 )
$$
EXIT CODE(0)

```

Abbildung 23. CRASUB*: CARMA-Start mit Batchübergabe

Anmerkung:

- Wenn Sie die SYSTSIN-Daten ändern, stellen Sie sicher, dass mindestens ein Leerzeichen vor der schließenden runden Klammer (") der PARM()-Anweisung steht, um die Verarbeitung der Zeichenfolge zu vereinfachen.
- Sie können Ihre eigenen DD-Anweisungen und Dateiverkettungen hinzufügen, um die CARMA-TSO-Umgebung anzupassen und so eine TSO-Anmeldeprozedur zu imitieren.
- Sie können optional den Wert für das CARMA-Zeitlimit ändern, indem Sie die Kommentarzeichen für die Zeile the SET CRAPRM2=420 in der CLIST CRASUB* entfernen und diese ändern. Das Zeitlimit gibt die Zeit (in Sekunden) an, die CARMA auf den nächsten Befehl vom Client wartet. Wenn Sie den Wert 0 festlegen, wird das Standardzeitlimit verwendet, das derzeit bei 420 Sekunden (7 Minuten) liegt.
- Details des CARMA-Startprozesses werden in der Datei rsecomm.log angezeigt, wenn der Server beendet wird. Weitere Informationen zum Festlegen des Detailierungsgrades von rsecomm.log finden Sie in „rsecomm.properties - RSE-Trace (optional)“ auf Seite 125.

CARMA-VSAM-Dateien

Der CARMA-Server erfordert Lesezugriff auf drei VSAM-Dateien. Die Beispielmuster zum Erstellen und Füllen dieser VSAM-Dateien befinden sich in

FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Anmerkung:

- Wenn Sie die Definitionen für einen (benutzerdefinierten) RAM mit einer vorhandenen VSAM-Konfiguration zusammenführen müssen, finden Sie entsprechende Informationen im Beispieljob FEK.#CUST.JCL(CRA#UADD). Dieser Job muss für jede geänderte CARMA-VSAM-Datei angepasst und übergeben werden. Weitere Informationen zur Satzstruktur, die von den verschiedenen CARMA-VSAM-Dateien verwendet wird, finden Sie im Handbuch *Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).
- Verwenden Sie den Beispieljob FEK.#CUST.JCL(CRA#UQRY), um die aktiven Definitionen aus einer VSAM in eine sequenzielle Datei zu extrahieren.

CRADEF - Konfigurationsdatei

Diese VSAM-Datei beschreibt die Funktionen, die von den definierten RAM unterstützt werden. RAM-Entwickler benötigen Aktualisierungszugriff auf diese Datei. Die Datei kann mit einem der folgenden Musterjobs erstellt werden:

- CRA\$VCAD füllt die Datei mit Daten für den CA Endevor® SCM-RAM.
- CRA\$VDEF füllt die Datei mit Daten für den Muster-RAM.

Mithilfe der erwähnten Beispieljobs können Sie einen definierten RAM bei der VSAM-Erstellung inaktivieren. Dadurch können Sie ein angepasstes CARMA-Setup unter Verwendung einer einzigen Master-Eingabedatei erstellen, die durch IBM bereitgestellt oder durch Ihre RAM-Entwickler angepasst werden kann.

CRAMSG - Nachrichtendatei

Diese VSAM-Datei enthält Nachrichten, die vom CARMA-Server ausgegeben werden. Die Datei kann mit einem der folgenden Musterjobs erstellt werden:

- CRA\$VMSG füllt die Datei mit generischen Serverdaten.

CRASTRS - Datei mit angepassten Zeichenfolgen

Diese VSAM-Datei enthält die Nachrichten, die von den definierten RAM ausgegeben werden. RAM-Entwickler benötigen Aktualisierungszugriff auf diese Datei. Die Datei kann mit einem der folgenden Musterjobs erstellt werden:

- CRA\$VCAS füllt die Datei mit Daten für den CA Endevor® SCM-RAM.
- CRA\$VSTR füllt die Datei mit Daten für den Muster-RAM.

Migrationshinweise für CARMA-VSAM

1. Ab Version 7.6.1 unterstützt Developer for System z ein neues Datenstrukturlayout für die VSAM-Datei für angepasste CARMA-Informationen (CRASTRS), um Längenbeschränkungen für Nachrichten zu vermeiden.

Vor Developer for System z Version 7.6.1 waren Zeichenfolgen, die in VSAM-Dateien für angepasste CARMA-Informationen definiert werden, auf bestimmte vordefinierte Längen begrenzt. Diese Begrenzung zwingt RAM-Entwickler, beschreibende Zeichenfolgen zu kürzen oder clientseitige Plug-ins zu verwenden, um Zeichenfolgen in vollständiger Länge anzuzeigen.

Die neue VSAM-Satzstruktur unterstützt ein Datenstrukturlayout mit variabler Länge für die VSAM-Datei für angepasste CARMA-Informationen (CRASTRS), bei dem Zeichenfolgen durch ein Begrenzungszeichen getrennt werden, anstatt einer festen Länge zu unterliegen.

Passen Sie die JCL FEK.SFEKSAMP(CRA#VS2) an und übergeben Sie sie, um Ihre vorhandene VSAM-Datei für angepasste CARMA-Informationen mit fester Länge (CRASTRS) in eine Datei mit neuem Format mit variabler Länge umzuwandeln.

Anmerkung:

- Ab Version 7.6.1 wird die VSAM-Beispieldatei für angepasste CARMA-Informationen im Format mit variabler Länge bereitgestellt.
 - Ab Version 7.6.1 unterstützt das CARMA-Lademodul (CRASERV) sowohl Formate mit festen Längen als mit variablen Längen für die VSAM-Datei für angepasste CARMA-Informationen.
 - Ältere Versionen des CARMA-Lademoduls unterstützen keine Formate mit variabler Länge. Wenn trotzdem eine VSAM-Datei für angepasste CARMA-Informationen mit einem Format mit variabler Länge verwendet wird, werden die Zeichenfolgen beschädigt.
2. Durch die Einführung von RAM für CA Endevor® SCM-Pakete ist es nicht mehr erforderlich, paketbezogene Aktionen im RAM des CA Endevor® SCM-Elements bereitzustellen. Daher wird das Bereitstellen von paketbezogenen Aktionen im CA Endevor® SCM-RAM nicht mehr unterstützt und die CARMA-VSAM-Dateien in Version 8.5 beinhalten nicht mehr die für diese Funktionalität erforderlichen Definitionen.

Sie können diese Funktionalität wiederherstellen, indem Sie die JCL FEK.SFEKSAMP(CRA#UADD) anpassen und übergeben, um die entfernten Elemente zurück in die CARMA-VSAM-Dateien zu stellen.

- Führen Sie FEK.SFEKVSM2(CRA0VPKD) mit FEK.#CUST.CRADEF zusammen.
- Führen Sie FEK.SFEKVSM2(CRA0VPKS) mit FEK.#CUST.CRASTRS zusammen.

Diese Zusammenführungsaktion ist bei jedem Aktualisieren der VSAM-Dateien während der Produktverwaltung erforderlich.

CARMA-RAM (Repository Access Manager)

Ein Repository Access Manager (RAM) ist eine Anwendungsprogrammierschnittstelle (API) für einen z/OS-basierten SCM (Software Configuration Manager). Ein CARMA-Server, der die RAM lädt und eine Standardschnittstelle für den Zugriff auf den SCM bereitstellt, kann durch Developer for System z oder durch von Benutzern geschriebene Anwendungen gestartet werden.

Der CARMA-Server muss die RAM-Lademodule über LINKLIST oder über STEPLIB/TASKLIB finden können.

Die RAM-Lademodule CRAR*, die von Developer for System z bereitgestellt werden, befinden sich in FEK.SFEKLOAD und der Beispiel Quellcode und die Kompilierjobs befinden sich in FEK.SFEKSAMP, sofern Sie bei der SMP/E-Installation von Developer for System z kein anderes übergeordnetes Qualifikationsmerkmal verwendet haben.

Die folgenden Abschnitte enthalten Anpassungshinweise für die RAM, die in Developer for System z verfügbar sind. Die Beispielmembers, auf die verwiesen wird, befinden sich in FEK.#CUST.*, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Eingehende Informationen zu CARMA sowie weitere Informationen zu den bereitgestellten Muster-RAM und zum bereitgestellten Muster Quellcode finden Sie im Handbuch *Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).

CA Endeavor® SCM-RAM

- Der CA Endeavor® SCM-RAM ist ein Produktions-RAM.
- Der CA Endeavor® SCM-RAM ermöglicht Clients von Developer for System z den direkten Zugriff auf CA Endeavor® SCM-Elemente.
- Der Name des Lademoduls lautet CRARN DVR.
- Der CA Endeavor® SCM-RAM verwendet viele zusätzliche Einstellungen im Vergleich zu einer konventionellen CARMA-Konfiguration. Verwenden Sie eine der vorkonfigurierten Konfigurationen, die den CA Endeavor® SCM-RAM unterstützen, als Ausgangspunkt und passen Sie diese an Ihre Anforderungen an.
- Die veraltete Startmethode für das TSO/ISPF-Client-Gateway kann nicht mit dem CA Endeavor® SCM-RAM verwendet werden.
- Der CA Endeavor® SCM-RAM verwendet mehrere Konfigurationsdateien, FEK.#CUST.PARMLIB(CRACFG), FEK.#CUST.PARMLIB(CRASHOW) und FEK.#CUST.PARMLIB(CRATMAP), die angepasst werden können. Weitere Informationen finden Sie in „CRACFG, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endeavor® SCM-RAM“ auf Seite 87.
- Der CA Endeavor® SCM-RAM verwendet eine Zuordnungs-Exec (FEK.SFEKPROC-(CRANDVRA)), die angepasst werden kann. Weitere Informationen hierzu enthält der Abschnitt „CRANDVRA - Zuordnungs-Exec für den CA Endeavor® SCM-RAM“ auf Seite 88.
- Der CA Endeavor® SCM-RAM unterstützt die Ausführung von CA Endeavor® SCM-Aktionen im Batch-Modus im Hintergrund. Für Batch-Aktionen ist eine Konfigurationsdatei (FEK.#CUST.PARMLIB(CRABCFG)) und eine Skeleton-JCL (FEK.#CUST.CNTL(CRABATCA)) erforderlich, die angepasst werden müssen. Weitere Informationen finden Sie in „Batch-Aktionen für CA Endeavor® SCM-RAM“ auf Seite 89.

RAM für CA Endeavor® SCM-Pakete

- Der RAM für CA Endeavor® SCM-Pakete ist ein Produktions-RAM.
- Der RAM für CA Endeavor® SCM-Pakete ermöglicht Clients von Developer for System z den direkten Zugriff auf CA Endeavor® SCM-Pakete.
- Der Name des Lademoduls lautet CRARPKGS.
- Der RAM für CA Endeavor® SCM-Pakete hat keine anpassbaren Einstellungen und muss in Verbindung mit dem CA Endeavor® SCM-RAM verwendet werden.

PDS-RAM

- Der PDS-RAM ist ein Muster-RAM. Verwenden Sie diesen RAM nicht in einer Produktionsumgebung.
- Der PDS-RAM gibt eine Dateiliste ähnlich zu **MVS-Dateien -> Meine Dateien** in der Ansicht 'Ferne Systeme' zurück.
- Der Name des Lademoduls lautet CRARPDS.
- Für den PDS-RAM wird vorausgesetzt, dass ISPF-Services verfügbar sind.
- Für den PDS-RAM ist es erforderlich, dass der DD CRARAM1 eine zusätzliche VSAM-Datei zugeordnet wird. Diese VSAM-Datei kann mit dem Musterjob FEK.#CUST.JCL(CRA#VPDS) zugeordnet und vorbereitet werden. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.

- Quellcode und Kompilierjobs sind in FEK.SFEKSAMP verfügbar. Weitere Informationen finden Sie in *Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).

Skeleton-RAM

- Der Skeleton-RAM ist ein Muster-RAM. Verwenden Sie diesen RAM nicht in einer Produktionsumgebung.
- Der Skeleton-RAM gibt ein funktionales Gerüst zurück, das als Ausgangspunkt für die Entwicklung Ihres eigenen RAM verwendet werden kann.
- Der Name des Lademoduls lautet CRARTEST.
- Quellcode und Kompilierjobs sind in FEK.SFEKSAMP verfügbar. Weitere Informationen finden Sie im Handbuch *Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).

SCLM-RAM

- Der SCLM-RAM ist ein Muster-RAM. Verwenden Sie diesen RAM nicht in einer Produktionsumgebung.
- Der SCLM-RAM gibt einen Basiseintrag in SCLM (Software Configuration Manager von ISPF) zurück. Dieser RAM ist standardmäßig nicht aktiviert.
- Der Name des Lademoduls lautet CRARSCLM.
- Der SCLM-RAM benötigt die ISPF-Services, um verfügbar zu sein.
- Für den SCLM-RAM ist es erforderlich, dass der DD CRARAM2 eine zusätzliche VSAM-Datei zugeordnet wird. Diese VSAM-Datei kann mit dem Musterjob FEK.#CUST.JCL(CRA#VSLM) zugeordnet und vorbereitet werden. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.
- Für den SCLM-RAM müssen die verschiedenen benutzerspezifischen Dateien vorhanden sein. Passen Sie FEK.#CUST.JCL(CRA#ASLM) an, damit diese Dateien zugeordnet werden. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Vor der Verwendung von CARMA mit dem SCLM-RAM muss jeder Benutzer CRA#ASLM einmal übergeben. Andernfalls kommt es zu einem Zuordnungsfehler.
- Der SCLM-RAM ist standardmäßig nicht aktiviert. Damit der RAM aktiviert wird, muss er in den CARMA-VSAM-Dateien definiert werden, auf die die DD CRADEF und CRASTRS verweisen. Verwenden Sie den Musterjob FEK.#CUST.JCL(CRA#UADD), um FEK.SFEKVSM2(CRA0SLMD) in CRADEF und FEK.SFEKVSM2(CRA0SLMS) in CRASTRS zusammenzuführen. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.
- Quellcode und Kompilierjobs sind in FEK.SFEKSAMP verfügbar. Weitere Informationen finden Sie im Handbuch *Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).

CRACFG, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endevor® SCM-RAM

Die folgenden, für den CA Endevor® SCM-RAM spezifischen CARMA-Komponenten können unabhängig von der gewählten Serverstartmethode angepasst werden. Die unten angegebenen Beispielmembers befinden sich in FEK.#CUST.PARMLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP (FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

CRACFG, Interaktion von CA Endevor® SCM-RAM mit dem SCM

CRACFG gibt an, wie CA Endevor® SCM-RAM mit CA Endevor® SCM interagiert. Wenn Sie die Standardeinstellungen ändern möchten, finden Sie Anpassungsanweisungen in der im Member enthaltenen Dokumentation.

```
# ENTRY-STAGE-COPY-MODE = RETRIEVE-ADD
```

Abbildung 24. CRACFG - Interaktion von CA Endevor® SCM-RAM mit dem SCM

"CRASHOW": CA Endevor® SCM-RAM-Standardfilter

CRASHOW definiert Standardfilter für CA Endevor® SCM-Umgebungen, -Systeme usw. Wenn Sie die Standardeinstellungen ändern möchten, finden Sie Anpassungsanweisungen in der im Member enthaltenen Dokumentation.

```
ENV=*
TOENV=
STGID=*
TOSTGID=
SYS=*
SUBSYS=*
ELEM=*
TOELEM=
TYPE=*
#FILTER-DEP=YES
```

Abbildung 25. CRASHOW - CA Endevor® SCM-RAM-Standardfilter

Anmerkung: FILTER-DEP ist keine allgemeine CA Endevor® SCM-Variable, sondern eine für Developer for System z spezifische Variable, die die Abhängigkeitsscans für Elemente mit Speicherbedarfsverweisen auf andere CA Endevor® SCM-Repositorypositionen steuert.

CRATMAP - Zuordnungen von CA Endevor® SCM-RAM-Dateierweiterungen

CRATMAP überschreibt den CA Endevor® SCM-Typ mit Dateierweiterungszuordnungen. Wenn Sie die Standardwerte ändern wollen, finden Sie entsprechende Anpassungsanweisungen in der im Member enthaltenen Dokumentation.

```
# *      = cbl
# COBOL  = cbl
# COPY   = cpy
# ASM    = asm
# MACRO  = asm
# PROCESS = jcl
```

Abbildung 26. CRATMAP: CA Endevor® SCM-RAM-Standardfilter

CRANDVRA - Zuordnungs-Exec für den CA Endevor® SCM-RAM

Die Startmethode mit Batchübergabe und die Startmethode mit CRASTART rufen beide die REXX-Exec CRANDVRA auf, um benutzerspezifische Dateien zuzuordnen, die vom CA Endevor® SCM-RAM verwendet werden. Die Zuordnungen erfolgen in einer separaten Exec, weil eine Exec mehr Flexibilität ermöglicht als die Funktionen in der Batchübergabe-JCL CRASUBCA und in der CRASTART-Konfigurationsdatei crastart.endevor.conf. Die Zuordnungs-Exec ruft zudem den optionalen Benutzerexit auf.

DD-Anweisung	Dateiname	Typ
DEPEND	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.DEPEND	Permanent
BROWSE	&SYSPREF..&SYSUID..&SYSNAME..CRA\$NDVR. BROWSE	Temporär
C1PRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	Temporär
SPCLLIST	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.SPCLLIST	Temporär
PKGSCLS	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.PKGSCLS	Temporär
CRABJCLO	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRABJCLO	Temporär
ENHCEDIT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDIT	Temporär
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRAPARM	Temporär

Sie können eine Kopie dieser Zuordnungs-REXX-Exec anpassen, falls bestimmte Standardwerte, wie der Dateiname, nicht den Standards Ihres Standorts entsprechen. CRANDVRA befindet sich in FEK.SFEKPROC, sofern Sie bei der SMP/E-Installation von Developer for System z kein anderes übergeordnetes Qualifikationsmerkmal verwendet haben.

Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Weitere Informationen zu Zuordnungs-Execs finden Sie in „Angepasste Zuordnungs-Exec (optional)“ auf Seite 95.

Anmerkung: Sie sollten die Beispielzuordnungs-REXX in eine neue Datei kopieren und diese Kopie anpassen, damit die Konfiguration im Falle einer Wartung nicht überschrieben wird. Wenn Sie dies tun, müssen Sie den Verweis auf SFEKPROC in der DD-Anweisung 'SYSEXEC' Ihrer gewählten CARMA-Startmethode aktualisieren, damit sie mit Ihrem neuen Dateinamen übereinstimmt.

Batch-Aktionen für CA Endevor® SCM-RAM

Normalerweise werden CA Endevor® SCM-Aktionen wie “Generate Element” (Element generieren) online im Serveradressraum von CARMA ausgeführt. Dieses Verhalten führt zu Problemen, wenn Ihre CA Endevor® SCM-Prozeduren TSO aufrufen, da TSO bereits aktiv ist und daher die erforderlichen DDs wie SYSTSIN und SYSTSPRT bereits verwendet werden.

Zur Lösung dieses Problems unterstützt CA Endevor® SCM-RAM ab Version 8.0.3 Batch-Aktionen. Bei aktivierten Batch-Aktionen übergibt der CA Endevor® SCM-RAM einen anpassbaren Batch-Job, um Aktionen wie “Generate Element” (Element generieren) auszuführen. Durch die Verwendung eines Batch-Jobs können DDs wie SYSTSIN und SYSTSPRT von Ihren CA Endevor® SCM-Prozeduren zugeordnet werden, da TSO für die übergebene JCL nicht aktiv sein muss.

CA Endevor® SCM-RAM-Batch-Aktionen entsprechen den CA Endevor® SCM-Hintergrundaktionen von Developer for System z.

Wenn eine Anforderung zur Ausführung einer Aktion übergeben wird, die von Batch-Aktionen unterstützt wird, überprüft der CA Endeavor® SCM-RAM, ob die DD CRABCFG in CRASUBCA oder crastart.endevor.conf vorhanden ist, und überprüft, ob die Installation hinter dieser DD gültig ist. Wenn CRABCFG vorhanden und die Installation gültig ist, wird die Aktion im Batch ausgeführt. Wenn CRABCFG nicht vorhanden ist, wird die Aktion online ausgeführt. Clients ab Version 8.0.3 haben die Möglichkeit, dieses Verhalten zu überschreiben.

Beispiel:

```
//* Kommentarzeichen für CRABCFG und CRABSKEL für die Verwendung von Batch-Aktionen entfernen
//*CRABCFG DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRABCFG)
//*CRABSKEL DD DISP=SHR,DSN=FEK.#CUST.CNTL
```

Anmerkung:

- Die Umgebung ohne TSO ist nur für ausgewählte CA Endeavor® SCM-Aktionen verfügbar. Batch-Aktionen unterstützen keine Umgebung ohne TSO außerhalb dieses Bereichs.
- Die Konfigurationsdatei CRABCFG dokumentiert, welche CA Endeavor® SCM-Aktionen unterstützt werden.
- Mit dem bereitgestellten funktionalen Beispieljob FEK.#CUST.CNTL(CRABATCA) können die Batch-Aktionen ausgeführt werden. Allerdings sind die Batch-Aktionen so konzipiert, dass dieses Muster angepasst werden sollte, um Ihre aktuellen CA Endeavor® SCM-Prozeduren zu starten.
- Stellen Sie sicher, dass ausreichend JES-Initiatoren in der Klasse verfügbar sind, mit der JCLs für die Batch-Aktion übergeben werden.
- Stellen Sie bei der Verwendung von JES in einer SYSPLEX-Umgebung sicher, dass der Job auf dem aktuellen System ausgeführt wird oder dass die Informationen zur Beendigung zurück zu dem System geleitet werden, auf dem sich Developer for System z befindet, damit der CA Endeavor® SCM-RAM den Status prüfen kann.
- Wenn Developer for System z-Client und -Hostsystem eine Version ab 8.5.1 haben, kann der Client eine angepasste JOB-Karte und zusätzliche JCL-Anweisungen für die Batch-Aktion der JCL vor der Übergabe bereitstellen.

CRABCFG - Konfiguration der Batch-Aktion für den CA Endeavor® SCM-RAM

CRABCFG definiert die Konfigurationsvariablen, die sich auf Batch-Aktionen des CA Endeavor® SCM-RAM beziehen.

CRABCFG befindet sich in FEK.#CUST.PARMLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Die folgende Beispieldatei CRABCFG muss an Ihre Systemumgebung angepasst werden. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit dem Nummernzeichen (#). Kommentarzeichen nach einer Anweisung und der zugehörige Wert werden unterstützt. Leerzeichen vor und nach dem Gleichheitszeichen (=) werden unterstützt. Zeilenfortsetzungen werden nicht unterstützt.

Anmerkung: Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.


```
# Position der Batch-Aktions-JCL
SKELETON-DD = CRABSKEL
#
# JCL-Members der Batch-Aktion in SKELETON-DD
DEFAULT-JOBCARD = CRABJOBC
ADD-ELEMENT      = CRABATCA
GENERATE-ELEMENT = CRABATCA
MOVE-ELEMENT     = CRABATCA
DELETE-ELEMENT   = CRABATCA
RETRIEVE-ELEMENT = CRABATCA
SIGNIN-ELEMENT   = CRABATCA
PRINT-ELEMENT    = CRABATCA
PRINT-MEMBER     = CRABATCA
#
# Befehlssubstitutionsschlüssel in Batch-Aktions-JCL
BSTIPT01-KEY = <CRA_BSTIPT01>
```

Abbildung 27. CRABCFG: Konfiguration der Batch-Aktion für den CA Endavor® SCM-RAM

SKELETON-DD

Name der DD-Anweisung, die auf mindestens eine PDS(E)-Datei mit Skeleton-JCLs der Batch-Aktion verweist. Der Beispielwert ist CRABSKEL. Bei Bedarf können Sie diesen Wert ändern. Diese DD muss für den CARMA-Server in CRASUBCA oder crastart.endavor.conf definiert werden.

DEFAULT-JOBCARD

Name des Members, das eine Standardjobkarte enthält. Falls nicht durch eine benutzerspezifische Jobkarte, die auf dem Developer for System z-Client ab Version 8.5.1 gespeichert ist, außer Kraft gesetzt, wird diese Standardjobkarte verwendet, um den Schlüssel <JOBCARD> in einer Skeleton-JCL zu ersetzen. Bei Bedarf können Sie diesen Wert ändern.

GENERATE-ELEMENT und weitere CA Endavor® SCM-Aktionen

Die Schlüsselnamen stellen die CA Endavor® SCM-Aktionen dar, die von der Batch-Aktion unterstützt werden, und können nicht geändert werden. Der jedem Schlüssel zugeordnete Wert ist der Mitgliedsname der zugehörigen Skeleton-JCL. Der Beispielwert für alle Schlüssel ist CRABATCA. Bei Bedarf können Sie diesen Wert ändern.

BSTIPT01-KEY

Substitutionsschlüssel für die tatsächliche CA Endavor® SCM-Befehlszeichenfolge. Der Beispielwert ist <CRA_BSTIPT01>. Bei Bedarf können Sie diesen Wert ändern. Das erste Vorkommen dieses Substitutionsschlüssels außerhalb eines Kommentars wird in der Skeleton-JCL durch die Befehlszeichenfolge ersetzt, mit der CA Endavor® SCM angewiesen wird, die angeforderte Aktion für das angeforderte Element auszuführen.

CRABATCA - JCL für Batch-Aktionen des CA Endavor® SCM-RAM

CRABATCA ist eine Beispiel-Skeleton-JCL, die für Batch-Aktionen verwendet wird. Wenn Sie die Standardwerte ändern wollen, finden Sie entsprechende Anpassungsanweisungen in der im Member enthaltenen Dokumentation.

CRABATCA befindet sich in FEK.#CUST.CNTL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Die Änderungen sind für alle neuen Aufrufe aktiv. Es ist kein Serverneustart erforderlich.


```

//<JOB CARD>
//*
//CRABATCA EXEC PGM=NDVRC1,DYNAMNBR=1500,REGION=4096K,PARM='C1BM3000'
//STEPLIB DD DISP=SHR,DSN=CA.NDVRU.AUTHLIB * NDVR R12
// DD DISP=SHR,DSN=CA.NDVR.AUTHLIB * NDVR R12
//* DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU * NDVR R14
//* DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH * NDVR R14
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CONLIB * NDVR R12
//*CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD * NDVR R14
//C1MSG1 DD SYSOUT=*
//C1MSG2 DD SYSOUT=*
//C1PRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYMDUMP DD DUMMY
//SYSIN DD DUMMY
//BSTIPT01 DD *
SET STOPRC 16 .
<CRA_BSTIPT01>
//

```

Abbildung 28. CRABATCA: JCL für die Batch-Aktion des CA Endeavor® SCM-RAM

CRABJOBC - Jobkarte für Batch-Aktionen des CA Endeavor® SCM-RAM

CRABJOBC ist eine standardmäßige Beispieljobkarte, die für Skeleton-JCL mit Batch-Aktionen verwendet wird, die den Schlüssel <JOB CARD> angibt. Wenn Sie die Standardwerte ändern wollen, finden Sie entsprechende Anpassungsanweisungen in der im Member enthaltenen Dokumentation.

CRABJOBC befindet sich in FEK.#CUST.CNTL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Die Änderungen sind für alle neuen Aufrufe aktiv. Es ist kein Serverneustart erforderlich.

```

//<USERID>B JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*PROCS JCLLIB ORDER=(COBOL.V4R1M0.SIGYPROC,CBC.SCCNPRC)

```

Abbildung 29. CRABJOBC: Jobkarte für Batch-Aktionen des CA Endeavor® SCM-RAM

CRAALLOC - Angepasste Zuordnungs-Exec für den RAM

Die Startmethode mit Batchübergabe und die Startmethode mit CRASTART rufen die REXX-Exec CRAALLOC auf, um benutzerspezifische Dateien zuzuordnen, die von einem benutzergeschriebenen RAM verwendet werden können. Die Zuordnungen erfolgen in einer separaten Exec, da eine Exec mehr Flexibilität bietet, als die Funktionen in der Batchübergabe-JCL CRASUBMT und in der CRASTART-Konfigurationsdatei crastart.conf. Die Zuordnungs-Exec ruft zudem den optionalen Benutzerexit auf.

DD-Anweisung	Dateiname	Typ
CRAPARM	&SYSPREF.&SYSUID.. &SYSNAME..CRA\$CUST.CRAPARM	Temporär

Sie können eine Kopie dieser Zuordnungs-REXX-Exec anpassen, falls bestimmte Standardwerte, wie der Dateiname, nicht den Standards Ihres Standorts entsprechen. CRAALLOC befindet sich in FEK.SFEKPROC, sofern Sie bei der SMP/E-Installation von Developer for System z kein anderes übergeordnetes Qualifikationsmerkmal verwendet haben.

Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Weitere Informationen zu Zuordnungs-Execs finden Sie im Abschnitt Angepasste Zuordnungs-Exec (optional).

Anmerkung: Sie sollten die Beispielzuordnungs-REXX in eine neue Datei kopieren und diese Kopie anpassen, damit die Konfiguration im Falle einer Wartung nicht überschrieben wird. Wenn Sie dies tun, müssen Sie den Verweis auf SFEKPROC in der DD-Anweisung 'SYSEXEC' Ihrer gewählten CARMA-Startmethode aktualisieren, damit sie mit Ihrem neuen Dateinamen übereinstimmt.

CARMA-Rückkehrcodes

CARMA kann verschiedene Fehlercodes an den Client oder in den Hostsystemprotokollen zurückmelden. Die mit dem Fehler bereitgestellten Details und die Informationen in Tabelle 12 können Sie dabei unterstützen, den Fehler zu finden und eine Lösung zu erarbeiten.

Tabelle 12. CARMA-Rückkehrcodes

Fehlerbereich	Fehlertyp
4-99	Allgemeine CARMA-Fehler
100-199	Allgemeine RAM-Fehler
200-399	CRASERV-Fehler (CARMA-Server)
400-499	RSE-Fehler (CARMA-Miner)
500-899	RAM-spezifische Fehler
900-999	TSO- und TCP/IP-Fehler

Die folgenden Rückkehrcodes treten häufiger auf:

- 220: Der CARMA-Server wird aufgrund des Inaktivitätszeitlimits beendet. Dies ist kein Fehler.
- 990: Der CARMA-Server kann keine Verbindung mit dem Port herstellen, an dem der CARMA-Miner empfangsbereit ist.

(Optional) Unterstützung mehrerer RAM

CARMA verfügt über die Funktionalität zum Definieren mehrerer RAM und zum gleichzeitigen Ausführen dieser RAM. Da jedoch auch bei mehreren RAM nur ein CARMA-Server für einen Benutzer aktiv ist, müssen möglicherweise einige Änderungen an der Konfiguration vorgenommen werden, damit sie funktioniert.

RAM werden von einem RAM-Entwickler in der VSAM-Datei CRADEF in der CARMA-Konfiguration definiert. Während des Starts erkennt der CARMA-Server CRASERV alle definierten RAM und sendet die Informationen an den CARMA-Client. Der Benutzer kann dann einen oder mehrere RAM auswählen, die in den CARMA-Server geladen werden.

Da RAM als Plug-ins des CARMA-Servers aktiv sind, müssen Sie sicherstellen, dass alle Voraussetzungen, wie Dateizuordnungen, für jeden der RAM im Adress-

raum des CARMA-Servers verfügbar sind. Diese Anforderung kann Änderungen an den CARMA-Konfigurationsbeispielen wie CRASUBMT oder crastart.conf erfordern, die in Developer for System z enthalten sind.

Beispiel:

Im folgenden Beispiel wird eine vorhandene Konfiguration des CA Endevor® SCM-RAMs mit der Startmethode 'CRASTART' gestartet und anschließend der Beispiel-PDS-RAM hinzugefügt.

Definitionen für den CA Endevor® SCM-RAM:

- FEK.SFEKVSM2(CRA0VCAD): CRADEF-Definitionen
- FEK.SFEKVSM2(CRA0VCAS): CRASTRS-Definitionen
- /etc/rdz/crastart.endevor.conf: CRASTART-Konfigurationsdatei

Definitionen für den PDS-RAM:

- FEK.SFEKVSM2(CRA0VDEF): CRADEF-Definitionen
- FEK.SFEKVSM2(CRA0VSTR): CRASTRS-Definitionen
- FEK.#CUST.CRARAM1: CRARAM1-Definitionen

Zunächst sammelt der RAM-Entwickler alle Daten und Informationen, die der Systemprogrammierer benötigt, um die Konfiguration durchzuführen.

1. Extrahieren Sie die für den PDS-RAM spezifischen Daten aus den SFEKVSM2-Memberrn. Diese Member enthalten Definitionen für alle Muster-RAM, nicht nur für den PDS-RAM.
2. Führen Sie diese Daten mit den SFEKVSM2-Memberrn des Endevor® SCM-RAMs zusammen.
3. Erstellen Sie eine Liste mit den Voraussetzungen für den PDS-RAM:
 - DD-Anweisung 'CRARAM1', die mit FEK.#CUST.CRARAM1 verknüpft ist
 - TSO-Umgebung

Der Systemprogrammierer verwendet diese Daten anschließend, um die aktualisierten CARMA-VSAM-Dateien zu erstellen. Mithilfe der Informationen zu den Voraussetzungen erstellt er eine CRASTART-Konfigurationsdatei, die beide RAM unterstützt.

1. Verwenden Sie die kombinierten Daten als Eingabe für die Jobs CRA\$VDEF und CRA\$VSTR, um die aktualisierte CARMA-Konfiguration und die VSAM-Dateien für angepasste Informationen (CRADEF und CRASTRS) zu erstellen. Die VSAM CRAMSG ist für den CARMA-Server spezifisch und somit für beide RAM identisch.

2. Fügen Sie crastart.endevor.conf eine CRARAM1-Definition hinzu:

```
CRARAM1 = FEK.#CUST.CRARAM1
```

3. Überprüfen Sie die PROGRAM-Anweisung in crastart.endevor.conf, um sicherzustellen, dass sie die Umgebung bereitstellen kann, die von beiden RAM benötigt wird:

```
PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV)  
PARM(&CRAPRM1. &CRAPRM2.)
```

- IKJEFT01: TSO wird verwendet, um bestimmte autorisierte Aufrufe in einer nicht autorisierten Umgebung zuzulassen. Des Weiteren wird TSO als Umgebung für die Ausführung der CA Endevor® SCM-RAM-Vorzuordnungs-Exec verwendet.

- %CRANDVRA: Vorzuordnungs-Exec des CA Endeavor® SCM-RAMs, die sich in FEK.SFEKPROC befindet und die temporäre benutzerspezifische Arbeitsdateien zuordnet.
- NDVRC1: CA Endeavor®-Back-End mit integriertem Mechanismus zur Ausführung von TSO- und ISPF-Befehlen.
- PGM(CRASERV): Befehl zum Starten eines CARMA-Servers, im ISPF-Befehlsformat.
- PARM(&CRAPRM1. &CRAPRM2.): Parameter für CRASERV, im ISPF-Befehlsformat. &CRAPRM1 ist der zu verwendende Port und &CRAPRM2 ist der Zeitlimitwert.

Der CA Endeavor® SCM-RAM ist in einer ISPF-Umgebung aktiv, die voraussetzt, dass die vom PDS-RAM benötigte TSO-Umgebung ebenfalls verfügbar ist.

Angepasste Zuordnungs-Exec (optional)

In Bezug auf die Dateizuordnung gelten bei allen CARMA-Serverstartmethoden Einschränkungen. Die TSO-Präfixersetzung ist in JCL oder CRASTART beispielsweise nicht verfügbar.

Wenn Sie jedoch eine Exec erstellen, die nach dem Start von TSO oder ISPF und vor dem Start von CARMA aufgerufen wird, können Sie den kompletten in TSO bzw. ISPF verfügbaren Variablen- und Serviceumfang nutzen, um die erforderlichen Zuordnungen vorzunehmen.

Developer for System z verwendet in den einzelnen vorkonfigurierten Konfigurationen, die zuvor in diesem Kapitel beschrieben wurden, eine Zuordnungs-Exec. FEK.SFEKPROC(CRANDVRA), die Zuordnungs-Exec für CA Endeavor® SCM-RAM, und FEK.SFEKPROC(CRAALLOC), die Zuordnungs-Exec für benutzerdefinierte RAMs. Die Exec ordnet katalogisierte temporäre Dateien zu, die das TSO-Präfix des Benutzers als übergeordnetes Qualifikationsmerkmal verwenden. Die Zuordnungs-Exec ruft zudem den optionalen Benutzerexit auf.

Anweisungen zur Anpassung sind der Exec dokumentiert. Das Ändern der Zuordnungs-Exec wird unterstützt, jedoch nicht empfohlen, da Anpassungen erneut vorgenommen werden müssen, wenn der Service für die vorläufige Programmkorrektur die Exec aktualisiert. Verwenden Sie, wenn möglich, stattdessen den in „CARMA-Benutzerexit (optional)“ auf Seite 96 beschriebenen CARMA-Benutzerexit.

Anmerkung:

- Stellen Sie beim Aktualisieren einer Zuordnungs-Exec sicher, dass Sie keine Zuordnungen löschen, die zu einem früheren Zeitpunkt im CARMA-Startprozess von CRASTART oder Ihrer Start-JCL definiert wurden.
- Die von der Zuordnungs-Exec generierte Ausgabe wird in der DD "SYSTSPRT" des CARMA-Servers dargestellt.

Stellen Sie beim Aktualisieren einer Zuordnungs-Exec sicher, dass Sie keine Zuordnungen löschen, die zu einem früheren Zeitpunkt im CARMA-Startprozess von CRASTART oder Ihrer Start-JCL definiert wurden.

Die folgenden Beispiele zeigen, wie Sie eine Zuordnungs-Exec starten, die nur TSO erfordert.

crastart*.conf

```

SYSPROC = my.exec.library
PROGRAM = IKJEFT01 %myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )

CRASUB*
//SYSPROC DD DISP=SHR,DSN=my.exec.library
//SYSTSIN DD *
%myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )
/**

```

CARMA-Benutzerexit (optional)

CARMA unterstützt den Aufruf eines Benutzerexits, um eine spezielle Initialisierung beim Start und eine spezielle Bereinigung beim Herunterfahren des CARMA-Servers zu ermöglichen. Durch die Verwendung eines Benutzerexits muss die Zuordnungs-Exec, die durch den Service für die vorläufige Programmkorrektur verwaltet wird, seltener geändert werden.

Der Benutzerexit wird durch die Zuordnungs-Exec aufgerufen und zweimal ausgeführt. Der Initialisierungsaufruf findet nach der Zuordnung der temporären Dateien und vor dem Aufrufen des CARMA-Servers statt. Der Bereinigungsaufruf findet nach dem Herunterfahren des CARMA-Servers und vor dem Löschen der temporären Dateien statt. Wenn der erste Aufruf mit dem Rückgabecode 99 oder höher endet, wird der CARMA-Start unterbrochen. Dies bedeutet, dass weder der CARMA-Server noch der zweite Aufruf dieses Benutzerexits ausgeführt wird.

Ein Beispielbenutzerexit wird als FEK.SFEKSAMP(CRAEXIT) zur Verfügung gestellt, sofern Sie bei der SMP/E-Installation von Developer for System z kein anderes übergeordnetes Qualifikationsmerkmal verwendet haben. Im folgenden Beispielbenutzerexit sind die an den Benutzerexit übergebenen Startargumente ausführlich dokumentiert:

Startargument	Beschreibung
(STARTUP) (ENDING)	Anzeiger dafür, ob der Exitaufruf vor oder nach dem Aufrufen des CARMA-Servers stattfindet.
EXIT_RC=rc	Rückgabecode des vorherigen Aufrufs des Exits. rc ist während des (STARTUP)-Aufrufs immer 0.
CARMA_RC=rc	Rückgabecode des Aufrufs des CARMA-Servers. rc ist während des (STARTUP)-Aufrufs immer 0.
...	Startbefehl und Startargumente des CARMA-Servers. Beispiel: ISPSTART PGM(CRASERV) PARM(1312 420 EXIT=CRAEXIT CLIENT=9.0.1)

Die vom Benutzerexit generierte Ausgabe wird in der DD "SYSTSPRT" des CARMA-Servers angezeigt.

IRXJCL oder CRAXJCL (optional)

Falls der CARMA-Server mit TSO (IKJEFTxx) gestartet wird, können Probleme auftreten, wenn Ihre RAM Services aufrufen, die ihrerseits die REXX-Batchschnittstelle IRXJCL aufrufen. Zu diesen Problemen kann es kommen, wenn die von RAM zuvor aufgerufenen Prozessoren bisher ohne TSO oder nur in Online-TSO gearbeitet haben und DD SYSTSIN oder SYSTSPRT dynamisch zuordnen. Zur Umgehung dieses Problems wird das Beispielprogramm CRAXJCL bereitgestellt.

Ein Versuch Ihres Prozessors, die für IRXJCL erforderliche DD SYSTSIN oder SYSTSPRT zuzuordnen, könnte fehlschlagen, weil diese DD-Namen bereits von der für CARMA erforderlichen Komponente Batch-TSO zugeordnet und geöffnet wurden. Das Ersatzmodul CRAXJCL versucht eine Zuordnung von SYSTSIN und SYSTSPRT zu DUMMY, ignoriert jedoch die Fehler, die bei fehlgeschlagenen Zuordnungen auftreten. Anschließend wird IRXJCL aufgerufen, um die tatsächliche Aktion auszuführen.

Wenn Ihre Prozessoren in einer von TSO gestarteten CARMA-Umgebung arbeiten, stimmen die Zuordnungen von SYSTSIN und SYSTSPRT mit den von CARMA verwendeten überein. Arbeiten die Prozessoren außerhalb von TSO/CARMA, werden die SYSTSIN- und SYSTSPRT-Zuordnungen von CRAXJCL erstellt. Ihre Prozessoren sind somit nicht auf den Inhalt der SYSTSIN zugeordneten Datei angewiesen.

Es wird vorausgesetzt, dass Aufrufe von IRXJCL für die Übergabe des REXX-Namens und der Startparameter das Feld PARM verwenden, wie in der Veröffentlichung *TSO/E REXX Reference* (IBM Form SA22-7790) dokumentiert. SYSTSIN kann somit sicher von CARMA verwendet werden. Alle Ausgaben, die IRXJCL an SYSTSPRT sendet, werden in das CARMA-Protokoll geschrieben.

Prozessoren, die das Ersatzmodul CRAXJCL aufrufen, sollten nicht versuchen, die DD SYSTSIN oder SYSTSPRT vor dem Aufruf von CRAXJCL zuzuordnen.

CRAXJCL erstellen

Das Ersatzmodul CRAXJCL wird im Quellenformat bereitgestellt, da Sie es anpassen müssen, um die spezifischen Zuordnungen für die Verwendung für SYSTSPRT anzugeben. Die Zuordnung für SYSTSIN sollte in der Regel zu einer Pseudodatei erfolgen.

Der Assemblerbeispiel Quellcode und der Beispieljob für Kompilierung/Bindung sind in FEK.#CUST.ASM(CRAXJCL) und FEK.#CUST.JCL(CRA#CIRX) verfügbar, sofern Sie beim Anpassen und Übergeben des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Passen Sie den Assembler Quellcode von CRAXJCL an Ihre Anforderungen an. Stützen Sie sich dabei auf die Dokumentation innerhalb des Members. Passen Sie dann die JCL CRA#CIRX an und übergeben Sie sie, um das Lademodul CRAXJCL zu erstellen. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.

Falls erforderlich, können Sie IRXJCL umbenennen. Passen Sie die CRAXJCL-Quelle an, sodass dieser neue Name für IRXJCL aufgerufen wird, und kompilieren Sie sie. Benennen Sie anschließend das Lademodul CRAXJCL in IRXJCL um. Diese Konfiguration ist möglicherweise einfacher als das Ändern aller Aufrufe für IRXJCL.

Kapitel 4. SCLM Developer Toolkit (optional)

Das SCLM Developer Toolkit stellt die Tools bereit, mit denen die Funktionalität von SCLM auch auf dem Client verfügbar gemacht werden kann. SCLM selbst ist ein hostsystembasierter Quellcodemanager, der in ISPF enthalten ist.

Das SCLM Developer Toolkit enthält ein Eclipse-basiertes Plug-in, das die Schnittstelle zu SCLM bildet. Es ermöglicht den Zugriff auf alle SCLM-Prozesse für die bisherige Codeentwicklung und stellt Unterstützung für die vollständige Java- und Java EE-Entwicklung auf der Workstation mit Synchronisation mit SCLM auf dem Großrechner bereit. Zu diesen Synchronisationsaktivitäten zählen die Erstellung, Assemblierung und Implementierung des Java EE-Codes über den Großrechner.

Voraussetzungen und Prüfliste

Für diese Anpassungstask, für die die folgenden Ressourcen und speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines SCLM-Administrators und ggf. eines Sicherheitsadministrators:

- APF und LINKLIST aktualisieren
- SCLM-Sprachumsetzer für Java EE-Unterstützung definieren
- SCLM-Typen für Java EE-Unterstützung definieren
- Sicherheitsregel für die Aktualisierung einer SCLM-VSAM durch Benutzer (optional)
- Ant installieren (optional)

Für die Verwendung von SCLM Developer Toolkit an Ihrem Standort müssen Sie die folgenden Tasks ausführen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Überprüfen Sie die Voraussetzungen und Aktualisierungen von PARMLIB und passen Sie diese an. Weitere Details enthält der Abschnitt „Voraussetzungen“ auf Seite 100.
2. Passen Sie die Konfigurationsdateien für Developer for System z an. Weitere Details enthalten die folgenden Abschnitte:
 - „Aktualisierung von ISPF.conf für SCLMDT“ auf Seite 100
 - „Aktualisierung von rsed.envvars für SCLMDT“ auf Seite 101
3. Definieren Sie die Unterstützung für die Umsetzung langer/kurzer Namen (optional). Weitere Details enthält der Abschnitt „Umsetzung langer/kurzer Namen (optional)“ auf Seite 102.
4. Optional: Installieren Sie Ant und passen Sie es an, um die Java EE-Build-Unterstützung zu verwenden. Weitere Details enthält der Abschnitt „Ant installieren und anpassen (optional)“ auf Seite 104.
5. Aktualisieren Sie SCLM, um spezifische Komponenten für SCLMDT zu definieren. Weitere Details enthält der Abschnitt „SCLM-Updates für SCLMDT“ auf Seite 105.
6. Optional: Richten Sie eine automatisierte, regelmäßige Bereinigung des SCLMDT-Arbeitsbereichs ein. Weitere Details enthält der Abschnitt „Alte Dateien aus WORKAREA und /tmp entfernen“ auf Seite 106.

Voraussetzungen

Eine Liste der erforderlichen SCLM-Wartungsmaßnahmen finden Sie in *IBM Rational Developer for System z Prerequisites* (IBM Form SC23-7659).

In dieser Veröffentlichung sind auch die für Java EE-Builds im SCLM Developer Toolkit erforderlichen Ant-Spezifikationen dokumentiert.

Achtung: Das CLM Developer Toolkit arbeitet mit dem TSO/ISPF-Client-Gateway von ISPF, sodass z/OS ab Version 1.8 erforderlich ist.

Das SCLM Developer Toolkit erfordert zusätzliche Anpassungsschritte für System-einstellungen. Lesen Sie hierzu die Beschreibung im Abschnitt „PARMLIB-Änderungen“ auf Seite 19. Zu diesen Änderungen gehören die folgenden Aktionen:

- BPXPRMxx: Erhöhen Sie die maximale Anzahl von Prozessen pro z/OS UNIX-Benutzer-ID.
- PROGxx: Berechtigen Sie SYS1.LINKLIB und die REXX-Laufzeit (REXX.V1R4M0.SEAGLPA oder REXX.V1R4M0.SEAGALT) für APF.
- PROGxx/LPALSTxx: Stellen Sie ISP.SISPLPA, ISP.SISLOAD, SYS1.LINKLIB und die REXX-Laufzeit in LINKLIST/LPALIB.

Mit SDSF oder dem TSO-Befehl **OUTPUT** ruft das SCLM Developer Toolkit den Fertigstellungsstatus von Jobs und Jobausgaben ab. Beide Methoden erfordern zusätzliche Aufmerksamkeit:

- SDSF muss separat bestellt, installiert und konfiguriert werden. SDSF erfordert außerdem die Verwendung von JES2.
- Mit den Standardeinstellungen für den TSO-Befehl **OUTPUT** kann ein Benutzer nur die Jobausgaben abrufen, die mit dieser speziellen Benutzer-ID beginnen. Zur vollständigen Nutzung der Funktion **OUTPUT** müssen Sie unter Umständen den Beispiel-TSO/E-Exit IKJEFF53 so modifizieren, dass ein Benutzer Ausgaben für Jobs abrufen kann, deren Eigner er ist, auch wenn die Ausgaben nicht mit der Benutzer-ID dieses Benutzers beginnen. Weitere Informationen zu diesem Exit enthält die Veröffentlichung *TSO/E Customization* (IBM Form SA22-7783).

Benutzer müssen die Zugriffsrechte READ, WRITE und EXECUTE für die z/OS UNIX-Verzeichnisse /tmp/ und /var/rdz/WORKAREA/ haben. Das Verzeichnis WORKAREA/ befindet sich in /var/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Aktualisierung von ISPF.conf für SCLMDT

SCLM Developer Toolkit verwendet die ISPF/SCLM-Standard-Skeletons, um sicherzustellen, dass die Skeleton-Bibliothek ISP.SISPLIB der ISPLIB-Verkettung in ISPF.conf zugeordnet wird. Die Verwendung der Datei ISP.SISPSENU ist optional.

Die Datei ISPF.conf befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Anmerkung: Änderungen werden für alle Clients wirksam, die sich nach der Aktualisierung mit dem Host verbinden.

Das folgende Beispiel zeigt die Datei `ISPF.conf`, die Sie an Ihre Systemumgebung anpassen müssen. Kommentarzeilen beginnen mit einem Stern (*). Fügen Sie Dateien zur Verkettung in derselben Zeile hinzu und trennen Sie die einzelnen Namen jeweils durch ein Komma (.). Weitere Details zur Anpassung der Datei `'ISPF.conf'` finden Sie in „`ISPF.conf` - Konfigurationsdatei für TSO/ISPF Client Gateway“ auf Seite 59.

```
* ERFORDERLICH:
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
ispmllib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISLOAD
```

```
* OPTIONAL:
*allocjob = ISP.SISPSAMP(ISPZISP2)
*ISPF_timeout = 900
```

Abbildung 30. Aktualisierung von `ISPF.conf` für `SCLMDT`

Anmerkung:

- Sie können Ihre eigenen DD-Anweisungen und Dateiverkettungen hinzufügen, um die TSO-Umgebung anzupassen und so eine TSO-Anmeldeprozedur zu imitieren. Weitere Informationen finden Sie in "TSO-Umgebung anpassen" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Wenn Sie Batch-Builds ausführen, stellen Sie sicher, dass die angepasste Version des Skeleton `FLMLIBS` vor der `ISPF/SCLM`-Skeleton-Bibliothek verknüpft ist.
`ispslib=h1q.USERSKEL,ISP.SISPSLIB`

Aktualisierung von `rsed.envvars` für `SCLMDT`

Das `SCLM Developer Toolkit` verwendet einige Anweisungen in `rsed.envvars`, um Dateigruppen und Verzeichnisse zu finden.

Die Datei `rsed.envvars` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Anmerkung: Starten Sie die gestartete `RSED`-Task erneut, damit die Änderungen wirksam werden.

Das folgende Codebeispiel zeigt die `SCLMDT`-Anweisungen in `rsed.envvars`, die Sie an Ihre Systemumgebung anpassen müssen. Weitere Details zur Anpassung der Datei `rsed.envvars` finden Sie in „`RSE`-Konfigurationsdatei `rsed.envvars`“ auf Seite 39.

```

_SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:FEK.SFEKAUTH:FEK.SFEKLOAD
#_SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/apache/Ant/apache-ant-1.7.1
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CGI_ISPHOME
CGI_DWORK=$_SCLMDT_WORK_HOME

```

Abbildung 31. Aktualisierung von *rsed.envvars* für SCLMDT

Umsetzung langer/kurzer Namen (optional)

SCLM Developer Toolkit bietet die Möglichkeit, Dateien mit langen Namen in SCLM zu speichern. Dateien mit langen Dateinamen sind Dateien mit Namen, die mehr als acht Zeichen enthalten oder in gemischter Groß-/Kleinschreibung vorliegen. Das Speichern langer Dateinamen wird durch die Verwendung einer VSAM-Datei realisiert, die die Zuordnung des langen Dateinamens zu dem in SCLM verwendeten acht Zeichen langen Membernamen enthält.

Anmerkung:

- In den Vorversionen von z/OS 1.8 wird diese Funktion über eine vorläufige Programmkorrektur zum ISPF/SCLM-Basisprodukt (zu APAR OA11426) bereitgestellt.
- Die Umsetzung langer/kurzer Namen wird auch von anderen Produkten mit Bezug zu SCLM verwendet, z. B. vom IBM SCLM Administrator Toolkit.

LSTRANS.FILE - VSAM-Datei für die Umsetzung langer/kurzer Namen - erstellen

Zum Erstellen der VSAM für die Umsetzung langer/kurzer Namen passen Sie das Beispielmember FLM02LST in der ISPF-Beispielbibliothek ISP.SISPSAMP an und übergeben es. Bei den Konfigurationsschritten in dieser Veröffentlichung wird davon ausgegangen, dass Sie die VSAM wie in der folgenden Beispielkonfigurations-JCL FEK.#CUST.LSTRANS.FILE nennen.

```

//FLM02LST JOB <Jobparameter>
//*
/* ACHTUNG: Dies ist keine JCL-Prozedur und kein vollständiger Job.
/* Vor Verwendung dieses Beispiels müssen Sie die folgenden
/* Änderungen vornehmen:
/* 1. Passen Sie die Jobparameter an Ihre Systemanforderungen an.
/* 2. Ersetzen Sie ***** durch die Platteneinheit für die VSAM.
/* 3. Ersetzen Sie alle Verweise auf FEK.#CUST.LSTRANS.FILE durch
/* Ihre Namenskonvention für die SCLM-Umsetzungs-VSAM.
/*
//CREATE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE FEK.#CUST.LSTRANS.FILE
SET MAXCC=0
DEFINE CLUSTER(NAME(FEK.#CUST.LSTRANS.FILE) -
              VOLUMES(*****)) -
              RECORDSIZE(58 2048) -
              SHAREOPTIONS(3 3) -
              CYLINDERS(1 1) -
              KEYS(8 0) -
              INDEXED) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.INDEX))

/* DEFINE ALTERNATE INDEX WITH NONUNIQUE KEYS -> ESDS */

DEFINE ALTERNATEINDEX(-
              NAME(FEK.#CUST.LSTRANS.FILE.AIX) -
              RELATE(FEK.#CUST.LSTRANS.FILE) -
              RECORDSIZE(58 2048) -
              VOLUMES(*****)) -
              CYLINDERS(1 1) -
              KEYS(50 8) -
              UPGRADE -
              NONUNIQUEKEY) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.AIX.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.AIX.INDEX))

/*
/*
//PRIME EXEC PGM=IDCAMS,COND=(0,LT)
//SYSPRINT DD SYSOUT=*
//INITREC DD *
INITREC1
/*
//SYSIN DD *
REPRO INFILE(INITREC) -
      OUTDATASET(FEK.#CUST.LSTRANS.FILE)
IF LASTCC = 4 THEN SET MAXCC=0

BLDINDEX IDS(FEK.#CUST.LSTRANS.FILE) -
          ODS(FEK.#CUST.LSTRANS.FILE.AIX)

IF LASTCC = 0 THEN -
  DEFINE PATH (NAME(FEK.#CUST.LSTRANS.FILE.PATH) -
              PATHENTRY (FEK.#CUST.LSTRANS.FILE.AIX))
/*

```

Abbildung 32. FLM02LST: Konfigurations-JCL für Umsetzung langer/kurzer Namen

Anmerkung: Benutzer benötigen für diese VSAM-Datei die Zugriffsberechtigung UPDATE. Lesen Sie hierzu die Beschreibung in "Sicherheitsaspekte" in *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

Aktualisierung von rsed.envvars für die Umsetzung langer/kurzer Namen

Entfernen Sie vor Verwendung der Umsetzung langer/kurzer Namen das Kommentarzeichen und setzen Sie die Umgebungsvariable `_SCLMDT_TRANTABLE` in `rsed.envvars`, damit der Name der VSAM für die Umsetzung langer Namen in Kurznamen übereinstimmt.

Die Datei `rsed.envvars` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Anmerkung: Starten Sie die gestartete RSED-Task erneut, damit die Änderungen wirksam werden.

Ant installieren und anpassen (optional)

Dieser Schritt ist nur erforderlich, wenn Sie in SCLM Java EE-Build-Unterstützung verwenden möchten.

Apache Ant ist ein quelloffenes Java-Build-Tool, das Sie von der Webseite <http://ant.apache.org/> herunterladen können. Ant besteht aus Textdateien und Scripts, die im ASCII-Format verteilt werden. Für die Ausführung unter z/OS UNIX ist daher eine ASCII-EBCDIC-Umsetzung erforderlich.

Führen Sie die folgenden Schritte aus, um Ant unter z/OS zu implementieren und für Developer for System z zu definieren:

- Laden Sie die neueste Ant-Datei in Binärform in das z/OS UNIX-Dateisystem herunter. Laden Sie die .zip-Version von ANT herunter.
- Öffnen Sie eine z/OS UNIX-Befehlszeilensitzung, um die Installation fortzusetzen, beispielsweise mit dem Befehl **TSO OMVS**.
- Erstellen Sie für die Ant-Installation ein Ausgangsverzeichnis mit dem Befehl **mkdir -p /home-dir** und machen Sie es mit dem Befehl **cd /home-dir** zum aktuellen Verzeichnis.
- Extrahieren Sie die Datei mit dem JAR-Extraktionsbefehl **jar -xf apache-ant-1.7.1.zip** in das aktuelle Verzeichnis. Für die Verwendung des Befehls **jar** muss in Ihrem lokalen z/OS UNIX-PATH ein Java-Verzeichnis 'bin' vorhanden sein. Ist dies nicht der Fall, verwenden Sie den Befehl mit dem vollständig qualifizierten Pfad zur Java-Position 'bin' (z. B. `/usr/lpp/java/J6.0/bin/jar -xf apache-ant-1.7.1.zip`).
- Konvertieren Sie alle Ant-Textdateien in EBCDIC. Führen Sie dazu das Beispielscript `/usr/lpp/rdz/samples/BWBTRANT` aus, das optional angepasst werden kann.

Anmerkung: Führen Sie dieses Script nur einmal aus. Durch mehrmaliges Ausführen wird Ihre Ant-Installation beschädigt.

- Suchen Sie innerhalb des Verzeichnisses ANT eine Textdatei, z. B. `apache-ant-1.7.1/README`, und öffnen Sie sie, um den Erfolg der Umsetzung zu überprüfen. Wenn die Datei lesbar ist, war die Umsetzung erfolgreich.
- Erteilen Sie mit dem Befehl **chmod -R 755 *** allen Benutzer die Zugriffsberechtigungen zum Lesen und Ausführen von Dateien im ANT-Verzeichnis.
- Definieren Sie vor der Verwendung von Ant die Umgebungsvariablen `JAVA_HOME` und `ANT_HOME` in `rsed.envvars`.

- JAVA_HOME muss auf das Java-Ausgangsverzeichnis verweisen. Beispiel:
JAVA_HOME=/usr/lpp/java/IBM/J6.0
- ANT_HOME muss auf das Ant-Ausgangsverzeichnis verweisen. Beispiel:
ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1

Beispiel:

- TSO OMVS
- mkdir -p /usr/lpp/Apache/Ant
- cd /usr/lpp/Apache/Ant
- jar -xf /u/userid/apache-ant-1.7.1
- /usr/lpp/rdz/samples/BWBTRANT
- cat ./apache-ant-1.7.1/README
- chmod -R 755 *
- oedit /etc/rsed.envvars

Testen Sie wie folgt, ob die Ant-Initialisierung erfolgreich war:

- Fügen Sie das Ant- und Java-Verzeichnis 'bin' zur Umgebungsvariablen PATH hinzu.

Beispiel:

```
export PATH=/usr/lpp/Apache/Ant/apache-ant-1.7.1/bin:$PATH
export PATH=/usr/lpp/java/J6.0/bin:$PATH
```

- Führen Sie den Befehl ant -version aus, um die Version anzuzeigen, wenn die Installation erfolgreich war.

Beispiel:

```
ant -version
```

Anmerkung: Das Festlegen der Anweisung PATH auf diese Weise ist nur für Tests, jedoch nicht für den regulären Betrieb erforderlich.

SCLM-Updates für SCLMDT

SCLM selbst erfordert für eine Zusammenarbeit mit dem SCLM Developer Toolkit einige Anpassungsschritte. Weitere Informationen zu den erforderlichen Anpassungstasks finden Sie in der Veröffentlichung *IBM Rational Developer for System z SCLM Developer Toolkit Administrator's Guide* (IBM Form SC23-9801):

- Sprachumsetzer für Java EE-Unterstützung definieren
- SCLM-Typen für Java EE-Unterstützung definieren

Der SCLM-Administrator muss verschiedene anpassbare Werte von Developer for System z kennen, die in Tabelle 13 beschrieben sind, um die Anpassungs- und Projektdefinitionstasks ausführen zu können.

Tabelle 13. Prüfliste für den SCLM-Administrator

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Beispielbibliothek von Developer for System z	<ul style="list-style-type: none"> • FEK.SFEKSAMV • SMP/E-Installation 	
Beispielverzeichnis von Developer for System z	<ul style="list-style-type: none"> • /usr/lpp/rdz/samples • SMP/E-Installation 	

Tabelle 13. Prüfliste für den SCLM-Administrator (Forts.)

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Java-Verzeichnis bin	<ul style="list-style-type: none"> • /usr/lpp/java/J6.0/bin • rsed.envvars - \$JAVA_HOME/bin 	
Ant-Verzeichnis bin	<ul style="list-style-type: none"> • /usr/lpp/Apache/Ant/apache-ant-1.7.1/bin • rsed.envvars - \$ANT_HOME/bin 	
WORKAREA-Ausgangsverzeichnis	<ul style="list-style-type: none"> • /var/rdz • rsed.envvars - \$CGI_ISPWORK 	
Ausgangsverzeichnis für SCLMDT-Projektkonfigurationen	<ul style="list-style-type: none"> • /var/rdz/sclmdt • rsed.envvars - \$_SCLMDT_CONF_HOME 	
VSAM für Umsetzung langer/kurzer Namen	<ul style="list-style-type: none"> • FEK.#CUST.LSTRANS.FILE • rsed.envvars - \$_SCLMDT_TRANTABLE 	

Alte Dateien aus WORKAREA und /tmp entfernen

SCLM Developer Toolkit nutzt die Verzeichnisse WORKAREA und /tmp gemeinsam mit dem TSO/ISPF-Client-Gateway von ISPF. Für diese Verzeichnisse könnte eine regelmäßige Bereinigung erforderlich sein. Weitere Informationen zu dieser Task finden Sie in „Bereinigung von "WORKAREA" und "/tmp" (optional)" auf Seite 139.

Kapitel 5. (Optional) Application Deployment Manager (veraltet)

Anmerkung: Application Deployment Manager (ADM) wurde als veraltet markiert. Obwohl ADM weiterhin unterstützt wird, wird diese Funktion nicht weiter entwickelt.

Developer for System z verwendet bestimmte Funktionen des Application Deployment Manager als allgemeine Deployment-Methode für verschiedene Komponenten. Die in diesem Kapitel aufgelisteten Anpassungsschritte sind erforderlich, wenn eine oder mehrere der folgenden Funktionen verwendet werden:

- Enterprise Service Tools
- BMS Screen Designer
- MFS Screen Designer
- CICSTS-Codegenerierung

Anmerkung: Die Enterprise Service Tools umfassen mehrere Tools, zu denen Service Flow Modeler (SFM) und XML Services for the Enterprise gehören.

Durch die Anpassung von Application Deployment Manager wird der CRD-Server (CICS Resource Definition) hinzugefügt, der als CICS-Anwendung unter z/OS ausgeführt wird, um die folgenden Funktionen zu unterstützen:

- CICS-Ressourcenabfragen
- CRD-Installationsanforderungen und -Deinstallationsanforderungen (CRD, CICS Resource Definition) in Umgebungen mit und ohne CICSplex SM
- Anforderungen nach einer schrittweisen Aktualisierung von Programmen und Maskengruppen
- Prüfanforderungen für Pipelines
- Export-, Import- und Aktualisierungsanforderungen für Manifeste

"CICSTS-Aspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02) enthält weitere Informationen für CICS-Administratoren zum CRD-Server.

Voraussetzungen und Prüfliste

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines CICS-Administrators, eines TCP/IP-Administrators und eines Sicherheitsadministrators:

- TCP/IP-Port für externe Kommunikation definieren
- JCL für die CICS-Region aktualisieren
- CSD für die CICS-Region aktualisieren
- Gruppe für CICS-Region definieren
- Sicherheitsregel für die Aktualisierung einer ADM-VSAM durch Administratoren erstellen
- CICSTS-Sicherheit einrichten
- CICS-Transaktionsnamen definieren (optional)
- Sicherheitsregel für die Aktualisierung einer ADM-VSAM durch Benutzer erstellen (optional)

Führen Sie die folgenden Tasks aus, um mit der Verwendung von Application Deployment Manager (ADM) an Ihrem Standort zu beginnen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Erstellen Sie das CRD-Repository. Weitere Details enthält der Abschnitt „CRD-Repository“.
2. Wählen Sie die zu verwendende CICS-Schnittstelle (RESTful oder Web-Service) aus. Die Schnittstellen können gemeinsam verwendet werden. Weitere Details enthält der Abschnitt „RESTful oder Web-Service“ auf Seite 109.
3. Führen Sie, falls erforderlich, die REST-Service-spezifischen Anpassungen durch. Weitere Details enthält der Abschnitt „CRD-Server mit der RESTful-Schnittstelle“ auf Seite 109.
 - Definieren Sie den CRD-Server für die primäre CICS-Verbindungsregion.
 - Optional: Definieren Sie den CRD-Server für nicht primäre CICS-Verbindungsregionen.
 - Optional: Passen Sie die CRD-Servertransaktions-IDs an.
4. Führen Sie, falls erforderlich, die Web-Service-spezifischen Anpassungen durch. Weitere Details enthält der Abschnitt „CRD-Server mit der Web-Service-Schnittstelle“ auf Seite 111.
 - Fügen Sie der CICS-RPL-Verkettung den Pipelinenachrichtenhandler hinzu.
 - Definieren Sie den CRD-Server für die primäre CICS-Verbindungsregion.
 - Optional: Definieren Sie den CRD-Server für nicht primäre CICS-Verbindungsregionen.
5. Optional: Erstellen Sie das Manifestrepository. Weitere Details enthält der Abschnitt „Manifest-Repository (optional)“ auf Seite 113.

CRD-Repository

Passen Sie den Job ADNVCRD an und übergeben Sie ihn, um die VSAM-Datei für das CRD-Repository anzulegen und zu initialisieren. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.

ADNVCRD ist in FEK.#CUST.JCL enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Erstellen Sie ein gesondertes Repository für jede primäre CICS-Verbindungsregion. Eine gemeinsame Nutzung des Repositories impliziert, dass alle zugehörigen CICS-Regionen dieselben im Repository gespeicherten Werte verwenden.

Anmerkung:

- Um die Unterstützung von URIMAP zu aktivieren, muss ein vorhandenes CRD-Server-Repository vergrößert werden. Die Unterstützung von URIMAP wurde dem Verwaltungsdienstprogramm in Developer for System z Version 7.6.1 hinzugefügt. Weitere Informationen finden Sie in "Migrationshinweise zum Verwaltungsdienstprogramm" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Sofern Sie keine anders lautende Benachrichtigung erhalten, können Sie Ihr aktuelles CRD-Server-Repository, das Ihre angepasste Werte enthält, für alle Releases von Developer for System z wiederverwenden.

Benutzer müssen Lesezugriff auf das CRD-Repository und CICS-Administratoren Aktualisierungszugriff haben.

CICS-Verwaltungsdienstprogramm

Mit dem von Developer for System z bereitgestellten Verwaltungsdienstprogramm können CICS-Administratoren die Standardwerte für CICS-Ressourcendefinitionen angeben. Diese Standardwerte können schreibgeschützt oder für den Anwendungsentwickler editierbar sein.

Das Verwaltungsdienstprogramm wird vom Beispieljob ADNJSPAU aufgerufen. Zur Verwendung dieses Dienstprogramms ist Aktualisierungszugriff auf das CRD-Repository erforderlich.

ADNJSPAU befindet sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Weitere Informationen hierzu finden Sie in "CICSTS-Aspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

RESTful oder Web-Service

CICS Transaction Server ab Version 4.1 hat eine HTTP-Schnittstelle, die nach RESTful-Prinzipien (Representational State Transfer) entworfen wurde. Diese RESTful-Schnittstelle ist jetzt die strategische CICSTS-Schnittstelle, die von Clientanwendungen verwendet wird. Die ältere Web-Service-Schnittstelle wurde eingefroren. Erweiterungen werden nur für die RESTful-Schnittstelle entwickelt.

Application Deployment Manager richtet sich nach dieser Absichtserklärung. Für alle Services, die ab Developer for System z Version 7.6 neu sind, ist der RESTful-CRD-Server erforderlich.

Bei Bedarf können die RESTful- und Web-Service-Schnittstellen gleichzeitig in einer CICS-Region aktiv sein. In diesem Fall sind in der Region zwei CRD-Server aktiv. Beide Server verwenden gemeinsam dasselbe CRD-Repository. CICS gibt einige Warnungen zu doppelten Definitionen aus, wenn die zweite Schnittstelle in der Region definiert wird.

CRD-Server mit der RESTful-Schnittstelle

Die Informationen in diesem Abschnitt beschreiben, wie Sie den CRD-Server definieren, der die RESTful-Schnittstelle für die Kommunikation mit dem Developer for System z-Client verwendet.

Bei Bedarf können die RESTful- und Web-Service-Schnittstellen gleichzeitig in einer CICS-Region aktiv sein. In diesem Fall sind in der Region zwei CRD-Server aktiv. Beide Server verwenden gemeinsam dasselbe CRD-Repository. CICS gibt einige Warnungen zu doppelten Definitionen aus, wenn die zweite Schnittstelle in der Region definiert wird.

Primäre CICS-Verbindungsregion

Der CRD-Server muss für die primäre Verbindungsregion definiert werden. Diese WOR (Web Owning Region) verarbeitet die Web-Service-Anforderungen von Developer for System z.

- Stellen Sie die Lademodule FEK.SFEKLOAD(ADNCRD*, ADNANAL und ADNREST) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.

- Passen Sie den Job ADNCSDRS an und übergeben Sie ihn, um die CICS-Systemdefinition (CSD) für die primäre CICS-Verbindungsregion zu aktualisieren. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.

ADNCSDRS befindet sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP (FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

- Installieren Sie die ADM-Gruppe für diese Region mit dem entsprechenden CEDA-Befehl. Beispiel:

```
CEDA INSTALL GROUP (ADNPCRGP)
```

Nicht primäre CICS-Verbindungsregionen

Der CRD-Server kann auch mit zusätzlichen, nicht primären Verbindungsregionen verwendet werden. Dabei handelt es sich in der Regel um AOR-Regionen (Application Owning Regions).

Anmerkung: Sie müssen diese Schritte nicht ausführen, wenn Ihre CICS-Ressourcendefinitionen mit BAS (Business Application Services) von CICSplex SM verwaltet werden.

- Stellen Sie das Application Deployment Manager-Lademodul FEK.SFEKLOAD (ADNCRD*) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) dieser nicht primären Verbindungsregionen. Fügen Sie dazu die Installationsdatei zur Kette hinzu, sodass angewendete Wartungen automatisch für CICS verfügbar sind.
- Passen Sie den Job ADNCS DAR an und übergeben Sie ihn, um CSD für diese nicht primären Verbindungsregionen zu aktualisieren. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.

ADNCS DAR befindet sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP (FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

- Installieren Sie die ADM-Gruppe für diese Regionen mit dem entsprechenden CEDA-Befehl. Beispiel:

```
CEDA INSTALL GROUP (ADNARRGP)
```

CRD-Servertransaktions-IDs anpassen (optional)

Developer for System z stellt mehrere Transaktionen bereit, die der CRD-Server beim Definieren und Abfragen von CICS-Ressourcen verwendet.

Tabelle 14. Standard-Transaktions-IDs des CRD-Servers

Transaktion	Beschreibung
ADMS	Für Änderungen an CICS-Ressourcen, die vom Manifestverarbeitungstool angefordert werden. Diese Transaktion ist normalerweise für CICS-Administratoren bestimmt.
ADMI	Für Anforderungen, die CICS-Ressourcen definieren, installieren oder deinstallieren.
ADMR	Für alle anderen Anforderungen, die CICS-Umgebungsinformationen oder -Ressourceninformationen abrufen.

Indem Sie die folgenden Schritte ausführen, können Sie die Transaktions-IDs ändern, damit sie mit den Standardwerten Ihres Standorts übereinstimmen:

1. Passen Sie den Job ADNTXNC an und übergeben Sie ihn, um das Lademodul ADNRCUST zu erstellen. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.
2. Stellen Sie das resultierende Lademodul ADNRCUST in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der CICS-Region, in der der CRD-Server definiert ist.
3. Passen Sie den Job ADNCSDTX an und übergeben Sie ihn, um ADNRCUST als Programm in den CICS-Regionen zu definieren, in denen der CRD-Server definiert ist. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.

Anmerkung: Der RESTful-CRD-Server versucht immer, das Lademodul ADNRCUST zu laden. Sie können daher einen kleinen Leistungsvorteil erzielen, wenn Sie das Lademodul ADNRCUST erstellen und definieren, auch wenn Sie die Transaktions-IDs nicht ändern.

CRD-Server mit der Web-Service-Schnittstelle

Die Informationen in diesem Abschnitt beschreiben, wie der CRD-Server definiert wird, der die Web-Service-Schnittstelle für die Kommunikation mit dem Developer for System z-Client verwendet.

Bei Bedarf können die RESTful- und Web-Service-Schnittstellen gleichzeitig in einer CICS-Region aktiv sein. In diesem Fall sind in der Region zwei CRD-Server aktiv. Beide Server verwenden gemeinsam dasselbe CRD-Repository. CICS gibt einige Warnungen zu doppelten Definitionen aus, wenn die zweite Schnittstelle in der Region definiert wird.

Pipelinesachrichtenhandler

Der Pipelinesachrichtenhandler (ADNTMSGH) wird für die Sicherheit verwendet. Er verarbeitet die Benutzer-ID und das Kennwort im SOAP-Header. ADNTMSGH wird von der Beispielpipelinekonfigurationsdatei referenziert und muss deshalb in die CICS-RPL-Kette gestellt werden. Weitere Informationen zum Pipelinesachrichtenhandler und die erforderliche Sicherheitskonfiguration finden Sie in "CICSTS-Aspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

Developer for System z stellt mehrere Transaktionen bereit, die der CRD-Server beim Definieren und Abfragen von CICS-Ressourcen verwendet. Die Transaktions-IDs legt ADNTMSGH je nach erforderlicher Operation fest. Für standortspezifische Anpassungen von ADNTMSGH steht COBOL-Beispielquellcode zur Verfügung.

Tabelle 15. Standard-Transaktions-IDs des CRD-Servers

Transaktion	Beschreibung
ADMS	Für Änderungen an CICS-Ressourcen, die vom Manifestverarbeitungstool angefordert werden. Diese Transaktion ist normalerweise für CICS-Administratoren bestimmt.
ADMI	Für Anforderungen, die CICS-Ressourcen definieren, installieren oder deinstallieren.
ADMR	Für alle anderen Anforderungen, die CICS-Umgebungsinformationen oder -Ressourceninformationen abrufen.

Standardmodul verwenden:

- Stellen Sie das Lademodul FEK.SFEKLOAD(ADNTMSGH) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion. Fügen Sie dazu die Installationsdatei zur Kette hinzu, sodass angewendete Wartungen automatisch für CICS verfügbar sind.

ADNTMSGH anpassen:

Die Beispielmembers ADNMSGH* befinden sich in FEK.#CUST.JCL und FEK.#CUST.COBOL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

- Passen Sie den Beispielquellcode (COBOL) FEK.#CUST.COBOL(ADNMSGHS) des Pipelinenachrichtenhandlers an die Standards Ihres Standorts an.
- Passen Sie den Job FEK.#CUST.JCL(ADNMSGHC) an und übergeben Sie ihn, um die angepasste Quelle ADNMSGHS zu kompilieren. Anweisungen zur Anpassung finden Sie in der in ADNMSGHC enthaltenen Dokumentation. Das resultierende Lademodul muss den Namen ADNTMSGH haben.
- Stellen Sie das resultierende Lademodul ADNTMSGH in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion.

Anmerkung: Stellen Sie sicher, dass das angepasste Lademodul ADNTMSGH vor allen Verweisen auf FEK.SFEKLOAD angegeben ist, da andernfalls das Standardmodul verwendet wird.

Primäre CICS-Verbindungsregion

Der CRD-Server muss für die primäre Verbindungsregion definiert werden. Diese Region verarbeitet die Serviceanforderungen von Developer for System z.

- Stellen Sie die Lademodule FEK.SFEKLOAD(ADNCRD*, ADNANAL und ADNREST) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion. Fügen Sie dazu die Installationsdatei zur Kette hinzu, sodass angewendete Wartungen automatisch für CICS verfügbar sind. Das Lademodul des Pipelinenachrichtenhandlers ADNTMSGH muss ebenfalls in der RPL-Verkettung enthalten sein, wie in „Pipelinenachrichtenhandler“ auf Seite 111 beschrieben.
- Passen Sie den Job ADNCSDWS an und übergeben Sie ihn, um die CICS-Systemdefinition (CSD) für die primäre CICS-Verbindungsregion zu aktualisieren. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Die in diesem Job verwendeten Transaktions-IDs müssen mit den vom Pipelinenachrichtenhandler verwendeten übereinstimmen, die möglicherweise angepasst wurden.

ADNCSDWS ist in FEK.#CUST.JCL enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

- Installieren Sie die ADM-Gruppe für diese Region mit dem entsprechenden CEDA-Befehl. Beispiel:
CEDA INSTALL GROUP(ADNPCRGP)

Nicht primäre CICS-Verbindungsregionen

Der CRD-Server kann auch mit zusätzlichen, nicht primären Verbindungsregionen verwendet werden. Dabei handelt es sich in der Regel um AOR-Regionen (Application Owning Regions).

Anmerkung: Diese Schritte müssen nicht ausgeführt werden, wenn die CICS-Ressourcendefinitionen mit BAS (Business Application Services) von CICSplex SM verwaltet werden.

- Stellen Sie die Application Deployment Manager-Lademodule FEK.SFEKLOAD(ADNCRD*) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) dieser nicht primären Verbindungsregionen. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.
- Passen Sie den Job ADNCSAR an und übergeben Sie ihn, um CSD für diese nicht primären Verbindungsregionen zu aktualisieren. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.
ADNCSAR befindet sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.
- Installieren Sie die ADM-Gruppe für diese Regionen mit dem entsprechenden CEDA-Befehl. Beispiel:
CEDA INSTALL GROUP(ADNARRGP)

Manifest-Repository (optional)

Developer for System z gibt Clients die Möglichkeit, Manifeste, die ausgewählte CICS-Ressourcen beschreiben, anzuzeigen und ggf. zu ändern. Änderungen können je nach den vom CICS-Administrator festgelegten Berechtigungen direkt vorgenommen oder in das Manifest-Repository zur weiteren Verarbeitung durch einen CICS-Administrator exportiert werden.

Anmerkung:

- Dieser Schritt ist nur erforderlich, wenn Manifeste aus Developer for System z exportiert werden, um sie mit dem Manifestverarbeitungstool zu verarbeiten.
- Das Manifestverarbeitungstool ist ein Plug-in für IBM CICS-Explorer.

Wenn Sie die VSAM-Datei für das Manifestrepository zuordnen und initialisieren und diese Datei für die primäre CICS-Verbindungsregion definieren wollen, passen Sie den Job ADNMFST an und übergeben ihn. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. Für jede primäre CICS-Verbindungsregion muss ein gesondertes Manifestrepository erstellt werden. Alle Benutzer benötigen Aktualisierungszugriff auf das Manifest-Repository.

ADNMFST befindet sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Kapitel 6. Hostbasierte Codeanalyse (optional)

Ähnlich wie der Developer for System z-Client unterstützt der Developer for System z-Host die Ausführung von Codeanalysetools, die als separates Produkt unter der Bezeichnung Rational Developer for System z Host Utilities bereitgestellt werden. Ein Vorteil der Ausführung der Codeanalyse auf dem Host besteht darin, dass sie in Ihre tägliche Stapelverarbeitung integriert werden kann.

Die folgenden Codeanalysetools sind auf dem Host verfügbar:

- Codeüberprüfung: Mithilfe von Regeln unterschiedlicher Fehlerkategorien prüft die Codeüberprüfung Quellcode und meldet Regelverstöße.
- Codeabdeckung: Analysiert ein aktives Programm und generiert einen Bericht zu den Zeilen, die ausgeführt werden, im Vergleich zur Gesamtzahl der ausführbaren Zeilen.

Voraussetzungen und Prüfliste

Sie benötigen keine Hilfe anderer Administratoren, um mit der Verwendung der hostbasierten Codeanalysetools an Ihrem Standort zu beginnen, jedoch müssen Sie die folgenden Tasks ausführen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Installieren Sie Rational Developer for System z Host Utilities, wie in *Program Directory for IBM Rational Developer for System z Host Utilities* (IBM Form GI13-2864) dokumentiert. Bei Verwendung der bereitgestellten Standardwerte wird das Produkt mit dem übergeordneten Qualifikationsmerkmal AKG und dem z/OS UNIX-Pfad `/usr/lpp/rdzutil` installiert.
2. Erstellen Sie anpassbare Kopien der bereitgestellten Beispiele, indem Sie den Job `AKG.SAKGSAMP(AKGSETUP)` anpassen und übergeben. Dieser Job führt die folgenden Tasks aus:
 - `AKG.#CUST.PROCLIB` erstellen und mit `SYS1.PROCLIB`-Beispiel-Memberrn füllen
 - `AKG.#CUST.JCL` erstellen und mit Beispielkonfigurations-JCL füllen

Codeüberprüfung

Die Codeüberprüfung prüft Quellcode und meldet Regelverstöße mithilfe von Regeln unterschiedlicher Fehlerkategorien. Das Tool wird mit Regelprovidern für Cobol und PL/I geliefert, jedoch können weitere Regelprovider hinzugefügt werden.

Developer for System z Host Utilities stellen eine Beispielprozedur `AKGCR` bereit, die das Aufrufen von Codeüberprüfungsservices im Batch-Modus vereinfacht. Die Prozedur `AKGCR` befindet sich in `AKG.#CUST.PROCLIB`, sofern Sie bei der Anpassung und Übergabe des Jobs `AKG.SAKGSAMP(AKGSETUP)` keine andere Position angegeben haben.

Passen Sie die Beispielprozedur `AKG.#CUST.PROCLIB(AKGCR)` wie innerhalb des Members beschrieben an und kopieren Sie sie in `SYS1.PROCLIB`.

Wenn die Prozedur `AKGCR` nicht in eine Systemprozedurbibliothek kopiert werden kann, fordern Sie die Benutzer von Developer for System z auf, ihrem Aufrufjob eine JCLLIB-Karte nach der Jobkarte hinzuzufügen.

```
//MYJOB    JOB <Jobparameters>
//PROCS    JCLLIB ORDER=(AKG.#CUST.PROCLIB)
```

Verarbeitung der Codeüberprüfung ändern

Die Codeüberprüfung von Developer for System z ermöglicht die Einbeziehung von Code von Drittherstellern in den Prüfprozess. Sie können zum Beispiel einen Regelprovider bereitstellen, um C/C++-Code zu analysieren, oder den Cobol-Regelprovider erweitern, sodass sitespezifische Codierungskonventionen erkannt werden.

Die hostbasierte Codeüberprüfung ist ebenso ein Eclipse-Prozess, wie der Developer for System z-Client. Daher können die Erweiterungen, die von Ihrem Entwicklungsunterstützungsteam für die Codeüberprüfung auf dem Client vorgenommen werden, auf dem Host wiederverwendet werden.

Die Erweiterungen bestehen aus Eclipse-Plug-ins oder Eclipse-Funktionen. Zur Aktivierung dieser Erweiterungen müssen Sie sie für den vorhandenen Code verfügbar machen, wie im Konfigurationsjob AKGCRADD dokumentiert. Der Job AKGCRADD befindet sich in AKG.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs AKG.SAKGSAMP(AKGSETUP) keine andere Position angegeben haben.

Codeabdeckung

Die Codeabdeckung analysiert ein aktives Programm und generiert einen Bericht zu den Zeilen, die ausgeführt werden, im Vergleich zur Gesamtzahl der ausführbaren Zeilen. Beachten Sie, dass die Codeabdeckung eine TCP/IP-Verbindung zur Komponente 'Integrated Debugger' über einen ephemeren Port herstellt, um die erforderlichen Daten zu erfassen. Optional kann IBM Debug Tool for z/OS anstelle von Integrated Debugger verwendet werden.

Developer for System z Host Utilities stellen eine Beispielprozedur AKGCC bereit, die das Aufrufen von Codeabdeckungsservices im Batch-Modus vereinfacht. Die Prozedur AKGCC befindet sich in AKG.#CUST.PROCLIB, sofern Sie bei der Anpassung und Übergabe des Jobs AKG.SAKGSAMP(AKGSETUP) keine andere Position angegeben haben.

Passen Sie die Beispielprozedur AKG.#CUST.PROCLIB(AKGCC) wie innerhalb des Members beschrieben an und kopieren Sie sie in SYS1.PROCLIB.

Wenn die Prozedur AKGCC nicht in eine Systemprozedurbibliothek kopiert werden kann, fordern Sie die Benutzer von Developer for System z auf, ihrem Aufrufjob eine JCLLIB-Karte nach der Jobkarte hinzuzufügen.

```
//MYJOB    JOB <jobparameter>
//PROCS    JCLLIB ORDER=(AKG.#CUST.PROCLIB)
```

Ausgabe der Codeabdeckung

Die Ausgabe der Codeabdeckung ist für den Import in einen Developer for System z-Client vorgesehen und wird daher in eine z/OS UNIX-Datei geschrieben. Die Codeabdeckung kann darüber hinaus auch die Ergebnisse einer vorherigen Ausführung verwenden und diese mit den Ergebnissen der aktuellen Ausführung kombinieren. Dadurch wird ein einzelner Bericht generiert, der mehrere Codepfade abdeckt.

Infolgedessen versucht Developer for System z nicht, die Ausgabe einer Ausführung der Codeabdeckung zu entfernen, sodass die Ausgaben im Verlauf der Zeit kumulieren.

z/OS UNIX stellt ein Shell-Script mit dem Namen `skulker` bereit, das Dateien auf der Basis des Verzeichnisses, in dem sie sich befinden, und ihres Alters löscht. In Kombination mit dem z/OS UNIX-Dämon `cron`, der Befehle zu angegebenen Zeitpunkten (Datum/Uhrzeit) ausführt, können Sie ein automatisiertes Tool einrichten, das bestimmte Zielverzeichnisse in regelmäßigen Abständen bereinigt. Weitere Informationen zum Script `skulker` und zum Dämon `cron` enthält die Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802).

Kapitel 7. Weitere Anpassungstasks (optional)

In diesem Abschnitt werden verschiedene optionale Anpassungstasks kombiniert. Zur Konfiguration des erforderlichen Service führen Sie die Anweisungen im jeweiligen Abschnitt aus.

Anpassungen in Konfigurationsdateien von Developer for System z:

- „pushtoclient.properties - hostbasierte Clientsteuerung (optional)“
- „ssl.properties - RSE-SSL-Verschlüsselung (optional)“ auf Seite 122
- „rsecomm.properties - RSE-Trace (optional)“ auf Seite 125
- „include.conf“ (optional): Erzwungene Includes für C/C++-Content-Assist“ auf Seite 127

Anpassungen an oder für andere Produkte in Bezug auf Developer for System z:

- „(Optional) z/OS UNIX-Unterprojekte“ auf Seite 128
- „Unterstützung für Include-Vorprozessor (optional)“ auf Seite 129
- „xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)“ auf Seite 130
- „Enterprise Service Tools-Unterstützung (optional)“ auf Seite 131
- „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 131
- „IRZ-Diagnosenachrichten für generierten Code (optional)“ auf Seite 132
- „Integrated Debugger (optional)“ auf Seite 133
- „Unterstützung für Tools zur Problembestimmung (optional)“ auf Seite 136
- „Unterstützung für DB2- und IMS-Debug (optional)“ auf Seite 137
- „Unterstützung für File Manager (optional)“ auf Seite 138
- „Bereinigung von "WORKAREA" und "/tmp" (optional)“ auf Seite 139

pushtoclient.properties - hostbasierte Clientsteuerung (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks für eine Basisinstallation erforderlich.

Wenn Sie die Gruppenunterstützung aktivieren, benötigen Sie für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, die Unterstützung eines Sicherheitsadministrators oder eines LDAP-Administrators:

- Sicherheitsregel, die den Zugriff von Benutzern auf FEK.PTC.*-Profile erlaubt
- Oder Sie definieren die Benutzerzugehörigkeit von LDAP-Gruppen des Typs FEK.PTC.*

Clients von Developer for System z ab Version 8.0.1 können beim Herstellen einer Verbindung Clientkonfigurationsdateien und Informationen zur Produktaktualisierung vom Hostsystem abrufen. Auf diese Weise wird sichergestellt, dass alle Clients gemeinsame Einstellungen verwenden und auf dem aktuellen Stand sind.

z/OS-Projekte können einzeln in der Perspektive für z/OS-Projekte auf dem Client definiert werden. Sie können jedoch auch zentral auf dem Hostsystem definiert und anschließend benutzerabhängig an den Client weitergegeben werden. Solche hostbasierten Projekte sind in Aussehen und Funktionsweise mit den auf dem Client definierten Projekten identisch. Die Struktur, die Member und die Eigenschaft-

ten dieser Projekte können jedoch nicht vom Client geändert werden und sind nur bei bestehender Verbindung zum Hostsystem zugänglich.

In `pushtoclient.properties` wird für den Client angegeben, ob diese Funktionen aktiviert sind und wo die zugehörigen Daten gespeichert sind. Die Daten werden von einem Developer for System z-Clientadministrator oder einem Entwicklungsprojektmanager verwaltet.

Die Datei `pushtoclient.properties` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. Starten Sie die gestartete RSED-Task erneut, damit die Änderungen wirksam werden.

Ab Version 8.0.3 kann der Client-Administrator mehrere Clientkonfigurationsgruppen und mehrere Szenarios für die Clientaktualisierung erstellen, die dem Bedarf der verschiedenen Entwicklergruppen angepasst sind. Diese mehreren Gruppen und Szenarios können verwendet werden, um Benutzern auf der Basis von Kriterien wie der Zugehörigkeit zu einer LDAP-Gruppe oder der Zulassung zu einem Sicherheitsprofil eine angepasste Konfiguration zur Verfügung zu stellen. Weitere Informationen zur Unterstützung mehrerer Gruppen finden Sie in „Hinweise zu Push-to-Client“ in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

Das folgende Codebeispiel zeigt die Datei `pushtoclient.properties`, die Sie an Ihre Systemumgebung anpassen müssen. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare in derselben Zeile sind nicht zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
#
# host-based client control
#
config.enabled=false
product.enabled=false
reject.config.updates=false
reject.product.updates=false
accept.product.license=false
primary.system=false
pushtoclient.folder=/var/rdz/pushtoclient
default.store=com.ibm.ftt.configurations.USS
file.permission=RWX.RWX.RX
```

Abbildung 33. *pushtoclient.properties*: Konfigurationsdatei für hostbasierte Clientsteuerung

config.enabled

Gibt an, ob die hostbasierte Clientsteuerung für Konfigurationsdateien verwendet wird. Die Standardeinstellung ist `false`. Gültige Werte sind `true`, `false`, LDAP oder SAF. Die Bedeutung dieser Werte finden Sie in Tabelle 16 auf Seite 122.

product.enabled

Gibt an, ob die hostbasierte Clientsteuerung für Produktaktualisierungen verwendet wird. Die Standardeinstellung ist `false`. Gültige Werte sind `true`, `false`, LDAP oder SAF. Die Bedeutung dieser Werte finden Sie in Tabelle 16 auf Seite 122.

reject.config.updates

Gibt an, ob der Benutzer Konfigurationsaktualisierungen ablehnen kann, die durch Push an den Client übertragen werden. Die Standardeinstellung

ist false. Gültige Werte sind true, false, LDAP oder SAF. Die Bedeutung dieser Werte finden Sie in Tabelle 16 auf Seite 122.

reject.product.updates

Gibt an, ob der Benutzer Produktaktualisierungen ablehnen kann, die durch Push an den Client übertragen werden. Die Standardeinstellung ist false. Gültige Werte sind true, false, LDAP oder SAF. Die Bedeutung dieser Werte finden Sie in Tabelle 16 auf Seite 122.

accept.product.license

Zeigt an, ob die Produktlizenz automatisch während Aktualisierungen akzeptiert wird, die von Push-to-Client eingeleitet werden. Ist sie aktiviert, fordert IBM Installation Manager während der Clientaktualisierung nicht zum Akzeptieren der Lizenz auf. Die Standardeinstellung ist false. Die einzigen gültigen Werte sind true und false.

primary.system

Die hostbasierte Clientsteuerung unterstützt das Speichern von systemspezifischen Daten für jedes System, während gemeinsame Daten auf einem einzelnen System verwaltet werden, um den Verwaltungsaufwand zu reduzieren. Diese Anweisung gibt an, ob auf diesem System globale, nicht systemspezifische Clientdefinitionen gespeichert werden. Die Standardeinstellung ist false. Die einzigen gültigen Werte sind true und false.

Anmerkung: Stellen Sie sicher, dass Sie nur ein einziges System als primäres System definiert haben. Administratoren von Developer for System z-Clients können globale Konfigurationsdaten nur exportieren, wenn das Zielsystem ein primäres System ist. Developer for System z-Clients zeigen möglicherweise ein fehlerhaftes Verhalten, wenn sie Verbindungen zu mehreren primären Systemen mit Konfigurationen herstellen, die nicht synchron sind.

pushtoclient.folder

Das Basisverzeichnis für die Definitionen der hostbasierten Clientsteuerung. Die Standardeinstellung ist /var/rdz/pushtoclient.

default.store

Die hostbasierte Clientsteuerung unterstützt verschiedene Methoden zum Speichern der Daten, die mit Push an den Client übertragen werden. Mit dieser Anweisung wird der Treiber oder Speicher angegeben, der für den Zugriff auf die Daten verwendet wird. Die Standardeinstellung ist com.ibm.ftt.configurations.USS, die Daten unterstützt, die in unstrukturierten z/OS UNIX-Dateien gespeichert werden.

Developer for System z stellt nur den Speicher com.ibm.ftt.configurations.USS bereit. Ein Speicher eines Drittanbieters ist erforderlich, wenn sich die Daten an einer anderen Position befinden.

file.permission

Der Speicher com.ibm.ftt.configurations.USS verwendet file.permission, um die erforderlichen Zugriffsberechtigungen für Dateien zu ermitteln, die durch den Speicher erstellt werden. Die Standardeinstellung ist RWX.R-WX.RX, das heißt, der Eigentümer und die Standardgruppe des Eigentümers haben Lese- und Schreibzugriff auf die Verzeichnisstruktur und die darin gespeicherten Dateien. Alle anderen Personen haben nur Lesezugriff auf die Verzeichnisstruktur und die darin enthaltenen Dateien.

Nach den UNIX-Standards können Berechtigungen für drei Benutzertypen festgelegt werden: Eigentümer, Gruppe und Andere. Die Felder in der Maske file.permission stimmen mit dieser Reihenfolge überein und wer-

den durch einen Punkt (.) getrennt. Alle Felder können leer sein oder R, W, RW, X, RX, WX oder RWX als Wert enthalten (mit R = Lesen, W = Schreiben, X = Ausführen oder Verzeichnisinhalt auflisten).

Tabelle 16. Unterstützung der Push-to-Client-Gruppe

Schlüsselwert	Ist die zugehörige Push-to-Client-Funktion aktiviert?
False	Nein, inaktiviert
True	Ja, für alle aktiviert
LDAP	Ja, aber die Verfügbarkeit wird durch die Zugehörigkeit von LDAP-Gruppen gesteuert
SAF	Ja, aber die Verfügbarkeit wird durch die Zulassung zu Sicherheitsprofilen gesteuert

Anmerkung:

- Damit die hostbasierte Clientsteuerung aktiviert wird, muss eine Datei `keymapping.xml` im Verzeichnis `/var/rdz/pushtoclient` vorhanden sein. Diese Datei wird von einem Clientadministrator von Developer for System z erstellt und verwaltet.
- Weitere Informationen zu hostbasierten Projekten, zur hostbasierten Clientkonfiguration und zur Upgradesteuerung finden Sie in "Hinweise zu Push-to-Client" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).
- Wenn eine Datei erstellt wird, verwendet z/OS UNIX standardmäßig die effektive UID (Benutzer-ID) des erstellenden Threads und die GID (Gruppen-ID) des Eigentümerverzeichnis, nicht die effektive GID des erstellenden Threads. Weitere Informationen zum Ändern dieses Verhaltens oder zum Anpassen Ihrer Konfiguration der hostbasierten Clientsteuerung, damit die erforderliche GID-Zuweisung vorgenommen wird, finden Sie in "z/OS UNIX-Verzeichnisstruktur" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

ssl.properties - RSE-SSL-Verschlüsselung (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines Sicherheitsadministrators:

- LINKLIST aktualisieren
 - Sicherheitsregel für das Hinzufügen programmgesteuerter Dateien
 - Sicherheitsregel für das Hinzufügen von Zertifikaten für SSL (optional)
-

Die externe Client-Host-Kommunikation kann mit SSL (Secure Socket Layer) verschlüsselt werden. Dieses Feature ist standardmäßig inaktiviert und wird von den Einstellungen in `ssl.properties` gesteuert.

Anmerkung: Die Clientauthentifizierung mit einem X.509-Zertifikat erfordert die Verwendung einer mit SSL verschlüsselten Kommunikation.

Die Datei `ssl.properties` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Der Client kommuniziert während des Verbindungsaufbaus mit dem RSE-Dämon und während der aktuellen Sitzung mit dem RSE-Server. Wenn SSL aktiviert ist, sind beide Datenströme verschlüsselt.

Aufgrund unterschiedlicher Architektur unterstützen der RSE-Dämon und der RSE-Server verschiedene Mechanismen zum Speichern von Zertifikaten. Dies impliziert, dass für den RSE-Dämon sowie den RSE-Server SSL-Definitionen erforderlich sind. Ein gemeinsam genutztes Zertifikat kann verwendet werden, wenn der RSE-Dämon und der RSE-Server dieselbe Zertifikatsverwaltungsmethode verwenden.

Tabelle 17. Mechanismen für den SSL-Zertifikatsspeicher

Zertifikatsspeicher	Erstellt und verwaltet von	RSE-Dämon	RSE-Server
Schlüsseldatei	SAF-kompatibles Sicherheitsprodukt	Unterstützt	Unterstützt
Schlüsseldatenbank	Schlüsseldatei 'gskkyman' von z/OS UNIX	Unterstützt	/
Keystore	Java-keytool	/	Unterstützt

Anmerkung:

- Für die Verwaltung von Zertifikaten sind SAF-kompatible Schlüsseldateien die bevorzugte Methode.
- SAF-kompatible Schlüsseldateien speichern den privaten Schlüssel des Zertifikats in der Sicherheitsdatenbank oder mithilfe von ICSF, der Schnittstelle zur Verschlüsselungshardware in System z. Der Zugriff auf ICSF wird durch Profile in der Sicherheitsklasse CSFSERV geschützt.

Zum Verwalten von SSL verwendet der RSE-Dämon System SSL-Funktionen. Dies impliziert, dass SYS1.SIEALNKE von der Sicherheitssoftware programmgesteuert und für RSE bei Verwendung von LINKLIST oder der STEPLIB-Anweisung in rsed.envvars verfügbar sein muss.

Das folgende Codebeispiel zeigt die Datei `ssl.properties`, die Sie an Ihre Systemumgebung anpassen müssen. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert enthalten. Kommentare sind in derselben Zeile nicht zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
# ssl.properties – SSL-Konfigurationsdatei
enable_ssl=false

# Dämonmerkmale

#daemon_keydb_file=
#daemon_keydb_password=
#daemon_key_label=

# Servermerkmale

#server_keystore_file=
#server_keystore_password=
#server_keystore_label=
#server_keystore_type=JCERACFKS
```

Abbildung 34. SSL-Konfigurationsdatei *ssl.properties*

Die Dämon- und Servereigenschaften müssen nur festgelegt werden, wenn Sie SSL aktivieren. Weitere Informationen zur SSL-Konfiguration finden Sie in "SSL- und X.509-Authentifizierung konfigurieren" in der *Developer for System z - Hostkonfigurationsreferenz*.

enable_ssl

Aktivieren oder Inaktivieren der SSL-Kommunikation. Die Standardeinstellung ist `false`. Die einzigen gültigen Optionen sind `true` und `false`.

daemon_keydb_file

Name der Schlüsseldatei von RACF oder eines ähnlichen Sicherheitsprodukts. Wenn Sie anstelle einer Schlüsseldatei **gskkyman** verwendet haben, um eine Schlüsseldatenbank zu erstellen, geben Sie den Namen der Schlüsseldatenbank an. Wenn SSL aktiviert ist, entfernen Sie das Kommentarzeichen und passen diese Anweisung an.

daemon_keydb_password

Lassen Sie die Anweisung auf Kommentar gesetzt bzw. geben Sie keinen Wert an, wenn Sie eine Schlüsseldatei verwenden. Geben Sie andernfalls das Kennwort für die Schlüsseldatenbank an. Wenn SSL aktiviert ist und Sie eine **gskkyman**-Schlüsseldatenbank verwenden, entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an.

daemon_key_label

Die in der Schlüsseldatei oder Schlüsseldatenbank verwendete Bezeichnung des Zertifikats, sofern es sich nicht um das Standardzertifikat handelt. Wenn die Standardbezeichnung verwendet wird, muss diese Angabe auf Kommentar gesetzt werden. Wenn SSL aktiviert ist und Sie nicht das Standardsicherheitszertifikat verwenden, entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an. Bei Schlüsselbezeichnungen wird die Groß-/Kleinschreibung unterschieden.

server_keystore_file

Name des vom Java-Befehl **keytool** erstellten Keystore oder der Name der RACF-Schlüsseldatei oder eines ähnlichen Sicherheitsprodukts, wenn `server_keystore_type=JCERACFKS` angegeben ist. Wenn SSL aktiviert ist, entfernen Sie das Kommentarzeichen und passen diese Anweisung an.

server_keystore_password

Lassen Sie die Anweisung auf Kommentar gesetzt bzw. geben Sie keinen Wert an, wenn Sie eine Schlüsseldatei verwenden. Geben Sie andernfalls das Kennwort für den Keystore an. Wenn SSL aktiviert ist und Sie einen **keytool**-Keystore verwenden, entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an.

server_keystore_label

Die in der Schlüsseldatei oder dem Keystore verwendete Bezeichnung des Zertifikats. Standardmäßig wird das erste gültige Zertifikat ermittelt. Wenn SSL aktiviert ist und Sie nicht das Standardsicherheitszertifikat verwenden, entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an. Bei Schlüsselbezeichnungen wird die Groß-/Kleinschreibung unterschieden.

server_keystore_type

Keystoretyp. Die Standardeinstellung ist JKS. Gültige Werte:

Tabelle 18. Gültige Keystoretypen

Schlüsselwort	Keystoretyp
JKS	Java-Keystore
JCERACFKS	SAF-kompatible Schlüsseldatei, wobei der private Schlüssel des Zertifikats in der Sicherheitsdatenbank gespeichert ist
JCECCARACFKS	SAF-kompatible Schlüsseldatei, wobei der private Schlüssel des Zertifikats mithilfe von ICSF gespeichert wird, der Schnittstelle zur Verschlüsselungshardware in System z

Anmerkung: Zum Zeitpunkt der Veröffentlichung ist eine Aktualisierung der Datei `/usr/lpp/java/J6.0/lib/security/java.security` von IBM z/OS Java erforderlich, damit JCECCARACFKS unterstützt wird. Die folgende Zeile muss hinzugefügt werden:

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
```

Die Ergebnisdatei sieht wie folgt aus:

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
security.provider.5=com.ibm.security.cert.IBMCertPath
security.provider.6=com.ibm.security.sasl.IBMSASL
```

rsecomm.properties - RSE-Trace (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Developer for System z unterstützt zur Problemlösung verschiedene Trace-Stufen für den internen Programmablauf. RSE und einige von RSE aufgerufene Services ermitteln anhand der Einstellungen in `rsecomm.properties` den erforderlichen anfänglichen Detaillierungsgrad der Ausgabeprotokolle.

Achtung: Änderungen an diesen Einstellungen können zu Leistungseinbußen führen und sollten nur auf Anweisung des IBM Support Center vorgenommen werden.

Die Datei `rsecomm.properties` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position ange-

geben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

Das folgende Codebeispiel zeigt die Datei `rsecomm.properties`, die Sie an Ihre Trace-Anforderungen anpassen können. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert enthalten. Kommentare sind in derselben Zeile nicht zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
# server.version - NICHT ÄNDERN!  
server.version=5.0.0  
  
# Protokollstufe  
# 0 - Fehlernachrichten protokollieren  
# 1 - Fehlernachrichten und Warnungen protokollieren  
# 2 - Fehlernachrichten, Warnungen und Informationen protokollieren  
debug_level=1#USER=userid  
#USER=(userid,userid,...)
```

Abbildung 35. `rsecomm.properties` – Konfigurationsdatei für Protokollierung

server.version

Version des Protokollierungsservers. Die Standardeinstellung ist 5.0.0. Modifizieren Sie diese Einstellung nicht.

debug_level

Detaillierungsgrad für Ausgabeprotokolle. Die Standardeinstellung ist 1, die angibt, dass Fehlernachrichten und Warnungen protokolliert werden. Die Anweisung `debug_level` steuert den Detaillierungsgrad mehrerer Services und dadurch mehrerer Ausgabedateien. Die Erhöhung des Detaillierungsgrades kann zu Leistungseinbußen führen und sollte nur auf Anweisung des IBM Support Center erfolgen. Weitere Informationen zu den Protokollen, die von dieser Anweisung gesteuert werden, finden Sie in "Traceerstellung für RSE" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

Gültige Werte:

0	Nur Fehlernachrichten protokollieren
1	Fehlernachrichten und Warnungen protokollieren
2	Fehlernachrichten, Warnungen und Informationsnachrichten protokollieren

Anmerkung: Mit den Bedienerbefehlen **modify rsecommlog**, **modify rseserverlog** und **modify rsedaemonlog** kann `debug_level` dynamisch für bestimmte Protokolldateien geändert werden, wie in Kapitel 11, „Bedienerbefehle“, auf Seite 193 beschrieben.

USER Legen Sie beim Serverstart die Debugstufe 2 (Fehlernachrichten, Warnungen und Informationsnachrichten protokollieren) für die angegebenen Benutzer-IDs fest. Als Debugstufe für alle anderen Benutzer wird die in der Anweisung `debug_level` angegebene Standardeinstellung verwendet. Die Anweisung `USER` ändert die Tracedetailstufe für RSE-Server (`rsecomm.log`) und die MVS-Dateiservices (`lock.log` und `ffs*.log`). Sie ist äquivalent zum Bedienerbefehl **modify trace user**.

"include.conf" (optional): Erzwungene Includes für C/C++-Content-Assist

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Content-Assist für C/C++ kann die Definitionen in `include.conf` verwenden, um erzwungene Includes für bestimmte Dateien oder Member vorzunehmen. Ein erzwungenes Include besteht aus einer Datei oder einem Verzeichnis, einem Dataset oder einem Dateimember, für das ein Parsing durchgeführt wird, wenn eine Content-Assist-Operation ausgeführt wird, unabhängig davon, ob diese Datei oder das Member mithilfe einer Vorprozessoranweisung in den Quellcode eingeschlossen wurde.

Bevor die Datei verwendet werden kann, muss in `rsed.envvars` mithilfe der Variablen `include.c` oder `include.cpp` auf die Datei verwiesen werden. Dieser Verweis in `rsed.envvars` bedeutet, dass Sie für die Verwendung durch C und C++ eine andere Datei angeben können. Die Variablen in `rsed.envvars` sind standardmäßig inaktiviert.

Die Beispieldatei `include.conf` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

Definitionen müssen in Spalte 1 beginnen. Wenn eine US-Codepage verwendet wird, beginnen Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur den Namen eines Verzeichnisses, einer Datei, eines Datasets oder eines Members aufweisen. Kommentare sind nicht in derselben Zeile zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```

# To include the stdio.h file from the /usr/include directory, input:
# /usr/include/stdio.h
#
# To include all files of the /usr/include directory and all of it's
# sub-directories, input:
# /usr/include
#
# Uncomment and customize variable FILETYPES to limit the z/OS UNIX
# wildcard include to selected (case sensitive) file types:
# The file types are specified in a comma-delimited list (no blanks)
# FILETYPES=H,h,hpp,C,c,cpp,cxx

# To include all members of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H
#
# To include the STDIOSTR member of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H(STDIOSTR)
# The sample list contains some commonly used C standard library files
/usr/include/assert.h
/usr/include/ctype.h
/usr/include/errno.h
/usr/include/float.h
/usr/include/limits.h
/usr/include/locale.h
/usr/include/math.h
/usr/include/setjmp.h
/usr/include/signal.h
/usr/include/stdarg.h
/usr/include/stddef.h
/usr/include/stdio.h
/usr/include/stdlib.h
/usr/include/string.h
/usr/include/time.h

```

Abbildung 36. "include.conf": Erzwungene Includes für C/C++-Content-Assist

(Optional) z/OS UNIX-Unterprojekte

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

REXEC (Remote Execution) ist ein TCP/IP-Service, mit dem Clients einen Befehl auf dem Hostsystem ausführen können. SSH (Secure Shell) ist ein ähnlicher Service, bei dem jedoch die gesamte Kommunikation mit SSL (Secure Sockets Layer) verschlüsselt wird. Developer for System z nutzt beide Services für ferne (hostbasierte) Aktionen in z/OS UNIX-Unterprojekten.

Anmerkung:

- Developer for System z verwendet die z/OS UNIX-Version von REXEC und nicht die TSO-Version.
- Wenn REXEC/SSH nicht für die Verwendung des Standardports konfiguriert ist, muss die Clientkomponente von Developer for System z den korrekten Port für z/OS UNIX-Unterprojekte definieren. Diese Konfiguration kann über die Benutzervorgabenseite eingerichtet werden, die durch Auswahl der Optionen **Fenster > Vorgaben > z/OS-Lösungen > USS-Unterprojekte > Optionen für ferne Aktionen** geöffnet wird. Informationen zum verwendeten Port finden Sie in „REXEC- oder SSH-Konfiguration“ auf Seite 129.

REXEC- oder SSH-Konfiguration

REXEC und SSH basieren auf Services, die von INETD (Internet Daemon) bereitgestellt werden. Dabei handelt es sich um einen weiteren TCP/IP-Service. Im Handbuch *Communications Server IP Configuration Guide* (IBM Form SC31-8775) sind die erforderlichen Konfigurationsschritte für INETD, REXEC und SSH beschrieben. Weitere Details und alternative Konfigurationsmethoden finden Sie im White Paper *Using INETD, REXEC and SSH with Rational Developer for System z* (IBM Form SC14-7301), das in der Bibliothek von Developer for System z unter <http://www-01.ibm.com/support/docview.wss?uid=swg27038517> verfügbar ist.

Ein von REXEC verwendeter allgemeiner Port ist 512. Zur Überprüfung des Ports, der verwendet wird, prüfen Sie `/etc/inetd.conf` und `/etc/services`.

- Suchen Sie in der Datei `/etc/inetd.conf` den Servicenamen (erstes Wort; in diesem Beispiel `exec`) des Servers `rexecd` (siebtes Wort).

```
exec stream tcp nowait OMVSKERN /usr/sbin/orexecd rexecd -LV
```

- Suchen Sie in der Datei `/etc/services` den Port (zweites Wort; in diesem Beispiel 512), der diesem Servicenamen (erstes Wort) zugeordnet ist.

```
exec      512/tcp      #REXEC      Befehlsserver
```

Dasselbe Prinzip gilt für SSH. Der allgemeine Port ist 22 und der Servername ist `sshd`.

Unterstützung für Include-Vorprozessor (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Developer for System z unterstützt die Interpretation und Erweiterung von Include-Anweisungen von COBOL und PL/I, einschließlich ausgewählter Include-Anweisungen von Fremdanbietern. Developer for System z stellt auch die Beispiel-REXX-Exec FEKRNPLI bereit, die von dem Developer for System z-Client aufgerufen werden kann, um die PL/I-Quelle durch Aufrufen des PL/I-Compilers zu erweitern.

FEKRNPLI befindet sich in `FEK.#CUST.CNTL`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in Kapitel 2, „Basisanpassung“, auf Seite 17.

Passen Sie die Beispiel-Exec `FEK.#CUST.CNTL(FEKRNPLI)` an, wie im Member beschrieben. Sie müssen die folgenden Informationen angeben:

- `compiler_hlq`: Das übergeordnete Qualifikationsmerkmal für den PL/I-Compiler

Der Developer for System z-Client verwendet zum Ausführen der Exec den TSO Commands Service. Dies bedeutet, dass der Benutzer die genaue Position der Exec nicht kennen muss, wenn die Exec FEKRNPLI in die Verknüpfung `SYSPROC` oder `SYSEXEC` für den TSO Commands-Service platziert wird. Der Benutzer muss nur den Namen kennen. TSO Commands Service verwendet standardmäßig das ISPF-Client-Gateway, um eine TSO-Umgebung zu erstellen, aber APPC wird ebenfalls unterstützt, wie im White Paper *Using APPC to provide TSO command services* (SC14-7291) dokumentiert. Bei Verwendung des ISPF-Client-Gateways wird die Verknüpfung `SYSPROC` bzw. `SYSEXEC` in `ISPF.conf` definiert. Weitere Details zur Anpassung dieser Datei finden Sie in „ISPF.conf - Konfigurationsdatei für TSO/ISPF Client Gateway“ auf Seite 59.

xUnit-Unterstützung für Enterprise COBOL und PL/I (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind aber die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich:

- LINKLIST aktualisieren
-

Frameworks, die Entwickler beim Schreiben von Code für die Ausführung von wiederholt anwendbaren, selbstprüfenden Komponententests unterstützen, werden zusammengefasst mit dem Namen xUnit bezeichnet. Developer for System z stellt ein solches Framework für Komponententests von Enterprise COBOL- und PL/I-Code bereit, das zUnit heißt.

Zur Verwendung des Frameworks zUnit müssen Entwickler über Zugriff auf die AZU*- und IAZU*-Ladmodule in der Ladebibliothek FEK.SFEKLOAD verfügen (über STEPLIB oder LINKLIST). Der zUnit-Test Runner AZUTSTRN wiederum benötigt Zugriff auf verschiedene Systembibliotheken (über STEPLIB oder LINKLIST):

- CEE.SCEERUN und CEE.SCEERUN2 (LE-Laufzeit)
- SYS1.CSSLIB (aufrufbare Systemservices)
- SYS1.SIXMLOD1 (XML-Toolkit)

Der zUnit-Test Runner benötigt auch Zugriff auf eine Ladebibliothek, die die verschiedenen Testfälle enthält. Diese Bibliothek ist wahrscheinlich für einen Entwickler spezifisch.

Der zUnit-Test Runner AZUTSTRN kann vom Developer for System z-Client im Batch-Modus, über die TSO-Befehlszeile und über die z/OS UNIX-Befehlszeile aufgerufen werden.

- Developer for System z stellt die Beispielprozedur AZUZUNIT bereit, um das Aufrufen des zUnit-Test Runners im Batch-Modus zu vereinfachen. AZUZUNIT befindet sich in FEK.#CUST.PROCLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in Kapitel 2, „Basisanpassung“, auf Seite 17.

Passen Sie die Beispielprozedur FEK.#CUST.PROCLIB(AZUZUNIT) wie innerhalb des Members beschrieben an und kopieren Sie sie in SYS1.PROCLIB.

Der Name der Prozedur und die Namen der Prozedurschritte stimmen mit den Standardmerkmalen des Developer for System z-Clients überein. Wenn der Name einer Prozedur oder eines Prozedurschritts geändert wird, muss die entsprechende Eigenschaftendatei auf allen Clients aktualisiert werden. Namen von Prozeduren oder Prozedurschritten sollten nicht geändert werden.

Wenn die Prozedur AZUZUNIT nicht in eine Systemprozedurbibliothek kopiert werden kann, fordern Sie die Benutzer von Developer for System z auf, ihrem Aufrufjob eine JCLLIB-Karte nach der JOB-Karte hinzuzufügen.

```
//MYJOB JOB <Jobparameters>  
//PROCS JCLLIB ORDER=(FEK.#CUST.PROCLIB)
```

- Zum Aufrufen des zUnit-Test Runners von z/OS UNIX (mithilfe des Scripts /usr/lpp/rdz/bin/zunit) können Sie die erforderlichen Dateien, die keine LINKLIST-Dateien sind, in der Anweisung STEPLIB von rsed.envvars angeben. Dadurch vereinfacht sich die Konfiguration für den Entwickler.

Die Datei rsed.envvars befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

Das Script *zunit* ermöglicht es dem Benutzer, Dateien anzugeben, die der vom Script verwendeten Anweisung STEPLIB hinzugefügt werden.

- Zum Aufrufen des zUnit-Test Runners über die TSO-Befehlszeile mithilfe der Exec FEK.SFEKPROC(FEKZUNIT) müssen die Systembibliotheken in LINKLIST vorhanden sein. Wenn dies nicht der Fall ist, müssen Entwickler die Systemdateien bei jeder Aufrufinstanz des zUnit-Test Runners angeben. Sie können auch eine Wrapper-Exec schreiben, die die **TSOLIB**-Zuordnungen dieser Dateien für sie durchführt. Sie können FEKZUNIT selbst als Beispiel dafür verwenden, wie diese Wrapper-Exec codiert werden muss.

Der zUnit-Test Runner ermöglicht eine automatische Neuformatierung von Testberichten. Developer for System z stellt Beispielkonvertierungen (z. B. die Konvertierung in das Ant- oder junit-Format) bereit, die sich in `/usr/lpp/rdz/samples/zunit/xsd` und `/usr/lpp/rdz/samples/zunit/xsl` befinden, wenn Sie Developer for System z an der Standardposition `/usr/lpp/rdz` installiert haben.

Enterprise Service Tools-Unterstützung (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Der Developer for System z-Client verfügt über eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools'. Abhängig vom generierten Codetyp liegen diesem Code Funktionen zugrunde, die durch die Hostsysteminstallation von Developer for System z bereitgestellt werden. In den folgenden Abschnitten wird beschrieben, wie Sie diese Hostsystemfunktionen verfügbar machen:

- Kapitel 5, „(Optional) Application Deployment Manager (veraltet)“, auf Seite 107
- „Unterstützung bidirektionaler Sprachen für CICS (optional)“
- „IRZ-Diagnosenachrichten für generierten Code (optional)“ auf Seite 132

Anmerkung: Die Enterprise Service Tools umfassen mehrere Tools, zu denen Service Flow Modeler (SFM) und XML Services for the Enterprise gehören.

Unterstützung bidirektionaler Sprachen für CICS (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines CICS-Administrators:

- JCL für die CICS-Region aktualisieren
 - Programm für CICS definieren
-

Die Komponente Enterprise Service Tools von Developer for System z unterstützt verschiedene Formate für arabische und hebräische Schnittstellennachrichten und die bidirektionale Datendarstellung und -bearbeitung in allen Editoren und Ansichten. In Terminalanwendungen werden Anzeigen von links nach rechts und von rechts nach links sowie numerische Felder und Felder mit entgegengesetzter Anzeigenausrichtung unterstützt.

Zu den zusätzlichen bidirektionalen Features und Funktionen gehören unter anderem:

- Der Enterprise Service Tools-Serviceanforderer gibt dynamisch bidirektionale Attribute von Schnittstellennachrichten an.
- Die bidirektionale Datenverarbeitung in Service-Flows basiert auf bidirektionalen Attributen wie Texttyp, Textausrichtung, numerischer Richtungswechsel und symmetrischer Richtungswechsel. Diese Attribute können in verschiedenen Stadien der Erstellung von Schnittstellen- und Terminal-Flows angegeben werden.
- Der von Enterprise Service Tools generierte Laufzeitcode umfasst die Umsetzung von Daten in Feldern von Nachrichten mit verschiedenen bidirektionalen Attributen.

Von Enterprise Service Tools generierter Code kann die BIDI-Konvertierung auch in anderen Umgebungen als CICS SFR (Service Flow Runtime) unterstützen. Ein Beispiel sind Batchanwendungen. Sie können die Enterprise Service Tools-Generatoren veranlassen, alle Aufrufe bidirektionaler Umsetzungsroutinen aufzunehmen, indem Sie in den Generierungsassistenten von Enterprise Service Tools die entsprechenden BIDI-Konvertierungsattribute angeben und die generierten Programme mit der entsprechenden Bibliothek für bidirektionale Umsetzung (FEK.SFEKLOAD) verknüpfen.

Führen Sie die folgenden Tasks aus, um die CICS-Unterstützung für bidirektionale Sprachen zu aktivieren:

1. Stellen Sie die FEK.SFEKLOAD-Ladmodule FEJBDCMP und FEJBDTRX in die CICS-RPL-Verkettung (DD-Anweisung DFHRPL). Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.

Wichtig: Wenn Sie nicht die Installationsdatei zur Kette hinzufügen, sondern die Module in eine neue oder vorhandene Datei kopieren, beachten Sie, dass diese Module DLLs sind und in einer PDSE-Bibliothek enthalten sein müssen.

2. Definieren Sie FEJBDCMP und FEJBDTRX mithilfe des entsprechenden CEDA-Befehls als Programme für CICS. Beispiel:

```
CEDA DEF PROG(FEJBDCMP) LANG(LE) G(XXX)
CEDA DEF PROG(FEJBDTRX) LANG(LE) G(XXX)
```

IRZ-Diagnosenachrichten für generierten Code (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind aber die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich:

- LINKLIST aktualisieren
 - JCL für die CICS-Region aktualisieren
-

Im Client von Developer for System z gibt es eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools'. Alle IRZM*- und IIRZ*-Module in der Ladebibliothek FEK.SFEKLMOD müssen dem generierten Code verfügbar gemacht werden, damit der von Enterprise Service Tools generierte Code Diagnosefehler-nachrichten ausgeben kann. Enterprise Service Tools kann Code für die folgenden Umgebungen generieren:

- CICS
- IMS
- MVS-Batch

Wenn der generierte Code in einer CICS-Transaktion ausgeführt wird, fügen Sie alle IRZM*- und IIRZ*-Module in FEK.SFEKLMOD zur DD "DFHRPL" der CICS-Region hinzu. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch verfügbar sind.

Machen Sie in allen anderen Situationen alle IRZM*- und IIRZ*-Module in der Ladebibliothek FEK.SFEKLMOD mithilfe von STEPLIB oder LINKLIST verfügbar. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch verfügbar sind.

Wenn Sie STEPLIB verwenden, definieren Sie die nicht über LINKLIST verfügbaren Module in der Anweisung STEPLIB der Task, die den Code ausführt.

Wenn die Lademodule nicht verfügbar sind und der generierte Code einen Fehler feststellt, wird die folgende Nachricht ausgegeben:

IRZ9999S Abruf des Texts der Laufzeitnachricht 'Language Environment' ist fehlgeschlagen. Überprüfen Sie, ob das Laufzeitnachrichtenmodul 'Language Environment' für Facility-IRZ in DFHRPL oder STEPLIB installiert ist.

Anmerkung:

- Das Modul FEK.SFEKLMOD(IRZPWSIO) wird während der IMS-MPP-Top-down-Codegenerierung statisch verknüpft. Aus diesem Grund muss das Modul nicht während der Laufzeit des generierten Codes verfügbar sein. Es sollte nur bei der Kompilierung verfügbar sein.
- Für Version 8.5 wurden die IRZ*- und IIRZ*-Lademodule und Diagnosenachrichten aus der Ladebibliothek FEK.SFEKLOAD in FEK.SFEKLMOD verschoben.
- In Version 9.0.1 enthalten FEK.SFEKLMOD(IRZPWSIO) und das zugehörige PL/I-Beispiel FEK.SFEKSAMP(IRZPWSH) Member, die von Developer for System z nach IMS Version 12 verschoben wurden. Die Module werden entsprechend in IMS.SDFSRESL(DFSPWSIO) und IMS.SDFSMSPL(DFSPWSH) umbenannt.

Integrated Debugger (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder spezielle Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines CICS-Administrators, eines TCP/IP- und eines Sicherheitsadministrators:

- LINKLIST aktualisieren
- APF-Autorisierung
- Gestartete Task definieren
- Sicherheitsprofile und Zugriffslisten definieren
- TCP/IP-Ports für Client-Host und hostinterne Kommunikation reservieren
- (Optional) SVC hinzufügen (erfordert IPL)
- (Optional) Link-Pack-Bereich für SVC aktualisiert
- (Optional) JCL für CICS-Regionen aktualisieren
- (Optional) CICS-Systemdefinition aktualisieren

Mit der Hostkomponente Developer for System z Integrated Debugger können Clients ab Version 9.0.1 bei verschiedenen LE-basierten Anwendungen (LE - Language Environment) Fehler beheben. Hierzu zählen auch CICS-Transaktionen, die in schreibgeschützten Speicher geladen wurden.

Einen Überblick über den Datenfluss von Integrated Debugger finden Sie im Abschnitt "Integrated Debugger" des Kapitels "Wissenswertes zu Developer for System z" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

Für die Verwendung von Integrated Debugger an Ihrem Standort müssen Sie die folgenden Tasks ausführen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Für Integrated Debugger muss die optionale gestartete DBGMGR-Task aktiv sein (zusammen mit der obligatorisch gestarteten Task RSED). Informationen zur DBGMGR-Start-JCL finden Sie im Abschnitt „DBGMGR, gestartete Task für den Debug-Manager“ auf Seite 27.
2. Die Konfiguration von Integrated Debugger wird über Startargumente der gestarteten DBGMGR-Task verwaltet. Details hierzu finden Sie im Abschnitt „Konfigurationsparameter für Integrated Debugger“ auf Seite 135.
3. Für die gestartete DBGMGR-Task muss die Bibliothek FEK.SFEKAUTH APF-autorisiert sein. Details hierzu finden Sie im Abschnitt „Parmlib-Aktualisierungen für Integrated Debugger“ auf Seite 135.
4. Ihre Anwendung muss auf Integrated Debugger zugreifen können und über STEPLIB- bzw. LINKLIST-Aktualisierungen verfügen. Details hierzu finden Sie im Abschnitt „Parmlib-Aktualisierungen für Integrated Debugger“ auf Seite 135.
5. Für Integrated Debugger muss die Benutzer-ID für die Anwendung, bei der Fehler behoben werden sollen, über ein gültiges OMVS-Segment verfügen. Details hierzu finden Sie im Abschnitt „Sicherheitsupdates für Integrated Debugger“ auf Seite 136.
6. Für die gestartete DBGMGR-Task sind einige Sicherheitsberechtigungen erforderlich. Details hierzu finden Sie im Abschnitt „Sicherheitsupdates für Integrated Debugger“ auf Seite 136.

Die folgenden Schritte sind nur für die Fehlerbehebung bei CICS-Transaktionen erforderlich:

1. Mit Integrated Debugger können Fehler bei CICS-Transaktionen behoben werden. Dies setzt voraus, dass Integrated Debugger für CICS definiert ist. Details hierzu finden Sie im Abschnitt „CICS-Aktualisierungen für Integrated Debugger“ auf Seite 136.
2. (Optional) Integrated Debugger kann Fehler bei CICS-Transaktionen beheben, die in den Nur-Lese-Speicher geladen wurden. Hierzu muss ein Supervisoraufruf (SVC - Supervisor Call) für Developer for System z für Ihr System definiert werden. Das zugehörige Lademodul muss beim einleitenden Programmladen in den Link-Pack-Bereich geladen werden. Details hierzu finden Sie im Abschnitt „Parmlib-Aktualisierungen für Integrated Debugger“ auf Seite 135.
Wenn der Supervisoraufruf in einer Umgebung mit Fehlerstatus (nicht autorisierte Umgebung) verwendet wird, benötigen Benutzer eine Berechtigung für ein Sicherheitsprofil. Details hierzu finden Sie im Abschnitt „Sicherheitsupdates für Integrated Debugger“ auf Seite 136.

Beachten Sie, dass nur ein LE-basierter Debugger (LE - Language Environment), wie beispielsweise die Komponente 'Integrated Debugger', in einer bestimmten Anwendung oder CICS-Region aktiv sein kann. Ob es sich bei einem Debugger um einen LE-basierten Debugger handelt, können Sie gut daran erkennen, dass er ein CEEVDBG-Lademodul oder ein Alias bereitstellt, das für die Anwendung verfügbar sein muss.

Konfigurationsparameter für Integrated Debugger

Integrated Debugger ermöglicht eine Konfiguration der folgenden Variablen im Start-JCL-Code für DBGMR. Informationen zur DBGMR-Start-JCL finden Sie im Abschnitt „DBGMR, gestartete Task für den Debug-Manager“ auf Seite 27.

- Die Zeitzonendifferenz (standardmäßig EST5DST)
- Der für die externe Kommunikation (Client-Host-Kommunikation) verwendete Port (Standardwert 5335)
- Der für die interne, auf den Host begrenzte Kommunikation verwendete Port (standardmäßig 5336)
- Die für die Fehlerbehebung bei CICS-Transaktionen im Nur-Lese-Speicher verwendete SVC-Nummer (standardmäßig 251)
- Das übergeordnete Qualifikationsmerkmal der Ladebibliothek (standardmäßig FEK)

Parmlib-Aktualisierungen für Integrated Debugger

- Für die gestartete DBGMR-Task muss die Bibliothek FEK.SFEKAUTH APF-autorisiert sein. Details hierzu finden Sie im Abschnitt „APF-Berechtigungen in PROGxx“ auf Seite 22.
- Integrated Debugger muss in LE (Language Environment) aufgerufen werden können. Die Bibliothek FEK.SFEKAUTH muss deshalb in der Anwendung, bei der Fehler behoben werden sollen, in LINKLIST oder STEPLIB enthalten sein. Details hierzu finden Sie im Abschnitt „LINKLIST-Definitionen in PROGxx“ auf Seite 23.

Anmerkung: Stellen Sie bei Verwendung von LINKLIST sicher, dass FEK.SFEKAUTH vor den Bibliotheken anderer LE-basierter Debugger mit dem Lademodul CEEVDBG oder CEEV006 aufgeführt wird. Dies ist z. B. bei IBM Debug Tool for z/OS zu beachten, das h1q.SEQA*-Bibliotheken verwendet.

- Integrated Debugger verwendet die z/OS-Binder-API. Diese API ist ab z/OS 1.10 als /usr/lib/iewbndd.so verfügbar, ab z/OS 1.13 auch als SYS1.SIEAMIGE(IEWBND). Dies impliziert, dass ab z/OS 1.13 SYS1.SIEAMIGE ebenfalls in LINKLIST (oder STEPLIB) enthalten sein sollte. Details hierzu finden Sie im Abschnitt „Vorausgesetzte LINKLIST- und LPA-Definitionen“ auf Seite 24.

Anmerkung: Ist SYS1.SIEAMIGE bei Systemen mit z/OS Version 1.13 oder einer höheren Version nicht in LINKLIST oder STEPLIB enthalten, gibt Integrated Debugger die folgende Nachricht heraus und versucht, /usr/lib/iewbndd.so zu verwenden:

```
CEE3501S The module //IEWBND was not found
```

- Integrated Debugger kann Fehler in CICS-Transaktionen beheben, die in den Nur-Lese-Speicher geladen wurden. Hierzu muss ein Supervisoraufruf (SVC - Supervisor Call) für Developer for System z für Ihr System definiert werden. Die SVC-Standardnummer lautet 251. Das zugehörige Lademodul FEK.SFEKLPA(AQESVC01) muss beim einleitenden Programmladen in den Link-Pack-Bereich geladen werden. Details hierzu finden Sie in den Abschnitten „SVC-Definitionen in IEASVCxx“ auf Seite 21 und „LPA-Definitionen in LPALSTxx“ auf Seite 21.

TCP/IP-Aktualisierungen für Integrated Debugger

Integrated Debugger verwendet zwei TCP/IP-Ports. Details hierzu finden Sie im Abschnitt "Hinweise zu TCP/IP" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

- Port für die Client-Host-Kommunikation (standardmäßig 5335). Die Kommunikation an diesem Port kann verschlüsselt sein.
- Port für die hostinterne Kommunikation (standardmäßig 5336).

Sicherheitsupdates für Integrated Debugger

Die folgenden Sicherheitsdefinitionen sind für Integrated Debugger erforderlich. Details hierzu finden Sie im Abschnitt „Sicherheitsdefinitionen“ auf Seite 33.

- OMVS-Segment für die Benutzer-ID, mit der die Anwendung ausgeführt wird, bei der Fehler behoben werden sollen.
- Die gestartete DBGMR-Task
- Die BPX.SERVER-Berechtigung für die Benutzer-ID der gestarteten Task
- Programmsteuerung für die Ladebibliothek der gestarteten Task

Die folgenden Sicherheitsdefinitionen können optional Integrated Debugger verwendet werden. Details hierzu finden Sie im Abschnitt "Sicherheitsaspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

- Die AQE.AUTHDEBUG.WRITEBUFFER-Berechtigung, mit der Benutzer Fehler bei CICS-Transaktionen beheben können, die in einer Umgebung mit Fehlerstatus (nicht autorisierte Umgebung) in einen schreibgeschützten Speicher geladen wurden.

Anmerkung: Um die Migration von einer vorhandenen Developer for System z-Konfiguration ohne Integrated Debugger zu erleichtern, wird die Beispiel-JCL FEK.SFEKSAMP(AQERACF) mit RACF-Befehlen bereitgestellt, sodass nur die Sicherheitsdefinitionen für Integrated Debugger definiert werden.

CICS-Aktualisierungen für Integrated Debugger

Um Fehler bei CICS-Transaktionen zu beheben, sind die folgenden CICS-Aktualisierungen für Integrated Debugger erforderlich.

- CICS-JCL-Aktualisierung:
 - Definieren Sie die Ladebibliothek FEK,SFEKAUTH in der DD-Anweisung DFHRLPL der Region, wenn die Bibliothek nicht in LINKLIST vorhanden ist.
 - Definieren Sie für z/OS ab Version 1.13 die Ladebibliothek SYS1.SIEAMIGE in der DD-Anweisung STEPLIB der Region, wenn die Bibliothek nicht in LINKLIST vorhanden ist. Nähere Erläuterungen hierzu finden Sie in den Informationen zur z/OS-Binder-API im Abschnitt „Parmlib-Aktualisierungen für Integrated Debugger“ auf Seite 135.
- Aktualisierungen für die CICS-Systemdefinitionsdatei:

Definieren Sie den Debugger für eine CICS-Region wie dies in dem AQECSD-Beispieljob für die Aktualisierung einer CICS-Systemdefinitionsdatei dokumentiert ist. AQECSD befindet sich in FEK.#CUST.JCL, sofern Sie beim Anpassen und Übergeben des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 17.

Unterstützung für Tools zur Problembestimmung (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Developer for System z kann mit verschiedenen IBM z/OS-Tools zur Problembestimmung integriert werden. In den folgenden Abschnitten wird beschrieben, wie Sie diese Tools für den Developer for System z-Client verfügbar machen können:

- IBM Debug Tool for z/OS: Siehe „Unterstützung für DB2- und IMS-Debug (optional)“. Beachten Sie, dass die Komponente 'Integrated Debugger' ab Version 9.0.1 im Lieferumfang von Developer for System z enthalten ist. Diese kann anstelle von Debug Tool verwendet werden.
- IBM File Manager for z/OS: Siehe „Unterstützung für File Manager (optional)“ auf Seite 138.
- IBM Fault Analyzer for z/OS: Es ist keine Hostsystemkonfiguration für Developer for System z erforderlich. Beachten Sie, dass Fault Analyzer Integration (FAI) von Developer for System z ab Version 9.0 nicht mehr unterstützt wird. Auf älteren Clients, die diese Unterstützung noch haben, sollte die Funktion deinstalliert und das IBM Fault Analyzer-Plug-in für Eclipse installiert werden. Dieses Plug-in steht auf der Webseite für IBM Problem Determination Tools-Plug-ins unter der Adresse <http://www-01.ibm.com/software/awdtools/deployment/pdtpplugins/> zur Verfügung.

Unterstützung für DB2- und IMS-Debug (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks für die Konfiguration von Developer for System z erforderlich. Es müssen jedoch Voraussetzungen für die Konfiguration von IBM Debug Tool for z/OS erfüllt werden.

IBM Debug Tool for z/OS stellt einen angepassten Language Environment-Benutzerexit (LE-Benutzerexit) CEEBXITA bereit, der die TEST-Laufzeitoptionen zurückgibt, wenn er von der LE-Initialisierungslogik in gespeicherten Prozeduren von IMS und DB2 aufgerufen wird. IBM Debug Tool for z/OS stellt auch die Debug Tool-Erweiterung für Problem Determination Tools Common Components Server bereit, um die Dateien der TEST-Laufzeitoptionen für das z/OS-System zu erstellen und zu verwalten. Developer for System z kann die Unterstützung von IBM Debug Tool for z/OS für die Verwaltung von Debugprofilen für Laufzeiten von gespeicherten Prozeduren von IMS und DB2 verwenden und erweitern.

In der Dokumentation von IBM Debug Tool for z/OS wird die erforderliche Konfiguration, die hier nur kurz erwähnt wird, im Detail beschrieben.

- Angabe der TEST-Laufzeitoptionen über den Language Environment-Benutzerexit hlq.SEQA*
- Unterstützung für die DTSP-Ansicht 'Profil'
 - Installation von Problem Determination Tools Common Components Server (hlq.SIPV*, Job IPVGSVRJ)
 - Installation und Konfiguration der Debug Tool-Erweiterung für Problem Determination Tools Common Components (hlq.SEQA*)

Anmerkung:

- Das Produkt IBM Debug Tool for z/OS muss separat erworben, installiert und konfiguriert werden. Die Installation und Anpassung dieses Produkts ist nicht in diesem Handbuch beschrieben.
- Der Developer for System z-Client verwendet nicht das Plug-in für die DTSP-Ansicht 'Profil' für Eclipse.

- Der Developer for System z-Client verwendet für das reguläre Debugging im Stapelmodus nicht den Language Environment-Benutzerexit.
- Der Developer for System z-Client kommuniziert direkt mit dem Problem Determination Tools Common Components-Server. Dies setzt voraus, dass der Benutzer diese Portnummer kennen muss und dass der von diesem Server verwendete Port in der Firewall, die den z/OS-Host schützt, geöffnet sein muss.

Unterstützung für File Manager (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks für die Konfiguration von Developer for System z erforderlich. Es müssen jedoch Voraussetzungen für die Konfiguration von IBM File Manager for z/OS erfüllt werden.

Die ursprüngliche Integration von Developer for System z mit IBM File Manager for z/OS wurde in Developer for System z Version 8.0.3 als veraltet markiert und wird in Version 8.5 nicht mehr unterstützt. Die von dieser Funktion bereitgestellten Services wurden in andere Bereiche verschoben. Einige Funktionen, beispielsweise die unformatierte QSAM-Bearbeitung, sind jetzt Bestandteil der regulären Dateihandhabung durch Developer for System z. Für erweiterte Funktionen wie beispielsweise die Bearbeitung formatierter Daten unter Verwendung von Copybooks oder Kopfdatendateien ist erforderlich, dass das IBM File Manager-Plug-in für Eclipse im Developer for System z-Client installiert ist. Dieses Plug-in steht auf der Webseite für IBM Problem Determination Tools-Plug-ins unter der Adresse <http://www-01.ibm.com/software/awdtools/deployment/pdtpplugins/> zur Verfügung.

Das IBM File Manager-Plug-in für Eclipse verwendet den Server für die Tools für Fehlerbestimmung (Problem Determination Tools Server) für den Zugriff auf die File Manager-Services. Dieser Server wird von der File Manager-ISPF-Anzeigenschnittstelle nicht verwendet. Daher müssen Sie zusätzliche File Manager-Konfigurationstasks ausführen, die für die Tools für Fehlerbestimmung spezifisch sind. Weitere Details finden Sie in der File Manager-Dokumentation.

Die vom Server für die Tools für Fehlerbestimmung verwendete Portnummer muss in der `rsed.envvars`-Anweisung `PD_SERVER_PORT` angegeben werden.

Die Datei `rsed.envvars` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Anmerkung:

- Das Produkt IBM File Manager for z/OS muss separat erworben, installiert und konfiguriert werden. Die Installation und Anpassung dieses Produkts ist nicht in diesem Handbuch beschrieben.
- Der Developer for System z-Client kommuniziert direkt mit dem Server für die Tools für Fehlerbestimmung (Problem Determination Tools). Dies bedeutet, dass der von diesem Server verwendete Port in Ihrer Firewall, die das z/OS-Hostsystem schützt, geöffnet sein muss.

Bereinigung von "WORKAREA" und "/tmp" (optional)

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Das TSO/ISPF-Client-Gateway von ISPF und SCLM Developer Toolkit speichern in den Verzeichnissen WORKAREA und /tmp temporäre Arbeitsdateien, die vor dem Schließen der Sitzung entfernt werden. Temporäre Ausgaben bleiben jedoch manchmal enthalten. Dies ist beispielsweise der Fall, wenn während der Verarbeitung ein Kommunikationsfehler auftritt. Bereinigen Sie daher die Verzeichnisse WORKAREA und /tmp von Zeit zu Zeit.

z/OS UNIX stellt das Shell-Script skulker bereit, das Dateien auf der Basis des Verzeichnisses, in dem sie sich befinden, und ihres Alters löscht. In Kombination mit dem z/OS UNIX-Dämon cron, der Befehle zu angegebenen Zeitpunkten (Datum/Uhrzeit) ausführt, können Sie ein automatisiertes Tool einrichten, das bestimmte Zielverzeichnisse in regelmäßigen Abständen bereinigt. Weitere Informationen zum Script skulker und den Dämon cron finden Sie in *UNIX System Services Command Reference* (IBM Form SA22-7802).

Anmerkung: Das Verzeichnis WORKAREA befindet sich in /var/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Kapitel 8. Installationsprüfung

Nach der vollständigen Produktanpassung können Sie die in diesem Kapitel beschriebenen IVPs (Installation Verification Programs) verwenden, um die erfolgreiche Konfiguration der zentralen Produktkomponenten zu überprüfen.

Gestartete Tasks prüfen

JMON - JES Job Monitor

Starten Sie die gestartete Task bzw. den Benutzerjob JMON. Die Startinformationen in DD SYSOUT sollten mit der folgenden Nachricht enden:

```
FEJ211I Server ready to accept connections.
```

Wenn der Job mit dem Rückgabecode 66 beendet wird, verfügt FEK.SFEKAUTH nicht über eine APF-Berechtigung.

Anmerkung: Starten Sie JES Job Monitor, bevor Sie mit weiteren Installationsprüftests (IVP, Installation Verification Program) fortfahren.

RSED - RSE-Dämon

Starten Sie die gestartete Task bzw. den Benutzerjob RSED mit dem Parameter IVP=IVP. Bei Verwendung dieses Parameters wird der Server nach Ausführung einiger Installationsprüftests beendet. Die Ausgabe dieser Tests ist in DD STDOUT verfügbar. Bei bestimmten Fehlern sind auch in DD STDERR Daten verfügbar. Überprüfen Sie die DD STDOUT auf Nachrichten, die angeben, dass die folgenden IVPs erfolgreich waren:

- Java-Start
- JES Job Monitor-Verbindung
- TCP/IP konfigurieren

Die STDOUT-Daten sollten wie im folgenden Beispiel aussehen:

```
-----
RSE daemon startup script
-----

arguments: IVP -C/etc/rdz -P

RSE daemon IVP test

CDFMVS08 -- Fri Mar 23 17:50:52 2012 UTC
uid=8(STCRSE) gid=1(STCGROUP)

started from /usr/lpp/rdz/bin/rsed.sh
startup script version Aug09,2012

configuration files located in /etc/rdz -- startup argument
daemon port is 4035 -- set in rsed.envvars
debug level is 1 -- set in rsecomm.properties
TMPDIR=/tmp -- default

-----
current environment variables
-----
@="/usr/lpp/rdz/bin/rsed.sh" @[1]="-C/etc/rdz" @[2]="-P"
```

```

ANT_HOME="/usr/lpp/apache/Ant/apache-ant-1.7.1"
CGI_DWORK="/var/rdz"
CGI_ISPCONF="/etc/rdz"
CGI_ISPHOME="/usr/lpp/ispf"
CGI_ISPWORK="/var/rdz"
CGI_TRANTABLE="FEK.#CUST.LSTRANS.FILE"
CLASSPATH=".:usr/lpp/rdz/lib:usr/lpp/rdz/lib/dstore_core.jar:usr/lpp/
ERRNO="0"
HOME="/tmp"
IFS="
"
JAVA_HOME="/usr/lpp/java/J6.0"
JAVA_PROPAGATE="NO"
LANG="C"
LIBPATH=".:usr/lib:usr/lpp/java/J6.0/bin:usr/lpp/java/J6.0/bin/classi
LINENO="66"
LOGNAME="STCRSE"
MAILCHECK="600"
OLDPWD="/tmp"
OPTIND="1"
PATH=".:usr/lpp/java/J6.0/bin:usr/lpp/rdz/bin:usr/lpp/ispf/bin:/bin:/
PPID="33554711"
PS1="\$ "
PS2="> "
PS3="#? "
PS4="+ "
PWD="/etc/rdz"
RANDOM="27298"
RSE_CFG="/etc/rdz"
RSE_HOME="/usr/lpp/rdz"
RSE_LIB="/usr/lpp/rdz/lib"
SECONDS="0"
SHELL="/bin/sh"
STEPLIB="NONE"
TMPDIR="/tmp"
TZ="EST5EDT"
X_ARG="-T"
X_C="-- startup argument"
X_KEY="-T"
X_L="-- set in rsecomm.properties"
X_LOG="1"
X_P="-- set in rsed.envvars"
X_PORT="4035"
X_VAL=""
_="-----"
_BPX_SHAREAS="YES"
_BPX_SPAWN_SCRIPT="YES"
_CEE_DMPTARG="/tmp"
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_CMDSERV_BASE_HOME="/usr/lpp/ispf"
_CMDSERV_CONF_HOME="/etc/rdz"
_CMDSERV_WORK_HOME="/var/rdz"
_EDC_ADD_ERRNO2="1"
_RSE_ISPF_OPTS="&SESSION=SPAWN"
_RSE_DAEMON_CLASS="com.ibm.etools.zos.server.RseDaemon"
_RSE_DAEMON_IVP_TEST="1"
_RSE_HOST_CODEPAGE="IBM-1047"
_RSE_JAVA_OPTS="-DISPF_OPTS='&SESSION=SPAWN' -DA_PLUGIN_PATH=
_RSE_JMON_PORT="6715"
_RSE_LOG_LEVEL="1"
_RSE_POOL_SERVER_CLASS="com.ibm.etools.zos.server.ThreadPoolProcess"
_RSE_RSED_PORT="4035"
_RSE_SAF_CLASS="/usr/include/java_classes/IRRacf.jar"
_RSE_SCRIPT_VERSION="Jan09,2012"
_RSE_SERVER_CLASS="org.eclipse.dstore.core.server.Server"
_RSE_SERVER_TIMEOUT="120000"
_SCLMDT_BASE_HOME="/usr/lpp/rdz"

```

```
_SCLMDT_CONF_HOME="/var/rdz/sclmdt"
_SCLMDT_TRANSTABLE="FEK.#CUST.LSTRANS.FILE"
_SCLMDT_WORK_HOME="/var/rdz"
debug_level="1"
```

----- Address Space size limits

```
current address space size limit is 1913626624 (1825.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

----- service history

```
Fri Jun 14 13:47:39 2013 -- COPY -- HHOP900 v9000 created 14 Jun 2013
```

----- java service level

```
java full version "J2RE 1.6.0 IBM z/OS build pmz3160sr13-20130207_01(SR13)
```

----- LE runtime options

```
Options Report for Enclave main 05/23/12 1:50:52 PM
Language Environment V01 R11.00
```

LAST WHERE SET	OPTION
Installation default	ABPERC(NONE)
Programmer default	ABTERMENC(RETCODE)
Installation default	NOAIXBLD
Invocation command	ALL31(ON)
Programmer default	ANYHEAP(32768,16384,ANYWHERE,FREE)
Installation default	NOAUTOTASK
Programmer default	BELOWHEAP(32768,16384,FREE)
Installation default	CBLOPTS(ON)
Installation default	CBLPSTHPOP(ON)
Installation default	CBLQDA(OFF)
Installation default	CEEDUMP(60,SYSOUT=*,FREE=END,SPIN=UNALL
Installation default	CHECK(ON)
Installation default	COUNTRY(US)
Installation default	NODEBUG
Installation default	DEPTHCONDLMT(10)
Installation default	DYNDUMP(*USERID,NODYNAMIC,TDUMP)
Installation default	ENVAR("")
Installation default	ERRCOUNT(0)
Installation default	ERRUNIT(6)
Installation default	FILEHIST
Installation default	FILETAG(NOAUTOCVT,NOAUTOTAG)
Default setting	NOFLOW
Invocation command	HEAP(33554432,32768,ANYWHERE,KEEP,16384
Installation default	HEAPCHK(OFF,1,0,0,0)
Installation default	HEAPPOLS(OFF,8,10,32,10,128,10,256,10,
Installation default	INFOMSGFILTER(OFF,,,))
Installation default	INQPCOPN
Installation default	INTERRUPT(OFF)
Programmer default	LIBSTACK(32768,16384,FREE)
Installation default	MSGFILE(SYSOUT,FBA,121,0,NOENQ)
Installation default	MSGQ(15)
Installation default	NATLANG(ENU)
Ignored	NONONIPSTACK(See THREADSTACK)
Installation default	OCSTATUS
Installation default	NOPC
Installation default	PLITASKCOUNT(20)
Programmer default	POSIX(ON)

```

Installation default      PROFILE(OFF,"")
Installation default      PRTUNIT(6)
Installation default      PUNUNIT(7)
Installation default      RDRUNIT(5)
Installation default      RECPAD(OFF)
Invocation command        RPTOPTS(ON)
Installation default      RPTSTG(OFF)
Installation default      NORTEREUS
Installation default      NOSIMVRD
Programmer default
STACK(65536,65536,ANYWHERE,KEEP,524288,131072)
Installation default      STORAGE(NONE,NONE,NONE,0)
Installation default      TERMTHDACT(TRACE,,96)
Installation default      NOTEST(ALL,"*","PROMPT","INSPREF")
Installation default      THREADHEAP(4096,4096,ANYWHERE,KEEP)
Installation default      THREADSTACK(OFF,4096,4096,ANYWHERE,KEEP,131072,
Installation default      TRACE(OFF,4096,DUMP,LE=0)
Invocation command        TRAP(ON,SPIE)
Installation default      UPSI(00000000)
Installation default      NOUSRHDLR(,)
Installation default      VCTRSAVE(OFF)
Installation default      XPLINK(OFF)
Installation default      XUFLOW(AUTO)

-----
java startup test...
-----
java full version "JRE 1.6.0 IBM z/OS build pmz3160sr13-20130207_01
(SR13)"
java version "1.6.0"
Java(TM) SE Runtime Environment (build pmz3160sr13-20130207_01(SR13))
IBM J9 VM (build 2.4, JRE 1.6.0 IBM J9 2.4 z/OS s390-31 jvmmz3160sr13-
20130114_1
J9VM - 20130114_134867
JIT - r9_20130108_31100
GC - 20121212_AA
JCL - 20130204_01

-----
JES Job Monitor test...
-----

executed on CDFMVS08 -- Fri Mar 23 17:50:52 EDT 2012
executed by uid=8(STCRSE) gid=1(STCGROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1913626624 (1825.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

testing JES Job Monitor on port 6715...
hostName=CDFMVS08
hostAddr=9.42.112.75
IPv4 is supported
Waiting for JES Job Monitor response...
ACKNOWLEDGE01v03
Success

-----
TCP/IP IVP test...
-----

executed on CDFMVS08 -- Fri Mar 23 17:50:53 EDT 2012
executed by uid=8(STCRSE) gid=1(STCGROUP)
using /etc/rdz/rsed.envvars

current address space size limit is 1913626624 (1825.0 MB)

```

```

maximum address space size limit is 2147483647 (2048.0 MB)
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
Resolver Trace Initialization Complete -> 2012/05/23 17:50:54.208378

res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table = Default
UserId/JobName = STCRSE
Caller API = LE C Sockets
Caller Mode = EBCDIC
(L) DataSetPrefix = TCP/IP
(L) HostName = CDFMVS08
(L) TcpIpJobName = TCP/IP
(L) DomainOrigin = RALEIGH.IBM.COM
(L) NameServer = 9.42.206.2
                  9.42.206.3
(L) NsPortAddr = 53 (L) ResolverTimeout = 10
(L) ResolveVia = UDP (L) ResolverUdpRetries = 1
(*) Options NDots = 1
(*) SockNoTestStor
(*) AlwaysWto = NO (L) MessageCase = MIXED
(*) LookUp = DNS LOCAL
res_init Succeeded
res_init Started: 2012/05/23 17:50:54.229888
res_init Ended: 2012/05/23 17:50:54.229898
*****
MVS TCP/IP NETSTAT CS V1R11 TCP/IP Name: TCP/IP 17:50:54
Tcpip started at 11:31:40 on 05/23/2012 with IPv6 enabled

-----
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match

-----
RSE daemon IVP ended -- return code 0 -- Fri Mar 23 17:50:55 EDT 2012
-----

```

Anmerkung: Starten Sie den RSE-Dämon ohne die IVP-Parameter, bevor Sie mit weiteren IVP-Tests fortfahren. Der RSE-Dämon gibt nach einem erfolgreichen Start die folgende Konsolennachricht aus:

```
FEK002I RseDaemon started. (port=4035)
```

DBGMGR, Debug-Manager

Starten Sie die optionale gestartete Task bzw. den Benutzerjob DBGMGR. Der Server gibt nach einem erfolgreichen Start die folgende Konsolennachricht aus, wobei clientport für die Nummer des Ports steht, der für die externe Kommunikation (Client-Host) verwendet wird, und hostport für die Portnummer für interne, auf den Host begrenzte Kommunikation.

```
AQECM001I Debug Manager startup complete (clientport/hostport)
```

Wenn der Job mit dem Rückgabecode 66 beendet wird, verfügt FEK.SFEKAUTH nicht über eine APF-Berechtigung.

| **Anmerkung:** Starten Sie DBGMGR, bevor Sie mit den anderen auf das Debugging
| bezogenen IVP-Tests fortfahren.

IVP-Bedienerbefehle

Ein aktiver RSE-Dämon unterstützt den **IVP**-Änderungsbefehl, mit dem Sie ausgewählte IVPs über die Konsole ausführen können.

Wiederverwendbarkeit von PassTicket

Developer for System z erfordert, dass die generierten PassTickets wiederverwendet werden können, da die PassTicket-Erzeugung auf eines für einen Benutzer pro Sekunde beschränkt ist. Prüfen Sie die PassTicket-Wiederverwendbarkeit, indem Sie folgenden Bedienerbefehl ausführen. Ersetzen Sie userid durch eine gültige TSO-Benutzer-ID.

```
MODIFY RSED,APPL=IVP PASSTICKET,userid
```

Die vom Befehl zurückgegebene Ausgabe sollte etwa wie im folgenden Beispiel aussehen:

```
MODIFY RSED,APPL=IVP PASSTICKET,IBMUSER
```

```
+FEK900I PASSTICKET IVP: start: serverid=STCRSE userid=IBMUSER  
+FEK900I PASSTICKET IVP: the default applid=FEKAPPL  
+FEK900I PASSTICKET IVP: Success, PassTicket IVP finished normally  
+FEK901I PASSTICKET IVP Exit code = 0
```

RSE-Dämonverbindung

Führen Sie den folgenden Befehl aus, um die RSE-Dämonverbindung zu überprüfen. Ersetzen Sie userid durch eine gültige TSO-Benutzer-ID.

```
MODIFY RSED,APPL=IVP DAEMON,userid
```

Dieser Befehl entspricht funktional dem in „Services prüfen“ auf Seite 147 beschriebenen IVP fekfivpd, jedoch mit dem Vorteil, dass kein Kennwort erforderlich ist. RSE generiert ein PassTicket und verwendet dieses als Kennwort. Die vom Befehl zurückgegebene Ausgabe sollte etwa wie im folgenden Beispiel aussehen:

```
F RSED,APPL=IVP DAEMON,IBMUSER
```

```
+FEK900I DAEMON IVP: SSL is disabled  
+FEK900I DAEMON IVP: connected  
+FEK900I DAEMON IVP: 1343  
+FEK900I DAEMON IVP: 8878350  
+FEK900I DAEMON IVP: Success  
+FEK901I DAEMON IVP Exit code = 0
```

ISPF Client Gateway

Überprüfen Sie die ISPF-Client-Gateway-Verbindung, indem Sie folgenden Befehl ausführen. Ersetzen Sie userid durch eine gültige TSO-Benutzer-ID.

```
MODIFY RSED,APPL=IVP ISPF,userid
```

Dieser Befehl entspricht funktional dem in „Services prüfen“ auf Seite 147 beschriebenen IVP fekfivpi. Die vom Befehl zurückgegebene Ausgabe sollte etwa wie im folgenden Beispiel aussehen:

```
F RSED,APPL=IVP ISPF,IBMUSER
```

```
+FEK900I ISPF IVP: executed on CDFMVS08 -- Tue Sep 13 22:29:28 EDT 2011  
+FEK900I ISPF IVP: executed by uid=1(IBMUSER) gid=0(SYS1)  
+FEK900I ISPF IVP: using /etc/rdz/rsed.envvars
```

```

+FEK900I ISPF IVP: current address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: maximum address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: /etc/rdz/ISPF.conf content:
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: ispllib=ISP.SISPLLOAD
+FEK900I ISPF IVP: isplib=ISP.SISPMENU
+FEK900I ISPF IVP: isptlib=ISP.SISPTEU
+FEK900I ISPF IVP: ispplib=ISP.SISPPENU
+FEK900I ISPF IVP: ispslib=ISP.SISPSLIB
+FEK900I ISPF IVP: sysproc=ISP.SISPCLIB,FEK.SFEKPROC
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host install verification for RSE
+FEK900I ISPF IVP: Review IVP log messages from HOST below :
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Service level 22Feb2011
+FEK900I ISPF IVP: RSE connection and base TSO/ISPF session initializati
on check only
+FEK900I ISPF IVP: *** CHECK : ENVIRONMENT VARIABLES - key variables
displayed below :
+FEK900I ISPF IVP: Server PATH          = ./usr/lpp/java/J6.0/bin:/usr/l
pp/rdz/bin:/usr/lpp/ispf/bin:/bin:/usr/sbin
+FEK900I ISPF IVP: STEPLIB              = NONE
+FEK900I ISPF IVP: Temporary directory = /tmp
+FEK900I ISPF IVP: CGI_ISPHOME          = /usr/lpp/ispf
+FEK900I ISPF IVP: CGI_ISPCONF          = /etc/rdz
+FEK900I ISPF IVP: CGI_ISPWORK          = /var/rdz
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : USS MODULES
+FEK900I ISPF IVP: Checking ISPF Directory : /usr/lpp/ispf
+FEK900I ISPF IVP: Checking modules in /usr/lpp/ispf/bin directory
+FEK900I ISPF IVP: Checking for ISPF configuration file ISPF.conf
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : TSO/ISPF INITIALIZATION
+FEK900I ISPF IVP: ( TSO/ISPF session will be initialized )
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK: Shutting down TSO/ISPF IVP session
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host installation verification completed successfully
+FEK900I ISPF IVP: -----
-----
+FEK901I ISPF IVP Exit code = 0

```

Services prüfen

Die Installation von Developer for System z stellt mehrere Installationsprüfprogramme (IVP, Installation Verification Programs) für die Basisservices und die optionalen Services bereit. Die IVP-Skripts befinden sich im Installationsverzeichnis (Standardverzeichnis: /usr/lpp/rdz/bin/).

Tabelle 19. Installationsprüfprogramme für Services

fekfivpc	„CARMA-Verbindung (optional)“ auf Seite 153
fekfivpd	„RSE-Dämonverbindung“ auf Seite 151
fekfivpi	„Verbindung zum TSO/ISPF-Client-Gateway von ISPF“ auf Seite 152
fekfivpj	„JES Job Monitor-Verbindung“ auf Seite 151
fekfivps	„SCLMDT-Verbindung (optional)“ auf Seite 154
fekfivpt	„TCP/IP konfigurieren“ auf Seite 149

Für die nachfolgend beschriebenen Tasks wird vorausgesetzt, dass Sie in z/OS UNIX aktiv sind. Dies kann durch Ausführen des TSO-Befehls **OMVS** erreicht werden. Zur Rückkehr zu TSO verwenden Sie den Befehl **exit**.

Für die Benutzer-ID, die die Installationsprüfprogramme (Installation Verification Programs, IVPs) ausführt, ist eine große Regionsgröße erforderlich, weil speicherintensive Funktionen wie z. B. Java ausgeführt werden. Sie sollten die Regionsgröße auf 131.072 Kilobyte (128 Megabyte) oder mehr setzen.

Das folgende Fehlerbeispiel ist ein deutliches Anzeichen für eine nicht ausreichende Regionsgröße. Es können jedoch auch andere Fehler auftreten. Zum Beispiel könnte das Starten von Java fehlschlagen.

```
CEE5213S The signal SIGPIPE was received.
%z/OS UNIX command%: command was killed by signal number 13
  %line-number% ** %REXX command%
    +++ RC(137) +++
```

Anmerkung: Die gestarteten Tasks von Developer for System z müssen aktiv sein, bevor der IVP-Test gestartet wird.

Installationsprüfprogramm initialisieren

Bei allen Beispielbefehlen in diesem Abschnitt wird vorausgesetzt, dass bestimmte Umgebungsvariablen gesetzt sind. Wenn das der Fall ist, sind die IVP-Skripts über die Anweisung PATH verfügbar, und die Position der angepassten Konfigurationsdateien ist bekannt. Verwenden Sie die Befehle **pwd** und **cd**, um Ihr aktuelles Verzeichnis zu prüfen und das Verzeichnis mit den angepassten Konfigurationsdateien aufzurufen. Danach können Sie mit dem Shell-Skript **ivpinit** die RSE-Umgebungsvariablen setzen. Sehen Sie sich hierzu das folgende Beispiel an, in dem "\$" die z/OS UNIX-Eingabeaufforderung ist:

```
$ pwd
/u/userid
$ cd /etc/rdz
$ ./ivpinit
RSE configuration files located in /etc/rdz --default
added /usr/lpp/rdz/bin to PATH
```

Der erste Punkt (.) in **./ivpinit** ist ein z/OS UNIX-Befehl zur Ausführung der Shell in der aktuellen Umgebung, damit die in der Shell gesetzten Umgebungsvariablen auch nach dem Beenden der Shell in Kraft bleiben. Der zweite Punkt (.) bezieht sich auf das aktuelle Verzeichnis.

Anmerkung:

- Wenn `./ivpinit` nicht vor den `fekfivp*`-Scripts ausgeführt wird, muss der Pfad zu diesen Scripts angegeben werden, wenn sie aufgerufen werden. Sehen Sie sich dazu das folgende Beispiel an:

```
/usr/lpp/rdz/bin/fekfivpr 512 USERID
```

Wenn `./ivpinit` nicht zuerst ausgeführt wird, fordern alle `fekfivp*`-Scripts außerdem die Position der angepassten Datei `rsed.envvars` an.

- Einige IVP-Tests verwenden die TCP/IP-REXX-Socket-API, die erfordert, dass die TCP/IP-Ladebibliothek (standardmäßig `TCPIP.SEZALOAD`) in der `LINKLIST` oder in `STEPLIB` enthalten ist. Für die Ausführung solcher IVP-Tests kann der folgende Befehl erforderlich sein:

```
$ EXPORT STEPLIB=$STEPLIB:TCPIP.SEZALOAD
```

Wenn zu einer vorhandenen `STEPLIB` eine Bibliothek ohne APF-Berechtigung hinzugefügt wird, werden die APF-Berechtigungen der vorhandenen `STEPLIB`-Dateien entfernt.

Wenn sich `CEE.SCEELKED` in `LINKLIST` oder `STEPLIB` befindet, muss `TCPIP.SEZALOAD` vor `CEE.SCEELKED` eingefügt werden. Andernfalls wird für die TCP/IP-REXX-Socketaufrufe ein Systemabbruch 0C1 ausgegeben.

Informationen zur Diagnose von Verbindungsproblemen mit dem Hostsystem finden Sie in "Konfigurationsprobleme lösen" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02) sowie in den technischen Hinweisen im Supportabschnitt der Developer for System z-Website (<http://www-03.ibm.com/software/products/us/en/developerforsystemz/>).

Portverfügbarkeit

Die Portverfügbarkeit für JES Job Monitor und den RSE-Dämon können Sie durch Absetzen des Befehls **netstat** prüfen. Das Ergebnis sollte die von diesen Services verwendeten Ports wie in den folgenden Beispielen zeigen:

IPv4

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy      TCPIP Name: TCPIP      13:57:36
User Id Conn      Local Socket      Foreign Socket      State
-----
RSED      0000004B 0.0.0.0..4035     0.0.0.0..0         Listen
JMON      00000037 0.0.0.0..6715     0.0.0.0..0         Listen
```

IPv6

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy      TCPIP Name: TCPIP      14:03:35
User Id Conn      State
-----
RSED      0000004B Listen
Local Socket: 0.0.0.0..4035
Foreign Socket: 0.0.0.0..0
JMON      00000037 Listen
Local Socket: 0.0.0.0..6715
Foreign Socket: 0.0.0.0..0
```

TCP/IP konfigurieren

Developer for System z ist bei der Initialisierung davon abhängig, dass TCP/IP mit dem richtigen Hostnamen konfiguriert ist. Dies impliziert, dass die verschiedenen TCP/IP- und Resolver-Konfigurationsdateien ordnungsgemäß definiert sein müssen. Weitere Informationen zur TCP/IP- und Resolver-Konfiguration finden Sie in

"TCP/IP konfigurieren" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02). Führen Sie den folgenden Befehl aus, um die aktuellen Einstellungen zu überprüfen:

```
fekfivpt
```

Anmerkung: Dieses Installationsprüfverfahren (IVP) setzt den TCP/IP-Befehl **netstat -u** ab, dessen Ausführung möglicherweise durch Ihre Sicherheitssoftware verhindert wird. Informationen dazu finden Sie unter dem Profil `EZB.NETSTAT.mvsname.tcprocname.UP` in der Klasse `SERVAUTH`.

Der Befehl sollte etwa eine Ausgabe wie im folgenden Beispiel zurückgeben:

```
$ fekfivpt
```

```
executed on CDFMVS08 -- Wed Jul  2 13:11:54 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
```

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```
res_init Resolver values:
```

```
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset  = /etc/resolv.conf
Translation Table      = Default
UserId/JobName          = USERID
Caller API              = LE C Sockets
Caller Mode             = EBCDIC
(L) DataSetPrefix      = TCPIP
(L) HostName            = CDFMVS08
(L) TcpIpJobName        = TCPIP
(L) DomainOrigin        = RALEIGH.IBM.COM
(L) NameServer          = 9.42.206.2
                        9.42.206.3
(L) NsPortAddr          = 53           (L) ResolverTimeout      = 10
(L) ResolveVia          = UDP          (L) ResolverUdpRetries    = 1
(*) Options NDots       = 1
(*) SockNoTestStor      =
(*) AlwaysWto           = NO           (L) MessageCase          = MIXED
(*) LookUp              = DNS LOCAL
```

```
res_init Succeeded
```

```
res_init Started: 2008/07/02 13:11:54.755363
```

```
res_init Ended: 2008/07/02 13:11:54.755371
```

```
*****
```

```
MVS TCP/IP NETSTAT CS V1R9          TCPIP Name: TCPIP          13:11:54
```

```
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled
```

```
-----
host IP address:
-----
```

```
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75
```

```
Success, addresses match
```

RSE-Dämonverbindung

Führen Sie den folgenden Befehl aus, um die RSE-Dämonverbindung zu überprüfen.

```
fekfivpd
```

Nach einer Aufforderung zur Kennworteingabe sollte der Befehl eine Ausgabe wie im folgenden Beispiel zurückgeben:

```
$ fekfivpd
```

```
executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
attempting to connect userid USERID using port 4035 ...
```

```
Password:
SSL is disabled
connected
8108
570655399
Success
```

Beim Testen einer für SSL aktivierten Verbindung müssen Sie sicherstellen, dass die Benutzer-ID, über die das IVP ausgeführt wird, Zugriff auf alle erforderlichen Zertifikate hat, einschließlich der CA-Zertifikate, die zum Signieren des Developer for System z-Zertifikats verwendet werden. Die Bedienerbefehlversion dieses IVP, F RSED, APPL=IVP DAEMON,userid, verwendet die SSL-Konfiguration, die für das RSE-Hostsystem erstellt wurde, und ist deshalb weniger fehlerträchtig. Einige häufigere, auf SSL bezogene Fehler werden in der folgenden Liste aufgeführt:

- Stellen Sie sicher, dass die Benutzer-ID, über die das IVP ausgeführt wird, Zugriff auf alle erforderlichen Zertifikate hat, wenn Sie folgende Fehlermeldung erhalten: `gsk_environment_init() failed: Error detected while opening the certificate data base`
- Stellen Sie sicher, dass sich die unterzeichnenden CA-Zertifikate auch in der Schlüsselringdatei befinden, wenn Sie folgende Fehlermeldung erhalten: `gsk_secure_socket_init() failed: Certificate validation error`

JES Job Monitor-Verbindung

Führen Sie den folgenden Befehl aus, um die JES-Job-Monitor-Verbindung zu überprüfen.

```
fekfivpj
```

Der Befehl sollte die Bestätigungsnachricht von JES Job Monitor zurückgeben. Vergleichen Sie hierzu das folgende Beispiel (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpj
```

```
executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
testing JES Job Monitor on port 6715...
```

```
hostName=CDFMVS08
hostAddr=9.42.112.75
IPv4 is supported
Waiting for JES Job Monitor response...
ACKNOWLEDGE01v03
```

Success

Verbindung zum TSO/ISPF-Client-Gateway von ISPF

Überprüfen Sie die Verbindung zum TSO/ISPF Client Gateway von ISPF, indem Sie den folgenden Befehl ausführen:

```
fekfivpi
```

Der Befehl sollte die Ergebnisse der auf das TSO/ISPF-Client-Gateway von ISPF bezogenen Überprüfungen zurückgeben. Dazu gehören Variablen, HFS-Module und das Starten und Stoppen der TSO/ISPF-Sitzung. Die Ausgabe sollte etwa wie im folgenden Beispiel aussehen:

```
$ fekfivpi
```

```
executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----
/etc/rdz/ISPF.conf content:
-----
```

```
ispmllib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
```

```
-----
Host install verification for RSE
Review IVP log messages from HOST below :
-----
```

RSE connection and base TSO/ISPF session initialization check only

*** CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

```
Server PATH          =
/usr/lpp/java/J6.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:
/bin:/usr/sbin:.
```

```
STEPLIB              = FEK.SFEKAUTH:FEK.SFEKLOAD
```

```
CGI_ISPHOME   = /usr/lpp/ispf
CGI_ISPCONF   = /etc/rdz
CGI_ISPWORK   = /var/rdz
```

```
-----
*** CHECK : USS MODULES
Checking ISPF Directory : /usr/lpp/ispf
Checking modules in /usr/lpp/ispf/bin directory
Checking for ISPF configuration file ISPF.conf
RC=0
MSG: SUCCESSFUL
-----
```

```
*** CHECK : TSO/ISPF INITIALIZATION
( TSO/ISPF session will be initialized )
RC=0
MSG: SUCCESSFUL
```

```
-----
*** CHECK: Shutting down TSO/ISPF IVP session
RC=0
MSG: SUCCESSFUL
```

```
-----
Host installation verification completed successfully
-----
```

Anmerkung: Falls eine der ISPF-Überprüfungen fehlschlägt, werden detailliertere Informationen angezeigt.

Der Befehl fekfivpi kann mit den folgenden optionalen, nicht positionsgebundenen Parametern verwendet werden:

-file Der Befehl fekfivpi kann umfangreiche Ausgaben generieren, die Hunderte von Zeilen umfassen können. Der Parameter **-file** sendet diese Ausgabe an eine Datei \$TMPDIR/fekfivpi.log. Dabei ist \$TMPDIR der Wert der Anweisung TMPDIR in rsed.envvars (Standardwert: /tmp).

-debug Der Parameter **-debug** erstellt eine detaillierte Testausgabe. Verwenden Sie diese Option nur auf Anweisung des IBM Support Center.

CARMA-Verbindung (optional)

Überprüfen Sie die Verbindung zu CARMA, indem Sie den folgenden Befehl ausführen:

```
fekfivpc
```

Der Befehl sollte das Ergebnis der auf CARMA bezogenen Prüfungen zurückgeben, wie im folgenden Beispiel gezeigt:

```
$ fekfivpc
```

```
executed on CDFMVS08 -- Fri Aug 20 14:15:46 EDT 2010
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 140484608 ( 134.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
*** /etc/rdz/CRASRV.properties content:
port.start = 5227
port.range = 100
startup.script.name = /usr/lpp/rdz/bin/carma.startup.rex
clist.dsname = *CRASTART
crastart.stub = /usr/lpp/rdz/bin/CRASTART
crastart.configuration.file = /etc/rdz/crastart.endevor.conf
crastart.syslog = Partial
crastart.timeout = 420
```

```
*** Creating /tmp/fekfivpc.log
```

```
*** Verifying CARMA installation...
```

1. Creating CARMA connection (timeout after 60 seconds)
2. Initializing CARMA
3. Retrieving RAM list

The following RAMs were found

00 CA Endevor SCM Unique ID: COM.IBM.CARMA.ENDEVORRAM

```

4. Getting customization data for RAM 00
5. Initializing RAM 00
6. Retrieving Repository Instance List
   Found 6 Repository Instance(s)
7. Terminating RAM 00
8. Terminating CARMA

```

```
*** IVP Successful!!!!
```

Anmerkung: Falls das Installationsprüfverfahren (IVP) fehlschlägt, überprüfen Sie den Inhalt der Datei /tmp/fekfivpc.log. In diesem Protokoll wird die Kommunikation zwischen RSE und CARMA dokumentiert. Es kann hilfreiche Informationen zur Ermittlung der eigentlichen Fehlerursache enthalten.

Der Befehl fekfivpc kann mit den folgenden optionalen, nicht positionsgebundenen Parametern verwendet werden:

-noram

Standardmäßig startet fekfivpc den ersten RAM, der in der VSAM-Datei CRADEF definiert ist. Es kann Umstände geben, in denen der RAM nicht getestet werden soll. Dies kann zum Beispiel der Fall sein, wenn ein RAM eines Fremdanbieters zuerst aufgelistet wird und eine unerwartete Eingabe erfordert. In solchen Fällen können Sie das Startargument -noram verwenden, um die RAM-spezifischen Schritte (Schritte 4 - 7) des Installationsprüfverfahrens zu überspringen.

SCLMDT-Verbindung (optional)

Überprüfen Sie die Verbindung zum SCLM Developer Toolkit, indem Sie den folgenden Befehl ausführen:

```
fekfivps
```

Der Befehl sollte die Ergebnisse der auf SCLM Developer Toolkit bezogenen Überprüfungen zurückgeben. Dazu gehören Variablen, HFS-Module, die REXX-Laufzeit sowie das Starten und Stoppen der TSO/ISPF-Sitzung. Die Ausgabe sollte etwa wie im folgenden Beispiel aussehen:

```
$ fekfivps
```

```

executed on CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
executed by uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

```

```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
-----
/etc/rdz/ISPF.conf content:
-----
```

```

isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
sysproc=ISP.SISPCLIB,FEK.SFEKPROC

```

```

-----
Host install verification for RSE
Review IVP log messages from HOST below :
-----

```

```
*** CHECK : ENVIRONMENT VARIABLES - key variables displayed below :
```



```
Server PATH      = /usr/lpp/java/J6.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:  
/bin:/usr/sbin:.
```

```
STEPLIB          = FEK.SFEKAUTH:FEK.SFEKLOAD
```

```
CGI_ISPHOME      = /usr/lpp/ispf  
CGI_ISPCONF      = /etc/rdz  
CGI_ISPWORK      = /var/rdz  
_SCLMDT_CONF_HOME = /var/rdz/scldmt  
_SCLMDT_WORK_HOME = /var/rdz  
_SCLMDT_TRANTABLE = FEK.#CUST.LSTRANS.FILE
```

```
-----  
*** CHECK : JAVA PATH SETUP VERIFICATION  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
*** CHECK : USS MODULES  
Checking ISPF Directory : /usr/lpp/ispf  
Checking modules in /usr/lpp/ispf/bin directory  
Checking for ISPF configuration file ISPF.conf  
Checking install bin Directory : /usr/lpp/rdz/bin  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
*** CHECK : REXX RUNTIME ENVIRONMENT  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
*** CHECK : TSO/ISPF INITIALIZATION  
( TSO/ISPF session will be initialized )  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
*** CHECK: Shutting down TSO/ISPF IVP session  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
Host installation verification completed successfully  
-----
```

Anmerkung: Falls eine der SCLMDT-Überprüfungen fehlschlägt, werden detaillierte Informationen angezeigt.

Der Befehl fekfivps kann mit den folgenden optionalen, nicht positionsgebundenen Parametern verwendet werden:

-file Der Befehl fekfivps kann umfangreiche Ausgaben generieren, die Hunderte von Zeilen umfassen können. Der Parameter **-file** sendet diese Ausgabe an eine Datei \$TMPDIR/fekfivps.log. Dabei ist \$TMPDIR der Wert der Anweisung TMPDIR in rsed.envvars (Standardwert: /tmp).

-debug

Der Parameter **-debug** erstellt eine detaillierte Testausgabe. Verwenden Sie diese Option nur auf Anweisung des IBM Support Center.

Verbindung zu Integrated Debugger (optional)

Prüfen Sie die Verbindung zu Integrated Debugger, indem Sie den folgenden Befehl ausführen:

```
fekfivpe
```

Der Befehl sollte das Ergebnis der auf Integrated Debugger bezogenen Prüfungen zurückgeben. Die Ausgabe sollte wie im folgenden Beispiel aussehen:

Kapitel 9. Sicherheitsdefinitionen

Passen Sie das Beispielmembere FEKRACF an, das RACF- und z/OS UNIX-Beispielbefehle enthält, und übergeben Sie es, um die Basissicherheitsdefinitionen für Developer for System z zu erstellen.

FEKRACF ist in FEK.#CUST.JCL enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 17.

Weitere Informationen zu RACF-Befehlen finden Sie in der Veröffentlichung *RACF Command Language Reference* (IBM Form SA22-7687).

Anmerkung:

- Bei den Sites, die CA ACF2™ for z/OS verwenden, rufen Sie die Produktseite auf der CA-Unterstützungssite (<https://support.ca.com>) auf und suchen nach dem entsprechenden Dokument (TEC492389) mit Informationen zu Developer for System z. Dieses Dokument enthält Details zu den Sicherheitsbefehlen, die für eine ordnungsgemäße Konfiguration von Developer for System z erforderlich sind.
- Bei den Sites, die CA Top Secret® for z/OS verwenden, rufen Sie die Produktseite auf der CA-Unterstützungssite (<https://support.ca.com>) auf und suchen nach dem entsprechenden Dokument (TEC492091) mit Informationen zu Developer for System z. Dieses Dokument enthält Details zu den Sicherheitsbefehlen, die für eine ordnungsgemäße Konfiguration von Developer for System z erforderlich sind.

In den folgenden Abschnitten sind die erforderlichen Schritte, die optionale Konfiguration und mögliche Alternativen beschrieben.

Voraussetzungen und Prüfliste

Der Sicherheitsadministrator muss die in Tabelle 20 aufgelisteten Werte kennen, um die Sicherheitskonfiguration durchzuführen. Diese Werte wurden in früheren Schritten der Installation und Anpassung von Developer for System z definiert.

Tabelle 20. Variablen der Sicherheitskonfiguration

Beschreibung	<ul style="list-style-type: none">• Standardwert• Entsprechende Quelle	Wert
Übergeordnetes Qualifikationsmerkmal für das Developer for System z-Produkt	<ul style="list-style-type: none">• FEK• SMP/E-Installation	
Übergeordnetes Qualifikationsmerkmal für die Developer for System z-Anpassung	<ul style="list-style-type: none">• FEK.#CUST• FEK.SFEKSAMP(FEKSETUP), wie in „Anpassungskonfiguration“ auf Seite 17 beschrieben	
Name der gestarteten Task für Integrated Debugger	<ul style="list-style-type: none">• DBGMR• FEK.#CUST.PROCLIB(DBGMR), wie in „PROCLIB-Änderungen“ auf Seite 26 beschrieben	

Tabelle 20. Variablen der Sicherheitskonfiguration (Forts.)

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Name der gestarteten Task von JES Job Monitor	<ul style="list-style-type: none"> • JMON • FEK.#CUST.PROCLIB(JMON), wie in „PROCLIB-Änderungen“ auf Seite 26 beschrieben 	
Name der gestarteten Task des RSE-Dämons	<ul style="list-style-type: none"> • RSED • FEK.#CUST.PROCLIB(RSED), wie in „PROCLIB-Änderungen“ auf Seite 26 beschrieben 	
Anwendungs-ID	<ul style="list-style-type: none"> • FEKAPPL • /etc/rdz/rsed.envvars, wie in „Zusätzliche Java-Startparameter mit _RSE_JAVAOPTS definieren“ auf Seite 51 beschrieben 	

Die folgende Liste enthält eine Übersicht über die Aktionen, die zur Durchführung der Basissicherheitskonfiguration von Developer for System z erforderlich sind. Um diese Anforderungen zu erfüllen, können je nach erforderlicher Sicherheitsstufe verschiedene Methoden wie in den folgenden Abschnitten dokumentiert verwendet werden. Informationen zur Sicherheitskonfiguration optionaler Developer for System z-Services finden Sie in den vorherigen Abschnitten.

- „Sicherheitseinstellungen und -klassen aktivieren“
- „OMVS-Segment für Benutzer von Developer for System z definieren“ auf Seite 159
- „Gestartete Tasks für Developer for System z definieren“ auf Seite 160
- „RSE als sicheren z/OS UNIX-Server definieren“ auf Seite 161
- „Programmgesteuerte MVS-Bibliotheken für RSE definieren“ auf Seite 162
- „PassTicket-Unterstützung für RSE definieren“ auf Seite 163
- „Anwendungsschutz für RSE definieren“ auf Seite 164
- „JES-Befehlssicherheit definieren“ auf Seite 164
- „Dateiprofile definieren“ auf Seite 166
- „Programmgesteuerte z/OS UNIX-Dateien für RSE definieren“ auf Seite 170
- „Sicherheitseinstellungen prüfen“ auf Seite 171

Sicherheitseinstellungen und -klassen aktivieren

Developer for System z verwendet eine Reihe von Sicherheitsmechanismen, um für den Client eine geschützte und kontrollierte Hostsystemumgebung bereitzustellen. Zu diesem Zweck müssen mehrere Klassen und Sicherheitseinstellungen aktiv sein, wie in den folgenden RACF-Beispielbefehlen gezeigt:

- Anzeige der aktuellen Einstellungen
 - SETROPTS LIST
- Aktivieren der Funktionsklasse für z/OS UNIX- und digitale Zertifikatsprofile
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- Aktivieren der Definitionen für gestartete Tasks
 - SETROPTS GENERIC(STARTED)

- RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
- SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- Aktivieren der Konsolsicherheit für JES Job Monitor
 - SETROPTS GENERIC(CONSOLE)
 - SETROPTS CLASSACT(CONSOLE) RACLIST(CONSOLE)
- Aktivieren des Bedienerbefehlsschutzes für JES Job Monitor
 - SETROPTS GENERIC(OPERCMDS)
 - SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)
- Aktivieren des Anwendungsschutzes für RSE
 - SETROPTS GENERIC(APPL)
 - SETROPTS CLASSACT(APPL) RACLIST(APPL)
- Aktivieren der geschützten Anmeldung unter Verwendung von PassTickets für RSE
 - SETROPTS GENERIC(PTKTDATA)
 - SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
- Aktivieren der Programmsteuerung, um sicherzustellen, dass RSE nur gesicherten Code laden kann
 - RDEFINE PROGRAM ** ADDMEM('SYS1.CMDLIB'//NOPADCHK) UACC(READ)
 - SETROPTS WHEN(PROGRAM)

Anmerkung: Wenn die Klasse PROGRAM bereits ein Profil * enthält, sollten Sie das Profil ** nicht erstellen. Dadurch wird der von der Sicherheitssoftware verwendete Suchpfad unbestimmt und kompliziert. Führen Sie in einem solchen Fall die vorhandenen Definitionen aus dem Profil * mit den neuen Definitionen des Profils ** zusammen. Verwenden Sie das Profil **, wie in *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683) dokumentiert.

Achtung: Wenn "WHEN PROGRAM" aktiv ist, müssen einige Produkte (beispielsweise FTP) programmgesteuert sein. Testen Sie diese Programmsteuerung, bevor Sie sie auf einem Produktionssystem aktivieren.

- (Optional) Unterstützung für X.509-HostIdMappings und erweiterten Eingangsport (POE) aktivieren
 - SETROPTS GENERIC(SERVAUTH)
 - SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)

OMVS-Segment für Benutzer von Developer for System z definieren

Für jeden Benutzer von Developer for System z muss ein RACF-OMVS-Segment oder ein funktional entsprechendes Element definiert werden, das eine gültige z/OS UNIX-Benutzer-ID (UID, ungleich null) angibt. Darüber hinaus müssen für jeden Benutzer ein Ausgangsverzeichnis und ein Shellbefehl definiert werden. Für die Standardgruppe jedes Benutzers ist ebenfalls ein OMVS-Segment mit einer Gruppen-ID erforderlich.

Bei Verwendung der optionalen Komponente 'Integrated Debugger' ist für die Benutzer-ID zu der Anwendung, bei der Fehler behoben werden soll, sowie die zugehörige Standardgruppe ebenfalls ein gültiges RACF OMVS-Segment oder ein Äquivalent erforderlich.

Ersetzen Sie in den folgenden RACF-Beispielbefehlen die Platzhalter #userid, #user-identifizier, #group-name und #group-identifizier durch tatsächliche Werte:

- ALTUSER #userid
OMVS(UID(#user-identifizier) HOME(/u/#userid) PROGRAM(/bin/sh) NOASSIZEMAX)
- ALTGROUP #group-name OMVS(GID(#group-identifizier))

Gestartete Tasks für Developer for System z definieren

Die folgenden RACF-Beispielbefehle erstellen die gestarteten Tasks DBGMR, JMON und RSED mit der ihnen jeweils zugeordneten geschützten Benutzer-ID (STCDBGM, STCJMON und STCRSE) und der Gruppe STCGROUP. Ersetzen Sie die Platzhalter #group-id und #user-id-* durch gültige OMVS-IDs.

- ADDGROUP STCGROUP OMVS(GID(#group-id))
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
- ADDUSER STCDBM DFLTGRP(STCGROUP) NOPASSWORD
NAME('RDZ - DEBUG MANAGER')
OMVS(UID(#user-id-debug) HOME(/tmp) PROGRAM(/bin/sh))
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCJMON DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - JES JOBMONITOR')
OMVS(UID(#user-id-jmon) HOME(/tmp) PROGRAM(/bin/sh))
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCRSE DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - RSE DAEMON')
OMVS(UID(#user-id-rse) HOME(/tmp) PROGRAM(/bin/sh) ASSIZEMAX(2147483647)
)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE STARTED DBGMR.* DATA('RDZ - DEBUG MANAGER')
STDATA(USER(STCDBM) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED JMON.* DATA('RDZ - JES JOBMONITOR')
STDATA(USER(STCJMON) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED RSED.* DATA('RDZ - RSE DAEMON')
STDATA(USER(STCRSE) GROUP(STCGROUP) TRUSTED(NO))
- SETROPTS RACLIST(STARTED) REFRESH

Anmerkung:

- Stellen Sie sicher, dass die Benutzer-IDs der gestarteten Tasks durch Angabe des Schlüsselworts NOPASSWORD geschützt sind.
- Stellen Sie sicher, dass der RSE-Server eine eindeutige OMVS-Benutzer-ID besitzt, denn dieser Benutzer-ID werden Zugriffsrechte für z/OS UNIX gewährt.
- Der RSE-Dämon benötigt für den ordnungsgemäßen Betrieb einen großen Adressraum (2 GB). Legen Sie diesen Wert in der Variablen ASSIZEMAX des OMVS-Segments für die Benutzer-ID STCRSE fest. Die Einstellung dieses Werts stellt sicher, dass der RSE-Dämon unabhängig von Änderungen an MAXASSIZE in SYS1.PARMLIB(BPXPRMxx) die erforderliche Regionsgröße erhält.
- Für den ordnungsgemäßen Betrieb von RSE ist außerdem eine große Anzahl von Threads erforderlich. Sie können einen Grenzwert in der Variable THREADSMAX des OMVS-Segments für die Benutzer-ID STCRSE festlegen. Die Einstellung des

Grenzwerts stellt sicher, dass für RSE unabhängig von Änderungen an MAXTHREADS oder MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) der erforderliche Grenzwert für Threads festgelegt ist. Informationen zur Ermittlung des korrekten Werts für die Threadbegrenzung finden Sie in "Optimierungsaspekte" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).

- Für die Benutzer-ID STCJMON ist es ebenfalls sinnvoll, THREADSMAX im OMVS-Segment festzulegen, da JES Job Monitor für jede Clientverbindung einen Thread verwendet.
- Die gestartete Task von Integrated Debugger (DBGMGR) wird nur von der optionalen Funktion 'Integrated Debugger' verwendet.

Ziehen Sie in Betracht, die Benutzer-ID STCRSE einzuschränken. Benutzer mit dem Attribut RESTRICTED können nicht auf geschützte Ressourcen (MVS) zugreifen, für deren Zugriff sie nicht ausdrücklich berechtigt wurden.

ALTUSER STCRSE RESTRICTED

Um sicherzustellen, dass eingeschränkte Benutzer nicht über die 'anderen' Zugriffsbits Zugriff auf z/OS UNIX-Dateisystemressourcen erlangen, definieren Sie das Profil RESTRICTED.FILESYS.ACCESS in der Klasse UNIXPRIV mit UACC(NONE). Weitere Informationen zur Einschränkung von Benutzer-IDs finden Sie in *Security Server RACF Security Administrator's Guide* (SA22-7683).

Achtung: Wenn Sie Einschränkungen für Benutzer-IDs festgelegt haben, fügen Sie die Berechtigung für den Zugriff auf eine Ressource explizit mit dem TSO-Befehl **PERMIT** oder dem z/OS UNIX-Befehl **setfac1** hinzu. Zu den Ressourcen gehören solche, für die die Dokumentation von Developer for System z UACC verwendet, wie das Profil ** in der Klasse PROGRAM, oder für die allgemeine z/OS UNIX-Konventionen gelten, beispielsweise, dass jeder Lese- und Ausführungsberechtigung für Java-Bibliotheken besitzt. Testen Sie den Zugriff, bevor Sie ihn auf einem Produktionssystem aktivieren.

RSE als sicheren z/OS UNIX-Server definieren

RSE benötigt die Zugriffsberechtigung UPDATE für das Profil BPX.SERVER, um die Sicherheitsumgebung für den Client-Thread erstellen oder löschen zu können. Wenn dieses Profil nicht definiert ist, muss für RSE UID(0) verwendet werden. Dieser Schritt ist erforderlich, damit Clients die Verbindung herstellen können.

Integrated Debugger benötigt die Zugriffsberechtigung UPDATE für das Profil BPX.SERVER, um die Sicherheitsumgebung für den Debug-Thread erstellen oder löschen zu können. Wenn dieses Profil nicht definiert ist, muss UID(0) für die Benutzer-ID der gestarteten STCDBM-Task verwendet werden. Diese Berechtigung ist nur bei Verwendung der optionalen Funktion 'Integrated Debugger' erforderlich.

- RDEFINE FACILITY BPX.SERVER UACC(NONE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCRSE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCDBM)
- SETROPTS RACLIST(FACILITY) REFRESH

Achtung: Mit dem Definieren des Profils BPX.SERVER wechselt z/OS UNIX vollständig von der Sicherheit auf UNIX-Ebene zur Sicherheit auf z/OS UNIX-Ebene, die bedeutend sicherer ist. Möglicherweise hat dieser Wechsel Auswirkungen auf andere z/OS UNIX-Anwendungen und -Operationen. Testen Sie die Sicherheit, bevor Sie sie auf einem Produktionssystem aktivieren. Weitere Informationen zu den verschiedenen Sicherheitsstufen finden Sie in der Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Programmgesteuerte MVS-Bibliotheken für RSE definieren

Server mit der Berechtigung für BPX.SERVER müssen in einer sauberen, programm-gesteuerten Umgebung ausgeführt werden. Diese Voraussetzung impliziert, dass alle von RSE aufgerufenen Programme ebenfalls programmgesteuert sein müssen. Die Programmsteuerung von MVS-Ladebibliotheken wird von Ihrer Sicherheitssoftware verwaltet. Dieser Schritt ist erforderlich, damit Clients die Verbindung herstellen können.

RSE verwendet die Systembibliothek (SYS1.LINKLIB), die Bibliothek der Laufzeit von Language Environment (CEE.SCEERUN*) und die Ladebibliothek des TSO/ISPF-Client-Gateways von ISPF (ISP.SISPLoad).

- RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('ISP.SISPLoad'//NOPADCHK)
- SETROPTS WHEN(PROGRAM) REFRESH

Anmerkung: Wenn die Klasse PROGRAM bereits ein Profil * enthält, sollten Sie das Profil ** nicht verwenden. Durch das Profil wird der von Ihrer Sicherheitssoftware verwendete Suchpfad unbestimmt und kompliziert. Führen Sie in einem solchen Fall die vorhandenen Definitionen aus dem Profil * mit den neuen Definitionen des Profils ** zusammen. Verwenden Sie das Profil **, wie in *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683) dokumentiert.

Zur Unterstützung optionaler Services müssen die folgenden zusätzlich vorausgesetzten Bibliotheken programmgesteuert sein. Diese Liste enthält keine Dateien, die für ein Produkt spezifisch sind, mit dem Developer for System z interagiert, wie beispielsweise IBM File Manager.

- Alternative REXX-Laufzeitbibliothek für SCLM Developer Toolkit
 - REXX.*.SEAGALT
- Systemladebibliothek für SSL-Verschlüsselung
 - SYS1.SIEALNKE
- Developer for System z-Bibliothek für Integrated Debugger
 - FEK.SFEKAUTH

Anmerkung: Bibliotheken, die in den Link-Pack-Bereich (LPA) gestellt werden müssen, erfordern Programmsteuerberechtigungen, wenn für den Zugriff auf diese Bibliotheken LINKLIST oder STEPLIB verwendet wird. In dieser Veröffentlichung ist die Verwendung der folgenden LPA-Bibliotheken dokumentiert:

- ISPF für das TSO/ISPF-Client-Gateway
 - ISP.SISPLPA
- REXX-Laufzeitbibliothek für SCLM Developer Toolkit

- REXX.*.SEAGLPA
- Developer for System z für CARMA
 - FEK.SFEKLPA

PassTicket-Unterstützung für RSE definieren

Das Kennwort des Clients bzw. ein anderes Identifikationsmechanismus, wie ein X.509-Zertifikat, wird nur verwendet, um die Identität beim Herstellen der Verbindung zu überprüfen. Anschließend wird die Threadsicherheit mit PassTickets verwaltet. Dieser Schritt ist erforderlich, damit Clients die Verbindung herstellen können.

PassTickets sind vom System generierte Kennwörter mit einer Lebensdauer von ca. 10 Minuten. Die generierten PassTickets basieren auf einem geheimen Schlüssel. Dieser Schlüssel ist eine 64-Bit-Zahl (16 Hexadezimalzeichen). Ersetzen Sie in den folgenden RACF-Beispielbefehlen den Platzhalter `key16` durch eine benutzerdefinierte 16 Zeichen lange Hexadezimalzeichenfolge aus den Zeichen 0-9 und A-F.

- ```

RDEFINE PTKTDATA FEKAPPL UACC(NONE) SSIGNON(KEYMASKED(key16))
APPLDATA('NO REPLAY PROTECTION – DO NOT CHANGE')
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```

RDEFINE PTKTDATA IRRPTAUTH.FEKAPPL.* UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```

PERMIT IRRPTAUTH.FEKAPPL.* CLASS(PTKTDATA) ACCESS(UPDATE) ID(STCRSE)
```
- ```

SETROPTS RACLIST(PTKTDATA) REFRESH
```

RSE unterstützt die Verwendung von anderen Anwendungs-IDs als FEKAPPL. Entfernen Sie in `rsed.envvars` das Kommentarzeichen vor der Option `APPLID=FEKAPPL` und passen Sie sie an, um diese Funktion zu aktivieren, wie in „Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren“ auf Seite 51 dokumentiert. Die Definitionen der Klasse PTKTDATA müssen mit der eigentlichen, von RSE verwendeten Anwendungs-ID übereinstimmen.

Sie sollten OMVSAPPL nicht als Anwendungs-ID verwenden, da diese ID den geheimen Schlüssel für die meisten z/OS UNIX-Anwendungen offenlegt. Ebenso wenig sollten Sie die MVS-Standardanwendungs-ID (MVS gefolgt von der SMF-ID des Systems) verwenden, da diese ID den geheimen Schlüssel für die meisten MVS-Anwendungen, einschließlich Benutzer-Batch-Jobs, offenlegt.

Anmerkung:

- Wenn die Klasse PTKTDATA bereits definiert ist, überprüfen Sie, ob diese als eine generische Klasse definiert ist, bevor Sie die oben aufgeführten Profile erstellen. Ab z/OS Release 1.7 werden mit der Einführung der Java-Schnittstelle zu PassTickets generische Zeichen in der Klasse PTKTDATA unterstützt.
- Ersetzen Sie den Platzhalter (*) in der Definition IRRPTAUTH.FEKAPPL.* durch eine gültige Maske der Benutzer-ID, um die Benutzer-IDs einzuschränken, für die RSE ein PassTicket generieren kann.
- Abhängig von Ihren RACF-Einstellungen steht der Benutzer, der ein Profil definiert, möglicherweise auch auf der Zugriffsliste des Profils. Entfernen Sie diese Berechtigung für die PTKTDATA-Profile.

- Damit JES Job Monitor die vom RSE angegebenen PassTickets überprüfen kann, müssen JES Job Monitor und RSE dieselbe Anwendungs-ID besitzen. Für JES Job Monitor wird die Anwendungs-ID in der Konfigurationsdatei FEJJCNFG mit der Anweisung APPLID festgelegt.
- Wenn Sie auf Ihrem System ein Verschlüsselungsprodukt installiert haben und dieses verfügbar ist, kann der Anwendungsschlüssel zur sicheren Anmeldung für einen zusätzlichen Schutz verschlüsselt werden. Verwenden Sie dazu das Schlüsselwort KEYENCRYPTED anstelle von KEYMASKED. Weitere Informationen finden Sie in *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Achtung: Die Clientverbindungsanforderung schlägt fehl, wenn PassTickets nicht richtig konfiguriert sind.

Anwendungsschutz für RSE definieren

Während der Clientanmeldung prüft der RSE-Dämon, ob ein Benutzer die Anwendung verwenden darf.

- RDEFINE APPL FEKAPPL UACC(READ) DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- SETROPTS RACLIST(APPL) REFRESH

Anmerkung:

- RSE unterstützt die Verwendung von anderen Anwendungs-IDs als FEKAPPL. Ausführlichere Informationen dazu finden Sie in „PassTicket-Unterstützung für RSE definieren“ auf Seite 163. Die Klassendefinition APPL muss mit der eigentlichen, von RSE verwendeten Anwendungs-ID übereinstimmen.
- Die Clientverbindungsanforderung ist erfolgreich, wenn die Anwendungs-ID nicht in der Klasse APPL definiert ist.
- Die Clientverbindungsanforderung schlägt nur fehl, wenn die Anwendungs-ID definiert ist und der Benutzer keinen Lesezugriff auf das Profil hat.

JES-Befehlssicherheit definieren

JES Job Monitor setzt alle von einem Benutzer angeforderten JES-Bedienerbefehle über eine erweiterte MCS-Knsole (EMCS) ab, deren Name durch die Anweisung `CONSOLE_NAME` gesteuert wird, wie im Abschnitt „FEJJCNFG - Konfigurationsdatei für JES Job Monitor“ auf Seite 34 dokumentiert.

Die folgenden RACF-Beispielbefehl geben Developer for System z-Benutzern bedingten Zugriff auf eine eingeschränkte Gruppe von JES-Befehlen: Hold, Release, Cancel und Purge. Benutzer haben die Ausführungsberechtigung nur, wenn sie die Befehle über JES Job Monitor absetzen. Ersetzen Sie den Platzhalter `#console` durch den aktuellen Konsolennamen.

- RDEFINE OPERCMDS MVS.MCSOPER.#console UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE OPERCMDS JES%.** UACC(NONE)

- PERMIT JES%.** CLASS(OPERCMDS) ACCESS(UPDATE) WHEN(CONSOLE(JMON)) ID(*)
- SETROPTS RACLIST(OPERCMDS) REFRESH

Anmerkung:

- Wenn kein Profil MVS.MCSOPER.#console definiert ist, wird die Verwendung der Konsole zugelassen.
- Damit WHEN(CONSOLE(JMON)) funktioniert, muss die Klasse CONSOLE aktiviert sein. In der Klasse CONSOLE für EMCS-Konsolen ist jedoch keine aktuelle Profilüberprüfung vorhanden.
- Ersetzen Sie JMON nicht durch den aktuellen Konsolennamen in der Klausel WHEN(CONSOLE(JMON)). Das Schlüsselwort JMON stellt die Eingangsportanwendung und nicht den Konsolennamen dar.

Achtung: Wenn Sie in Ihrer Sicherheitssoftware die JES-Befehle mit dem universellen Zugriffsrecht NONE definieren, kann sich das negativ auf andere Anwendungen und Operationen auswirken. Testen Sie die Sicherheit, bevor Sie sie auf einem Produktionssystem aktivieren.

In Tabelle 21 und Tabelle 22 sehen Sie die Bedienerbefehle, die für JES2 und JES3 abgesetzt werden, sowie die eigenständigen Sicherheitsprofile zu deren Schutz.

Tabelle 21. Bedienerbefehle von JES2 Job Monitor

Aktion	Befehl	OPERCMDS-Profil	Erforderlicher Zugriff
Hold	\$Hx(jobid) x = {J, S oder T}	jesname.MODIFYHOLD.BAT jesname.MODIFYHOLD.STC jesname.MODIFYHOLD.TSU	UPDATE
Release	\$Ax(jobid) x = {J, S oder T}	jesname.MODIFYRELEASE.BAT jesname.MODIFYRELEASE.STC jesname.MODIFYRELEASE.TSU	UPDATE
Cancel	\$Cx(jobid) x = {J, S oder T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE
Purge	\$Cx(jobid),P x = {J, S oder T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE

Tabelle 22. Bedienerbefehle von JES3 Job Monitor

Aktion	Befehl	OPERCMDS-Profil	Erforderlicher Zugriff
Hold	*F,J=jobid,H	jesname.MODIFY.JOB	UPDATE
Release	*F,J=jobid,R	jesname.MODIFY.JOB	UPDATE
Cancel	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE
Purge	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE

Anmerkung:

- Die JES-Bedienerbefehle 'Hold', 'Release', 'Cancel' und 'Purge' sowie der Befehl 'Show JCL' können nur für Spooldateien abgesetzt werden, deren Eigner die Client-Benutzer-ID ist, sofern nicht in der Konfigurationsdatei für JES Job Monitor für LIMIT_COMMANDS= der Wert LIMITED oder NOLIMIT angegeben ist. Weitere Informationen finden Sie in "Aktionen für Beschränkungen der Jobziele" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Benutzer können jede Spooldatei anzeigen, sofern in der Konfigurationsdatei für JES Job Monitor nicht LIMIT_VIEW=USERID definiert ist. Weitere Informationen finden Sie in "Zugriff auf Spooldateien" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Auch Benutzer, die nicht berechtigt sind, diese Bedienerbefehle auszuführen, können mit JES Job Monitor Jobs übergeben und Jobausgaben lesen, sofern sie über eine ausreichende Berechtigung für mögliche Profile verfügen, die diese Ressourcen schützen, z. B. diejenigen in den Klassen JESINPUT, JESJOBS und JESSPOOL).

Ihre Sicherheitssoftware verhindert, dass ein Benutzer in einer TSO-Sitzung eine Konsole JMON erstellt, weil er sich so als JES Job Monitor Server ausgeben könnte. Auch wenn die Konsole erstellt werden kann, unterscheidet sich der Eingangspunkt (z. B. JES Job Monitor oder TSO). Von dieser Konsole abgesetzte JES-Befehle werden jedoch nicht die Sicherheitsprüfung bestehen, wenn Ihre Sicherheitssoftware wie in dieser Veröffentlichung beschrieben konfiguriert ist und der Benutzer nicht autorisiert ist, JES-Befehle über andere Mechanismen zu verwenden.

Dateiprofile definieren

Für die meisten Dateien (Datasets) von Developer for System z reicht das Zugriffsrecht READ für Benutzer und ALTER für Systemprogrammierer aus. Ersetzen Sie den Platzhalter #sysprog durch gültige Benutzer-IDs oder RACF-Gruppennamen. Fragen Sie außerdem den Systemprogrammierer, der das Produkt installiert und konfiguriert hat, nach den korrekten Dateinamen. Das bei der Installation verwendete übergeordnete Standardqualifikationsmerkmal ist FEK. Das übergeordnete Standardqualifikationsmerkmal für Dateien, die während des Anpassungsprozesses erstellt werden, ist FEK.#CUST.

- ```
ADDGROUP (FEK) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
```
- ```
ADDSD 'FEK.*.**' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
PERMIT 'FEK.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
```
- ```
SETROPTS GENERIC(DATASET) REFRESH
```

Anmerkung:

- Schützen Sie FEK.SFEKAUTH gegen Aktualisierungen, weil diese Datei eine APF-Berechtigung hat. Dasselbe gilt für FEK.SFEKLOAD und FEK.SFEKLPA, hier jedoch, weil diese Dateien programmgesteuert sind.
- Bei den Beispielbefehlen in dieser Veröffentlichung und im Job FEKRACF wird vorausgesetzt, dass EGN (Enhanced Generic Naming) aktiv ist. Wenn EGN aktiv ist, kann das Qualifikationsmerkmal ** verwendet werden, um eine beliebige Anzahl von Qualifikationsmerkmalen in der Klasse DATASET darzustellen. Erset-

zen Sie ** durch *, wenn EGN auf Ihrem System nicht aktiv ist. Weitere Informationen zu EGN finden Sie in *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Einige der optionalen Komponenten von Developer for System z erfordern zusätzliche Sicherheitsdateiprofile. Ersetzen Sie die Platzhalter #sysprog, #ram-developer und #cicsadmin durch gültige Benutzer-IDs oder RACF-Gruppennamen.

- Wenn die Umsetzung langer/kurzer Namen des SCLM Developer Toolkit verwendet wird, benötigen Benutzer das Zugriffsrecht UPDATE für die Zuordnungs-VSAM FEK.#CUST.LSTRANS.FILE.

```

-
ADDSD 'FEK.#CUST.LSTRANS*.**' UACC(UPDATE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
-
PERMIT 'FEK.#CUST.LSTRANS*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
SETROPTS GENERIC(DATASET) REFRESH

```

- CARMA-RAM-Entwickler (Repository Access Manager) benötigen das Zugriffsrecht UPDATE für die CARMA-VSAMs (FEK.#CUST.CRA*).

```

-
ADDSD 'FEK.#CUST.CRA*.**' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
-
PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
-
SETROPTS GENERIC(DATASET) REFRESH

```

- Wenn der CRD-Server (CICS Resource Definition) von Application Deployment Manager (ADM) verwendet wird, ist für CICS-Administratoren das Zugriffsrecht UPDATE für die VSAM mit dem CRD-Repository erforderlich.

```

-
ADDSD 'FEK.#CUST.ADNREP*.**' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
-
PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
-
SETROPTS GENERIC(DATASET) REFRESH

```

- Wenn das ADM-Manifest-Repository definiert ist, müssen alle Benutzer von CICS Transaction Server das Zugriffsrecht UPDATE auf die VSAM mit dem Manifest-Repository haben.

```

-
ADDSD 'FEK.#CUST.ADNMAN*.**' UACC(UPDATE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
-
PERMIT 'FEK.#CUST.ADNMAN*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
SETROPTS GENERIC(DATASET) REFRESH

```

Verwenden Sie die folgenden RACF-Beispielbefehle für eine besser geschützte Konfiguration, bei der auch die Zugriffsberechtigung READ kontrolliert wird.

- Dateischutz UACC(NONE)

```

-
  ADDGROUP (FEK)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
  OWNER(IBMUSER) SUPGROUP(SYS1)"
-
  ADDSD 'FEK.*.**'      UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.SFEKAUTH' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.SFEKLOAD' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.SFEKLMOD' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.SFEKPROC' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.#CUST.PARMLIB' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.#CUST.CNTL' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.#CUST.SQL' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
  ADDSD 'FEK.#CUST.LSTRANS.*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
-
  ADDSD 'FEK.#CUST.CRA*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
-
  ADDSD 'FEK.#CUST.ADNREP*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
-
  ADDSD 'FEK.#CUST.ADNMAN*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')

```

- Berechtigung für den Systemprogrammierer zur Verwaltung aller Bibliotheken

```

-
  PERMIT 'FEK.*.**      CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
  PERMIT 'FEK.SFEKAUTH CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
  PERMIT 'FEK.SFEKLOAD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
  PERMIT 'FEK.SFEKLMOD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)

```



```

- PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
- PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
- PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
- PERMIT 'FEK.#CUST.SQL' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
- PERMIT 'FEK.#CUST.LSTRANS.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
- PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
- PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
- PERMIT 'FEK.#CUST.ADNMAN*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)

```

- Berechtigung für Clients für den Zugriff auf die Ladebibliotheken und Exec-Bibliotheken

```

- PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(*)
-
- PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(*)
-
- PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(READ) ID(*)
-
- PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(READ) ID(*)
-
- PERMIT 'FEK.#CUST.SQL' CLASS(DATASET) ACCESS(READ) ID(*)

```

Anmerkung: Für FEK.SFEKLPA sind keine Berechtigungen erforderlich, weil der im LPA befindliche Code für alle zugänglich ist.

- Berechtigung für Integrated Debugger für den Zugriff auf die Ladebibliothek.
 - PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCDBM)
- Berechtigung für JES Job Monitor für den Zugriff auf die Lade- und Parameterbibliotheken

```

- PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
-
- PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(READ) ID(STCJMON)

```

- Berechtigung für Clients zum Aktualisieren der VSAM für die Umsetzung langer/kurzer Namen für SCLMDT (optional)

```

- PERMIT 'FEK.#CUST.LSTRANS.*.**' CLASS(DATASET) ACCESS(UPDATE) ID(*)

```

- Berechtigung für RAM-Entwickler zur Aktualisierung der CARMA-VSAMs für CARMA (optional)

```

- PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)

```

- Berechtigung für CICS-Benutzer, die VSAM mit dem CRD-Repository für Application Deployment Manager zu lesen (optional)
—

```
PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(READ) ID(*)
```
- Berechtigung für CICS-Administratoren, die VSAM mit dem CRD-Repository für Application Deployment Manager zu aktualisieren (optional)
—

```
PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
```
- Berechtigung für CICS-Benutzer, die VSAM mit dem Manifest-Repository für Application Deployment Manager zu aktualisieren (optional)
—

```
PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(UPDATE) ID(*)
```
- Berechtigung für CICS TS Server für den Zugriff auf die Ladebibliothek für BIDI und Application Deployment Manager (optional)
—

```
PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
```
- Berechtigung für CICS TS Server, IMS-Regionen und MVS-Batch-Jobs für den Zugriff auf die Ladebibliothek für IRZ-Nachrichten (optional)
—

```
PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#ims)
PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#batch)
```
- Aktivieren von Sicherheitsprofilen
—

```
SETROPTS GENERIC(DATASET) REFRESH
```

Wenn Sie das Zugriffsrecht READ für Systemdateien kontrollieren wollen, müssen Sie Servern und Benutzern von Developer for System z die Berechtigung READ für folgende Dateien erteilen:

- CEE.SCEERUN
- CEE.SCEERUN2
- CBC.SCLBDLL
- ISP.SISPLoad
- ISP.SISPLPA
- SYS1.LINKLIB
- SYS1.SIEALNKE
- SYS1.SIEAMIGE
- REXX.V1R4M0.SEAGLPA

Anmerkung: Wenn Sie die Alternativbibliothek für das REXX-Produktpaket verwenden, ist der Standardname der REXX-Laufzeitbibliothek REXX.*.SEAGALT und nicht REXX.*.SEAGLPA wie im vorherigen Beispiel.

Programmgesteuerte z/OS UNIX-Dateien für RSE definieren

Server mit der Berechtigung für BPX.SERVER müssen in einer sauberen, programm-gesteuerten Umgebung ausgeführt werden. Diese Voraussetzung impliziert, dass alle von RSE aufgerufenen Programme ebenfalls programmgesteuert sein müssen. Die Programmsteuerung für z/OS UNIX-Dateien wird mit dem Befehl **extattr** ver-

waltet. Für die Ausführung dieses Befehls benötigen Sie die Zugriffsberechtigung READ für BPX.FILEATTR.PROGCTL in der Klasse FACILITY oder die UID(0).

Der RSE-Server verwendet die gemeinsam genutzte Java-Bibliothek von RACF (/usr/lib/libIRRRacf*.so).

- extattr +p /usr/lib/libIRRRacf*.so

Anmerkung:

- Ab z/OS 1.9 wird /usr/lib/libIRRRacf*.so während der SMP/E-Installation von RACF im programmgesteuerten Modus installiert.
- Ab z/OS 1.10 ist /usr/lib/libIRRRacf*.so Teil der System Authorization Facility (SAF), die zum z/OS-Basisprodukt gehört. Damit ist die JAR-Datei auch für Kunden verfügbar, die kein RACF verwenden.
- Wenn Sie ein anderes Sicherheitsprodukt als RACF verwenden, kann eine andere Konfiguration erforderlich sein. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Sicherheitsprodukt.
- Bei der SMP/E-Installation von Developer for System z wird das Programmsteuerungsbit für interne RSE-Programme gesetzt.
- Verwenden Sie zum Anzeigen des aktuellen Status des Programmsteuerbits den z/OS UNIX-Befehl **ls -Eog**. Die Datei ist programmgesteuert, wenn der Buchstabe **p** in der zweiten Zeichenfolge angezeigt wird.

```
$ ls -Eog /usr/lib/libIRRRacf*.so
-rwxr-xr-x  aps-  2      69632 Oct  5  2007 /usr/lib/libIRRRacf.so
-rwxr-xr-x  aps-  2      69632 Oct  5  2007 /usr/lib/libIRRRacf64.so
```

Sicherheitseinstellungen prüfen

Verwenden Sie die folgenden Beispielbefehle, um die Ergebnisse Ihrer Anpassungen in Bezug auf die Sicherheit anzuzeigen.

- Sicherheitseinstellungen und -klassen
 - SETROPTS LIST
- OMVS-Segment für Benutzer
 - LISTUSER #userid NORACF OMVS
 - LISTGRP #group-name NORACF OMVS
- Gestartete Tasks
 - LISTGRP STCGROUP OMVS
 - LISTUSER STCDBM OMVS
 - LISTUSER STCJMON OMVS
 - LISTUSER STCRSE OMVS
 - RLIST STARTED DBGMGR.* ALL STDATA
 - RLIST STARTED JMON.* ALL STDATA
 - RLIST STARTED RSED.* ALL STDATA
- RSE als sicherer z/OS UNIX-Server
 - RLIST FACILITY BPX.SERVER ALL
- Programmgesteuerte MVS-Bibliotheken für RSE
 - RLIST PROGRAM ** ALL
- PassTicket-Unterstützung für RSE
 - RLIST PTKTDATA FEKAPPL ALL SSIGNON
 - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.* ALL

- Anwendungsschutz für RSE
 - RLIST APPL FEKAPPL ALL
- JES-Befehlssicherheit
 - RLIST CONSOLE JMON ALL
 - RLIST OPERCMDS MVS.MCSOPER.JMON ALL
 - RLIST OPERCMDS JES%,** ALL
- Dateiprofile
 - LISTGRP FEK
 - LISTDSD PREFIX(FEK) ALL
- Programmgesteuerte z/OS UNIX-Dateien für RSE
 - ls -E /usr/lib/libIRRRacf*.so

Optional können Profile vorhanden sein, die das Verhalten von Developer for System z für einen bestimmten Benutzer steuern. Diese Profile entsprechen dem Filter FEK.** und befinden sich standardmäßig in der Klasse FACILITY. Informationen finden Sie unter der Anweisung `_RSE_FEK_SAF_CLASS` in `rsed.envvars`. Sie können den Befehl **SEARCH** verwenden, um die Profilnamen aufzulisten. Verwenden Sie den Befehl **RLIST**, um die Details für ein Profil anzuzeigen.

- SEARCH CLASS(FACILITY) FILTER(FEK.**)
- RLIST FACILITY #profile-name ALL

Kapitel 10. Leitfaden für die Migration

Aspekte der Migration

In diesem Abschnitt werden die Installations- und Konfigurationsänderungen im Vergleich zu früheren Produktreleases erläutert. Darüber hinaus finden Sie hier allgemeine Richtlinien für die Migration auf dieses Release. Weitere Informationen finden Sie in den entsprechenden Abschnitten dieses Handbuchs.

- Wenn Sie mit einer früheren Version von IBM Rational Developer for System z, IBM WebSphere Developer for System z, IBM WebSphere Developer for zSeries oder IBM WebSphere Studio Enterprise Developer arbeiten, sichern Sie die zugehörigen angepassten Dateien, bevor Sie das Upgrade auf IBM Rational Developer for System z Version 9.0 durchführen.
- Wenn Sie planen, mehrere Instanzen von Developer for System z auszuführen, lesen Sie "Ausführung mehrerer Instanzen" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489-02).
- Wenn sich Ihr Migrationsszenario auf mehr als zwei Releases bezieht, müssen Sie die Anpassung erneut so durchführen, als ob kein älteres Release vorhanden wäre.

Zuvor konfigurierte Dateien sichern

Wenn Sie mit einer früheren Version von Developer for System z gearbeitet haben, sichern Sie die zugehörigen angepassten Dateien, bevor Sie diese Version von IBM Developer for System z installieren.

Anpassbare Developer for System z-Dateien sind an den folgenden Positionen zu finden:

- Version 8.0.1 und 8.5
 - FEK.#CUST.RDZ*.**, Arbeitsdateien des Konfigurationsdienstprogramms
 - FEK.SFEKSAMP, einige Member wurden durch den FEKSETUP-Beispieljob nach FEK.#CUST.* kopiert, wobei '*' PARMLIB, PROCLIB, JCL, CNTL, ASM und COBOL entspricht.
 - FEK.SFEKSAMV
 - /usr/lpp/rdz/samples/, einige Dateien wurden durch den FEKSETUP-Beispieljob nach /etc/rdz/ und /var/rdz/scldmt/* kopiert, wobei '*' CONFIG/, CONFIG/PROJECT/ und CONFIG/script/ entspricht.

Vorherige Developer for System z-Konfigurationen dokumentieren auch Änderungen an Konfigurationsdateien anderer Produkte.

- Version 8.0.1 und 8.5
 - SYS1.PARMLIB(BPXPRMxx)
Legt z/OS UNIX-Systemstandardwerte fest.
 - SYS1.PARMLIB(COMMNDxx)
Startet die Server beim einleitenden Programmladen.
 - SYS1.PARMLIB(LPALSTxx)
Fügt FEK.SFEKLPA zum LPA hinzu.
 - SYS1.PARMLIB(PROGxx)
Fügt FEK.SFEKAUTH und FEK.SFEKLOAD zu LINKLIST hinzu.

- (WLM)
Ordnet eine Anwendungsumgebung für eine gespeicherte DB2-Prozedur zu.

Migrationshinweise für Version 9.0

Die folgenden Migrationshinweise gelten ausschließlich für IBM Rational Developer for System z Version 9.0. Diese Hinweise gelten für eine Migration von IBM Rational Developer for System z Version 9.0.0 auf Version 9.0.1. Sie sind Zusätze zu den vorhandenen Migrationshinweisen für Version 9.0.0.

Alle aufgeführten Änderungen sind ab Version 9.0.1 gültig.

IBM Rational Developer for System z - FMID HHOP900

- CARMA: Die VSAM-Datei CRADEF für den CA Endevor® SCM-RAM wurde aktualisiert.
- CARMA: Das Lademodul CRASTART, das sich im LPA befindet, wurde aktualisiert und erfordert eine LPA-Aktualisierung.
- CARMA: Unterstützung hinzugefügt, um einen Benutzerexit während eines CARMA-Starts auszuführen.
- CARMA: Unterstützung für RAMs hinzugefügt, die Startargumente verarbeiten.
- CARMA: Neue anpassbare Member wurden hinzugefügt:
 - CRAEXIT: CARMA-Beispielbenutzerexit.
 - CRAALLOC: Zuordnungs-Exec für angepasste CARMA-Aufrufe für den RAM.
 - CRACFG: Konfigurationsdatei für die Verwendung des CA Endevor® SCM-RAM
- CARMA: Die folgenden anpassbaren Member wurden geändert:
 - CRASRV.properties
 - crastart.conf
 - crastart.endevor.conf
 - CRASUBMT
 - CRASUBCA
 - CRANDVRA
- CARMA: Es wurden zusätzliche DD-Anweisungen für den CA Endevor® SCM-RAM in "crastart.endevor.conf" und CRASUBCA hinzugefügt:
 - CRAPARM, von CRANDVRA zugeordnet
 - CRACFG
- CARMA: Es wurden zusätzliche DD-Anweisungen für den Nicht-“CA Endevor® SCM-RAM” in crastart.conf und CRASUBMT hinzugefügt:
 - CRAPARM, von CRAALLOC zugeordnet
- Anpassung: Der JCL-Job FEKSETUP verarbeitet jetzt die neuen Member:
 - CRACFG: In FEK.#CUST.PARMLIB(CRACFG) kopiert
 - AQESTC: In FEK.#CUST.PROCLIB(DBGMGR) kopiert
 - AQECSD: In FEK.#CUST.JCL(AQECSD) kopiert
- Integrated Debugger: Neuer optionaler Service
 - Aktualisierungen an den parmlibs IEASVCxx, LPALSTxx und PROGxx (APF und LINKLIST)
 - DBGMGR: JCL der gestarteten Task

- AQECSD: Beispiel-JCL für die Aktualisierung der CICS-Systemdefinition
- AQERACF: Beispiel-JCL für die Ausführung der Sicherheitskonfiguration nur für Integrated Debugger
- RSE: Aktualisierte PROCLIB-Member
 - ELAXFGO
- RSE: Neue optionale Anweisungen wurden in rsecomm.properties hinzugefügt:
 - USER
- RSE: Neue Bedienerbefehle
 - F rsed,APPL=TRACE {USER | SERVER | CLEAR}
- RSE: Neue optionale Anweisungen wurden in rsed.envvars hinzugefügt:
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.timeout
 - (_RSE_JAVAOPTS) -Dkeep.all.logs
 - (_RSE_JAVAOPTS) -Daudit.users
 - RSE_UBLD_DD
 - RSE_UBLD_STEPLIB
- zUnit: Es wurden neue optionale Startargumente hinzugefügt:
 - CLOCALE / -l

Anmerkung: Um die Migration von einer vorhandenen Developer for System z-Konfiguration ohne Integrated Debugger zu erleichtern, wird die Beispiel-JCL FEK.SFEKSAMP(AQERACF) mit RACF-Befehlen bereitgestellt, sodass nur die Sicherheitsdefinitionen für Integrated Debugger definiert werden.

IBM Rational Developer for System z Host Utilities, FMID HAKG900

Es gibt bei Version 9.0 keine Migrationshinweise, die speziell für dieses Produkt gelten.

Migration von Version 8.5 auf Version 9.0

Diese Hinweise beziehen sich auf eine Migration von einer Basisversion 8.5 auf Version 9.0. Sie enthält Änderungen, die bereits als Teil der Wartung von Version 8.5 dokumentiert sind. Die Änderungen, die Teil des Wartungsdatenstroms, und daher vermutlich bereits implementiert sind, sind mit dem Release gekennzeichnet, in dem sie eingeführt wurden.

IBM Rational Developer for System z - FMID HHOP900

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben daher FEK.* und /usr/lpp/rdz/*.
- CARMA: Die VSAM-Dateien CRADEF und CRASTRS für den CA Endevor® SCM-RAM müssen aktualisiert werden, damit die neue Unterstützung der anpassbaren CA Endevor® SCM-Batch-Aktionen verwendet wird (ab Version 8.5.1).
- CARMA: Unterstützung hinzugefügt, um einen RAM während der CRADEF VSAM-Erstellung zu inaktivieren (ab 8.5.1).
- CARMA: Unterstützung für nicht absolute Dateireferenzen in CRASRV.properties hinzugefügt (ab 8.5.1).
- CARMA: Neue Beispielmembers sind hinzugekommen:
 - CRABJOBC: Standardjobkarte für CA Endevor® SCM-Batch-Aktionen (ab Version 8.5.1)

- CARMA: Die folgenden anpassbaren Member wurden geändert:
 - CRASRV.properties (ab Version 8.5.1)
 - carma.startup.rex (ab Version 8.5.1)
 - CRA\$VCAD (ab Version 8.5.1)
 - CRA\$VDEF (ab Version 8.5.1)
 - CRABATCA (ab Version 8.5.1)
 - CRABCFG (ab Version 8.5.1)
 - CRANDVRA (ab Version 8.5.1)
- CARMA: Es wurden zusätzliche DD-Anweisungen für den CA Endevor® SCM-RAM in crastart.endevor.conf und CRASUBCA hinzugefügt:
 - CRABJCLO, von CRANDVRA zugeordnet (ab Version 8.5.1)
 - ENHCEDIT, von CRANDVRA zugeordnet CRANDVRA (ab Version 8.5.1)
- Anpassung: Der JCL-Job FEKSETUP verarbeitet jetzt die neuen Member:
 - CRABJOBC: Wird in FEK.#CUST.CNTL(CRABJOBC) kopiert (ab Version 8.5.1)
 - ELAXFSP: Wird in FEK.#CUST.PROCLIB(ELAXFSP) kopiert (ab Version 9.0)
 - ELAXFSQL: Wird in FEK.#CUST.PROCLIB(ELAXFSQL) kopiert (ab Version 9.0)
 - FEKTEP2: Wird in FEK.#CUST.SQL(FEKTEP2) kopiert (ab Version 9.0)
 - FEKTIAD: Wird in FEK.#CUST.SQL(FEKTEP2) kopiert (ab Version 9.0)
- Fault Analyzer-Integration: Unterstützung für FAI wurde eingestellt. Diese Änderung ist mit älteren Clients, die FAI weiterhin nutzen, nicht kompatibel.
- JES Job Monitor - Neue Operatorbefehle wurden der gestarteten Task JMON hinzugefügt:
 - MODIFY USERS (ab Version 8.5.1)
 - MODIFY -T{N | E | I | V} (ab Version 8.5.1)
 - MODIFY -M{N | E | W | I | V} (ab Version 8.5.1)
 - MODIFY TRACE {N | E | I | V} (ab Version 9.0)
 - MODIFY MESSAGE {N | E | W | I | V} (ab Version 9.0)
- JES Job Monitor - Neue optionale Anweisungen wurden FEJJCENFG hinzugefügt:
 - LOOPBACK_ONLY (ab Version 9.0)
- JES Job Monitor - Optionale Anweisungen wurden aus FEJJCENFG entfernt:
 - _BPXK_SETIBMOPT_TRANSPORT (ab Version 9.0)
- Problembestimmung: FEKLOGS JCL unterstützt nun die Angabe mehrerer Benutzer-IDs zum Sammeln von Benutzerprotokollen (ab Version 8.5.1).
- Problembestimmung: FEKLOGS JCL verwendet jetzt DD REFORMAT, um Protokolle zu erfassen, die zur schnelleren Problembestimmung erneut formatiert wurden (ab Version 8.5.1).
- Problembestimmung: Die folgenden anpassbaren Member wurden geändert:
 - FEKLOGS (ab Version 8.5.1)
- RSE: Der gestarteten Task RSED wurden neue Bedienerbefehle hinzugefügt:
 - MODIFY DISPLAY OWNER,DATASET=dataset (ab Version 9.0)
 - MODIFY DEBUG GC,PID=pid (ab Version 9.0)
- RSE: Neue nicht anpassbare Anweisungen wurden in rsed.envvars hinzugefügt:
 - _CMDSERV_BASE_HOME (ab Version 8.5.1)
 - _CMDSERV_CONF_HOME (ab Version 8.5.1)
 - _CMDSERV_WORK_HOME (ab Version 8.5.1)

- RSE_DSN_SFEKLOAD (ab Version 9.0)
- (_RSE_JAVAOPTS) -Dlock.info.timeout (ab Version 9.0)
- (_RSE_JAVAOPTS) -DDSTORE_INITIAL_SIZE (ab Version 9.0)
- (_RSE_JAVAOPTS) -DDSTORE_MAX_FREE (ab Version 9.0)
- RSE: Neue erforderliche Anweisungen wurden in `rsed.envvars` hinzugefügt:
 - RSE_HLQ (ab Version 9.0)
- RSE: Neue optionale Anweisungen wurden in `rsed.envvars` hinzugefügt:
 - (_RSE_JAVAOPTS) -DRSE_DSICALL (ab Version 8.5.1)
 - (_RSE_JAVAOPTS) -DDISABLE_REMOTE_INDEX_SEARCH (ab Version 8.5.1)
 - (_RSE_JAVAOPTS) -DDISABLE_TEXT_SEARCH (ab Version 9.0)
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.hits (ab Version 9.0)
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.datasets (ab Version 9.0)
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.lines (ab Version 9.0)
 - (_RSE_JAVAOPTS) -DDSTORE_SSL_ALGORITHM (ab Version 9.0)
- RSE: Der Standardwert für nicht anpassbare Anweisungen in `rsed.envvars` wurde geändert:
 - (_RSE_JAVAOPTS) -DSPIRIT_EXPIRY_TIME (ab Version 9.0)
- RSE: Der Standardwert für optionale Anweisungen in `rsed.envvars` wurde geändert:
 - (_RSE_JAVAOPTS) -Xms (ab Version 8.5.1)
 - (_RSE_JAVAOPTS) -Xmx (ab Version 8.5.1)
 - (_RSE_JAVAOPTS) -Dmaximum.clients (ab Version 8.5.1)
 - (_RSE_JAVAOPTS) -Dmaximum.threads (ab Version 8.5.1)
 - CGI_ISPPREF (ab Version 9.0)
- Sicherheit: Unterstützung für neue Sicherheitsprofile wurde hinzugefügt:
 - FEK.USR.** (ab Version 8.5.1)

Konfigurierbare Dateien

Tabelle 23 auf Seite 178 enthält eine Übersicht über die Dateien, die in Version 9.0 angepasst werden. Die Beispielbibliotheken von Developer for System z (FEK.SFEKSAMP, FEK.SFEKSAMV und `/usr/lpp/rdz/samples/`) enthalten mehr anpassbare Member, als hier aufgeführt sind, wie zum Beispiel CARMA-Beispielquellcode und Jobs für die Kompilierung dieses Codes. Die Beispielbibliotheken von Developer for System z Host Utilities (AKG.SAKGSAMP und `/usr/lpp/rdzutil/samples`) enthalten mehr anpassbare Member, als hier aufgeführt sind, wie zum Beispiel das Beispielscript für die Nachverarbeitung der Codeüberprüfung.

Die folgenden Member und Dateien sind nicht mehr anpassbar oder werden nicht mehr verwendet:

- LOCKD - gestartete Task
- ELAXMSAM - Beispiel für eine gespeicherte DB2-Prozedur
- ELAXMJCL - Beispiel-JCL für eine gespeicherte DB2-Prozedur

Anmerkung: Der Beispieljob FEKSETUP kopiert alle aufgelisteten Member in verschiedene Dateien (Datasets) und Verzeichnisse (standardmäßig in `FEK.#CUST.*` und `/etc/rdz/*`).

Tabelle 23. Anpassungen in Version 9.0

Member/Datei	Standardposition	Zweck	Migrationshinweise
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für die Erstellung von Dateien und Verzeichnissen und zum Füllen derselben mit anpassbaren Dateien	Aktualisiert, um Aktionen für Dateien zu entfernen, die nicht mehr verwendet werden, und Aktionen für neue Dateien hinzuzufügen
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JCL für JES Job Monitor	Keine
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Name für JMON-Member	Siehe JMON-Member
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	JCL für den RSE-Dämon	Keine
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Name für RSED-Member	Siehe RSED-Member
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL für ferne Projektbuilds usw.	ELAXFSP und ELAXFSQL sind neu, ELAXFCOC und ELAXFCP1 wurden für Unterstützung von Cobol Version 5 aktualisiert.
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für Sicherheitsdefinitionen	Keine
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Konfigurationsdatei für JES Job Monitor	Neue optionale Anweisungen sind hinzugekommen. Vorhandene optionale Anweisungen wurden entfernt.
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL für TSO-Übergabe	Keine
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für CARMA-Nachrichten	Keine
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für CARMA-Konfiguration	Unterstützung für das Ausschließen von RAM wurde hinzugefügt.
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für angepasste CARMA-Informationen	Keine

Tabelle 23. Anpassungen in Version 9.0 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der CARMA-Konfigurationsdatei (VSAM) für den CA Endevor® SCM-RAM	Unterstützung für das Ausschließen von RAM wurde hinzugefügt und VSAM-Eingaben wurden geändert.
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM-Datei für angepasste CARMA-Informationen für den CA Endevor® SCM-RAM	VSAM-Eingabe wurde geändert.
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CLIST für CARMA-Batch-Start	Keine
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CLIST für CARMA-Batchstart für den CA Endevor® SCM-RAM	Keine
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für Batchaktionen für den CA Endevor® SCM-RAM	Neue Anweisungen wurden hinzugefügt.
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA-JCL für Batchaktion für den CA Endevor® SCM-RAM	Unterstützung für variable Jobkarte wurde hinzugefügt.
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für den CA Endevor® SCM-RAM	Keine
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für den CA Endevor® SCM-RAM	Keine
CRANDVRA	FEK.SFEKPROC	CARMA-Zuordnungs-REXX für den CA Endevor® SCM-RAM	Neue DD-Zuordnungen wurden hinzugefügt.

Tabelle 23. Anpassungen in Version 9.0 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für SCLM-RAM-Nachrichten	Keine
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der SCLM-RAM-Dateien	Keine
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für PDS-RAM-Nachrichten	Keine
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Zusammenführen von RAM-Definitionen	Keine
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Extrahieren von RAM-Definitionen	Keine
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Beispielquellcode für die Ersetzung von IRXJCL	Keine
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung von CRAXJCL	Keine
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des RESTful-CRD-Servers für die primäre CICS-Region	Keine
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren von alternativen Transaktions-IDs für die primäre CICS-Region	Keine
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Erstellen von alternativen Transaktions-IDs	Keine
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung von ADNMSGHS	Keine
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Beispielquellcode für den Pipelinenachrichtenhandler	Keine

Tabelle 23. Anpassungen in Version 9.0 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung des CRD-Repositorys	Keine
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des Web-Service-CRD-Servers für die primäre CICS-Region	Keine
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des CRD-Servers für nicht primäre CICS-Regionen	Keine
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Aktualisierung der CRD-Standardwerte	Keine
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung und zum Definieren des Manifestrepositorys	Keine
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	Von ELAXF* verwendete SQL-Befehlseingabe	Neu, Anpassung ist optional.
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	Von ELAXF* verwendete SQL-Befehlseingabe	Neu, Anpassung ist optional.
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL-Prozedur für zUnit	Keine
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	REXX zum Aufrufen des PL/I-Compilers innerhalb des Vorprozessor-Frameworks	Keine
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Erfassen von Protokolldateien	Es wurden zusätzliche Überprüfungen hinzugefügt. Anpassungen an älteren Dateien müssen erneut vorgenommen werden.
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-Umgebungsvariablen	Ältere Kopien müssen durch diese ersetzt und die Anpassungen erneut vorgenommen werden.

Tabelle 23. Anpassungen in Version 9.0 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für TSO/ISPF Client Gateway	Keine
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei	Unterstützung für Standardwerte hinzugefügt
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei für die Verwendung von CRASTART	Keine
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei für die Verwendung von CRASTART für den CA Endevor® SCM-RAM	Keine
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Erzwungene Includes für C/C++-Content-Assist	Keine
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-SSL-Konfigurationsdatei	Keine
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-Tracekonfigurationsdatei	Keine
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Informationen mit Push in die Clientkonfigurationsdatei übertragen	Keine

IBM Rational Developer for System z Host Utilities, FMID HAKG900

Es gibt keine Migrationshinweise, da in Version 8.5 keine entsprechende Funktion vorhanden ist.

Konfigurationsdateien

Tabelle 24 enthält eine Übersicht über die Dateien, die in Version 9.0 angepasst werden. Die Beispielbibliotheken von Developer for System z Host Utilities (AKG.SAKGSAMP und /usr/lpp/rdzutil/samples) enthalten mehr anpassbare Member, als hier aufgeführt sind, wie zum Beispiel ein Beispielscript für die Nachverarbeitung der Codeüberprüfung.

Anmerkung: Der Beispieljob AKGSETUP kopiert alle aufgelisteten Member in verschiedene Dateigruppen und Verzeichnisse (standardmäßig in AKG.#CUST.*).

Tabelle 24. Anpassungen in Host Utilities Version 9.0. Anpassungen in Host Utilities Version 9.0

Member oder Datei	Standardposition	Zweck	Migrationshinweise
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL für die Erstellung von Dateigruppen (Datasets) und zum Füllen dieser Dateigruppen mit anpassbaren Dateien	Neu
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL für die Codeabdeckung	Neu
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL für die Codeüberprüfung	Neu
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL zum Hinzufügen von Code von Drittherstellern zur Codeüberprüfung	Neu

Migrationshinweise für Version 8.5

Die folgenden Migrationshinweise beziehen sich ausschließlich auf die Version 8.5. Diese Hinweise gelten für eine Migration von IBM Rational Developer for System z Version 8.5.0 auf 8.5.1 und sind Zusätze zu den vorhandenen Migrationshinweisen für Version 8.5.0.

Alle nachfolgend aufgeführten Änderungen sind ab Version 8.5.1 gültig.

- CARMA: Die VSAM-Dateien CRADEF und CRASTRS für den CA Endevor® SCM-RAM müssen aktualisiert werden, damit die neue Unterstützung der anpassbaren CA Endevor® SCM-Batch-Aktionen verwendet wird.
- CARMA: Unterstützung hinzugefügt, um einen RAM während der CRADEF VSAM-Erstellung zu inaktivieren.
- CARMA: Unterstützung für nicht absolute Dateireferenzen in CRASRV.properties hinzugefügt.
- CARMA: Neue Beispielmembers sind hinzugekommen:
 - CRABJOBC: Standardjobkarte für CA Endevor® SCM-Batch-Aktionen.
- CARMA: Die folgenden anpassbaren Members wurden geändert:
 - CRASRV.properties
 - carma.startup.rex
 - CRA\$VCAD
 - CRA\$VDEF
 - CRABATCA
 - CRABCFG
 - CRANDVRA
- CARMA: Es wurden zusätzliche DD-Anweisungen für den CA Endevor® SCM-RAM in crastart.endevor.conf und CRASUBCA hinzugefügt:
 - CRABJCLO, von CRANDVRA zugeordnet
 - ENHCEDIT, von CRANDVRA zugeordnet CRANDVRA
- Anpassung: Der JCL-Job FEKSETUP verarbeitet jetzt die neuen Members:
 - CRABJOBC: Wird in FEK.#CUST.CNTL(CRABJOBC) kopiert.
- JES Job Monitor - Neue Operatorbefehle wurden der gestarteten Task JMON hinzugefügt (ab Version 8.0.3.2):
 - MODIFY USERS
 - MODIFY -T{N | E | I | V}
 - MODIFY -M{N | E | W | I | V}
- Problembestimmung: FEKLOGS JCL unterstützt nun die Angabe mehrerer Benutzer-IDs zum Sammeln von Benutzerprotokollen.
- Problembestimmung: FEKLOGS JCL verwendet jetzt DD REFORMAT, um Protokolle zu erfassen, die zur schnelleren Problembestimmung erneut formatiert wurden.
- Problembestimmung: Die folgenden anpassbaren Members wurden geändert:
 - FEKLOGS
- RSE: Neue nicht anpassbare Anweisungen wurden in rsed.envvars hinzugefügt:
 - _CMDSERV_BASE_HOME
 - _CMDSERV_CONF_HOME
 - _CMDSERV_WORK_HOME
- RSE: Neue optionale Anweisungen wurden in rsed.envvars hinzugefügt:

- (_RSE_JAVAOPTS) -DRSE_DSICALL
- (_RSE_JAVAOPTS) -DDISABLE_REMOTE_INDEX_SEARCH
- RSE: Der Standardwert für optionale Anweisungen in rsed.envvars wurde geändert:
 - (_RSE_JAVAOPTS) -Xms
 - (_RSE_JAVAOPTS) -Xmx
 - (_RSE_JAVAOPTS) -Dmaximum.clients
 - (_RSE_JAVAOPTS) -Dmaximum.threads
- RSE: Der Standardwert für nicht anpassbare Anweisungen in rsed.envvars wurde geändert:
 - (_RSE_JAVAOPTS) -DDSTORE_SPIRIT_ON
- Sicherheit: Unterstützung für neue Sicherheitsprofile wurde hinzugefügt:
 - FEK.USR.**

Migration von Version 8.0.1 auf Version 8.5

Diese Hinweise gelten für eine Migration von einer Basisversion 8.0.1 auf Version 8.5. Sie enthält Änderungen, die bereits als Teil der Wartung von Version 8.0.1 dokumentiert sind. Die Änderungen, die Teil des Wartungsdatenstroms, und daher vermutlich bereits implementiert sind, sind mit dem Release gekennzeichnet, in dem sie eingeführt wurden.

IBM Rational Developer for System z - FMID HHOP850

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben daher FEK.* und /usr/lpp/rdz/*.
- CARMA - Das Lademodul CRASTART, das im LPA vorhanden ist, wurde aktualisiert und erfordert eine LPA-Aktualisierung (ab Version 8.0.3.2).
- CARMA - Die CRAMSG-VSAM muss aktualisiert werden (ab Version 8.0.3 und 8.5).
- CARMA - Die VSAM-Dateien CRADEF und CRASTRS für den CA Endevor[®] SCM-RAM müssen aktualisiert werden, damit die neue Unterstützung der CA Endevor[®] SCM-Batch-Aktionen (ab Version 8.0.3) und CA Endevor[®] SCM-Pakete (ab Version 8.0.3) verwendet werden.
- CARMA - Es wurde eine neue VSAM-Eingabe für CRADEF und CRASTRS hinzugefügt, um die Wiederherstellung von CA Endevor[®] SCM-Paketaktionen aus Menüs mit CA Endevor[®] SCM-Elementen zu ermöglichen.
 - CRA0VPKD - Wird mit CRADEF zusammengeführt.
 - CRA0VPKS - Wird mit CRASTRS zusammengeführt.
- CARMA - Neue Beispielmembers sind hinzugekommen (ab Version 8.0.3):
 - CRABCFG - Konfigurationsdatei für CA Endevor[®] SCM-Batch-Aktionen.
 - CRABATCA - Beispieljob für CA Endevor[®] SCM-Batch-Aktionen.
- CARMA - Die folgenden anpassbaren Members wurden geändert (ab Version 8.0.3, 8.0.3.1 und 8.5):
 - CRANDVRA
 - CRASHOW
 - CRASRV.properties
 - CRABCFG
- CARMA - Es wurden zusätzliche DD-Anweisungen für den CA Endevor[®] SCM-RAM in crastart.endevor.conf und CRASUBCA hinzugefügt (ab Version 8.0.3):

- CRABCFG
- CRABSKEL
- PKGSCLS (von CRANDVRA zugeordnet)
- Enterprise Service Tools - IRZ-Ladmodule und -Nachrichtenmodule wurden in eine neue Bibliothek verschoben (ab Version 8.5):
 - FEK.SFEKLMOD(IRZ* IIRZ*)
- File Manager-Integration wurde entfernt (ab Version 8.5). Einige Funktionen, beispielsweise die unformatierte QSAM-Bearbeitung, sind jetzt Bestandteil der regulären Dateihandhabung durch Developer for System z. Für erweiterte Funktionen wie beispielsweise die Bearbeitung formatierter Daten unter Verwendung von Copybooks oder Kopfdaten Dateien ist das Plug-in IBM File Manager für Eclipse erforderlich.
- Include-Vorprozessor - Neue Beispielmembers wurden hinzugefügt (ab Version 8.0.3.1):
 - FEKRNPLI
- Dienstprogramm zur Hostkonfiguration - Eine Migrationsoption wurde hinzugefügt (ab Version 8.0.2)
- JES Job Monitor - Neue Operatorbefehle wurden der gestarteten Task JMON hinzugefügt (ab Version 8.0.3.2):
 - MODIFY STORAGE
- JES Job Monitor - Neue optionale Anweisungen wurden zu FEJJCENFG hinzugefügt (ab Version 8.0.3.1 und 8.0.3.2):
 - LIMIT_CONSOLE
 - SEARCHALL
 - TRACE_STORAGE
- PROCLIB - Die folgenden PROCLIB-Members wurden geändert (ab Version 8.0.3):
 - ELAXFUOP
- RSE - Die Option zum Angeben von TMPDIR als Startargument für die gestarteten Tasks RSED und LOCKD wurde entfernt. Sie wurde durch eine nicht anpassbare Funktion ersetzt, die das Ausgangsverzeichnis für die Benutzer-ID der gestarteten Tasks auf TMPDIR festlegt, wenn /tmp zum Schreiben von Aktionen nicht zur Verfügung steht (ab Version 8.0.3.1).
- RSE - Der gestarteten Task LOCKD wurden neue Bedienerbefehle hinzugefügt (ab Version 8.0.2):
 - MODIFY DISPLAY TABLE
- RSE - Der gestarteten Task RSED wurden neue Bedienerbefehle hinzugefügt (ab Version 8.0.2, 8.0.3 und 8.0.3.2):
 - MODIFY IVP ISPF,userid
 - MODIFY IVP PASSTICKET,userid
 - MODIFY DEBUG HEAPDUMP,PID=pid
 - MODIFY DEBUG JAVACORE,PID=pid
- RSE - Die Bedienerbefehle der gestarteten Task RSED wurden erweitert (ab Version 8.0.2 und 8.0.3.1):
 - MODIFY DISPLAY CLIENT [{,LOGON | ,ID | ,USER}]
 - MODIFY DISPLAY PROCESS,CPU [,PID=pid]
- RSE - Die folgenden Konsolennachrichten sind neu (ab Version 8.0.3 und 8.0.3.1):
 - FEK910I = {0} IVP Exit code = {1}

- FEK211W User, {0}, not logged on
- RSE - Neue nicht anpassbare Anweisungen wurden in "rsed.envvars" hinzugefügt (ab Version 8.0.3):
 - (_RSE_JAVAOPTS) -Dldap.server.address
 - (_RSE_JAVAOPTS) -Dldap.server.port
 - (_RSE_JAVAOPTS) -Dldap.ptc.group.name.suffix
 - _RSE_PTC
- RSE - Neue optionale Anweisungen wurden in "rsed.envvars" hinzugefügt (ab Version 8.0.3, 8.0.3.1 und 8.5):
 - (_RSE_JAVAOPTS) -Daudit.action
 - (_RSE_JAVAOPTS) -Daudit.action.id
 - (_RSE_JAVAOPTS) -Dlogon.action
 - (_RSE_JAVAOPTS) -Dlogon.action.id
 - (_RSE_JAVAOPTS) -Dreject.logon.threshold
 - (_RSE_JAVAOPTS) -Dinclude.c
 - (_RSE_JAVAOPTS) -Dinclude.cpp
 - (_RSE_JAVAOPTS) -DCPP_CLEANUP_INTERVAL
 - (_RSE_JAVAOPTS) -DRIS_BUFFER
 - (_RSE_JAVAOPTS) -DDSTORE_TCP_NO_DELAY
 - _RSE_FEK_SAF_CLASS
 - _RSE_LDAP_SERVER
 - _RSE_LDAP_PORT
 - _RSE_LDAP_PTC_GROUP_SUFFIX
 - CGI_ISPPREF
- RSE - Vorhandene erforderliche Anweisungen wurden umbenannt (ab Version 8.5):
 - _CMDSERV_BASE_HOME -> CGI_ISPHOME
 - _CMDSERV_CONF_HOME -> CGI_ISPCONF
 - _CMDSERV_WORK_HOME -> CGI_ISPWORK
 - _RSE_CMDSERV_OPTS -> _RSE_ISPF_OPTS
- RSE - Vorhandene optionale Anweisungen wurden durch weitere Werte erweitert (ab Version 8.5):
 - STEPLIB
- RSE - Geänderte Interpretation der folgenden optionalen Anweisungen in "rsed.envvars" (ab Version 8.0.3):
 - (_RSE_JAVAOPTS) -Dprocess.cleanup.interval
- RSE - Folgende optionale Konfigurationsdateien sind neu (ab Version 8.5):
 - include.conf
- RSE - Neue optionale Anweisungen wurden in "pushtoclient.properties" hinzugefügt (ab Version 8.0.3):
 - accept.product.license
- RSE - Geänderte Interpretation der folgenden optionalen Anweisungen in "pushtoclient.properties" (ab Version 8.0.3):
 - config.enabled
 - product.enabled
 - reject.config.updates

- reject.product.updates
- RSE - Neue Beispiele für z/OS UNIX wurden hinzugefügt (ab Version 8.0.3 und 8.0.3.1):
 - process_audit.rex
 - process_logon.sh
- Sicherheit - Unterstützung für neue Sicherheitsprofile wurde hinzugefügt (seit Version 8.0.3):
 - FEK.PTC.**
- zUnit - Neue optionale PROCLIB-Member wurden hinzugefügt (ab Version 8.5):
 - AZUZUNIT
- Neue Veröffentlichung *Rational Developer for System z Messages and Codes* (IBM Form SC14-7497).
- Neue Veröffentlichung *Rational Developer for System z - Antworten auf gängige Fragen der Hostkonfiguration und -wartung* (IBM Form SC12-4724).

Konfigurierbare Dateien

Tabelle 25 zeigt einen Überblick über Dateien, die in Version 8.5 angepasst werden. Die Beispielbibliotheken von Developer for System z (FEK.SFEKSAMP, FEK.SFEKSAMV und /usr/lpp/rdz/samples/) enthalten mehr anpassbare Member, als hier aufgeführt sind, wie zum Beispiel CARMA-Beispielquellcode und Jobs für die Kompilierung dieses Codes.

Die folgenden Member und Dateien sind nicht mehr anpassbar oder werden nicht mehr verwendet.

- FMIEXT.properties wird nicht mehr verwendet.

Anmerkung: Der Beispieljob FEKSETUP kopiert alle aufgelisteten Member in verschiedene Dateien und Verzeichnisse (standardmäßig in FEK.#CUST.* und /etc/rdz/*).

Tabelle 25. Anpassungen in Version 8.5

Member/Datei	Standardposition	Zweck	Migrationshinweise
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für die Erstellung von Dateien (Datasets) und Verzeichnissen und zum Füllen derselben mit anpassbaren Dateien	Wurde aktualisiert, um neue anpassbare Member einzuschließen, eine neue Verzeichnisstruktur zu erstellen und Aktionen für Dateien entfernen, die nicht mehr verwendet werden.
JMON	FEK.SFEKSAMP (FEJJJCL) [FEK.#CUST.PROCLIB]	JCL für JES Job Monitor	Keine
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (JMON)]	Name für JMON-Member	Siehe JMON-Member

Tabelle 25. Anpassungen in Version 8.5 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
RSED	FEK.SFEKSAMP (FEKRSED) [FEK.#CUST.PROCLIB]	JCL für den RSE-Dämon	Geänderte TMPDIR-Unterstützung
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (RSED)]	Name für RSED-Member	Siehe RSED-Member
LOCKD	FEK.SFEKSAMP (FEKLOCKD) [FEK.#CUST.PROCLIB]	JCL für Sperrdämon	Geänderte TMPDIR-Unterstützung
FEKLOCKD	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (LOCKD)]	Name für LOCKD-Member	Siehe LOCKD-Member
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL für ferne Projektbuilds usw.	Member ELAXFUOP wurde geändert
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für Sicherheitsdefinitionen	Keine
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Konfigurationsdatei für JES Job Monitor	Neue optionale Anweisungen sind hinzugekommen.
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL für TSO-Übergabe	Keine
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für CARMA-Nachrichten	VSAM-Eingabe wurde geändert.
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für CARMA-Konfiguration	Keine
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für angepasste CARMA-Informationen	Keine
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der CARMA-Konfigurationsdatei (VSAM) für den CA Endevor® SCM-RAM	VSAM-Eingabe wurde geändert.
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM-Datei für angepasste CARMA-Informationen für den CA Endevor® SCM-RAM	VSAM-Eingabe wurde geändert.
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CLIST für CARMA-Batchstart	Keine
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CLIST für CARMA-Batchstart für den CA Endevor® SCM-RAM	Es wurden zusätzliche DD-Anweisungen hinzugefügt.

Tabelle 25. Anpassungen in Version 8.5 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für Batchaktionen für den CA Endeavor® SCM-RAM	NEU, Anpassung optional
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA-JCL für Batchaktion für den CA Endeavor® SCM-RAM	NEU, Anpassung optional
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für den CA Endeavor® SCM-RAM	Es wurden neue Filter hinzugefügt.
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für den CA Endeavor® SCM-RAM	Keine
CRANDVRA	FEK.SFEKPROC	CARMA-Zuordnungs-REXX für den CA Endeavor® SCM-RAM	Es wurden zusätzliche DD-Anweisungen hinzugefügt.
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für SCLM-RAM-Nachrichten	Keine
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der SCLM-RAM-Dateien	Keine
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für PDS-RAM-Nachrichten	Keine
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Zusammenführen von RAM-Definitionen	Keine
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Extrahieren von RAM-Definitionen	Keine
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Beispiel Quellcode für die Ersetzung von IRXJCL	Keine
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung von CRAXJCL	Keine
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des RESTful-CRD-Servers für die primäre CICS-Region	Keine
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren von alternativen Transaktions-IDs für die primäre CICS-Region	Keine
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Erstellen von alternativen Transaktions-IDs	Keine
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung von ADNMSGHS	Keine

Tabelle 25. Anpassungen in Version 8.5 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Beispielquellcode für den Pipelinenachrichtenhandler	Keine
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung des CRD-Repositorys	Keine
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des Web-Service-CRD-Servers für die primäre CICS-Region	Keine
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des CRD-Servers für nicht primäre CICS-Regionen	Keine
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Aktualisierung der CRD-Standardwerte	Keine
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung und zum Definieren des Manifestrepositorys	Keine
ELAXMSAM	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL-Prozedur des WLM-Adressraums	Keine
ELAXMJCL	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des Stored Procedure Builder für COBOL und PL/I für DB2	Keine
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL-Prozedur für zUnit	NEU, Anpassung optional
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	REXX zum Aufrufen des PL/I-Compilers innerhalb des Vorprozessor-Frameworks	NEU, Anpassung optional
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Erfassen von Protokolldateien	Es wurden zusätzliche Überprüfungen hinzugefügt. Anpassungen an älteren Dateien müssen erneut vorgenommen werden.
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-Umgebungsvariablen	Ältere Kopien müssen durch diese ersetzt und die Anpassungen erneut vorgenommen werden.
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für TSO/ISPF Client Gateway	Keine

Tabelle 25. Anpassungen in Version 8.5 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei	Ephemere Ports werden jetzt unterstützt.
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei für die Verwendung von CRASTART	Keine
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei für die Verwendung von CRASTART für den CA Endevor® SCM-RAM	Es wurden zusätzliche DD-Anweisungen hinzugefügt.
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Erzwungene Includes für C/C++-Content-Assist	NEU, Anpassung optional
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-SSL-Konfigurationsdatei	Keine
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-Tracekonfigurationsdatei	Einige Anweisungen sind jetzt optional.
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Informationen mit Push in die Clientkonfigurationsdatei übertragen	Zusätzliche Anweisungen wurden hinzugefügt und vorhandene Anweisungen erweitert.

Kapitel 11. Bedienerbefehle

In diesem Kapitel erhalten Sie einen Überblick über die für Developer for System z verfügbaren Bedienerbefehle bzw. Konsolenbefehle.

Start (S)

Mit dem Befehl **START** können Sie eine gestartete Task (STC) dynamisch starten. Die abgekürzte Fassung des Befehls ist der Buchstabe S.

Integrated Debugger

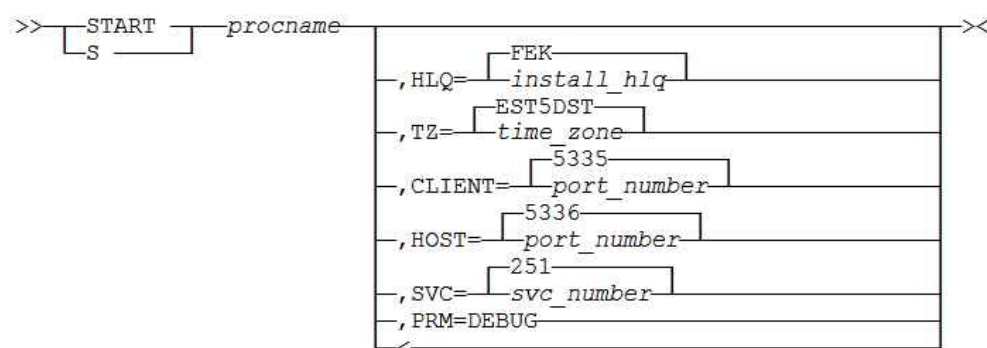


Abbildung 37. Bedienerbefehl 'START DBGMGR'

procname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der bei der Hostsystemkonfiguration verwendete Standardname ist DBGMGR.

HLQ=install_hlq

Für die Installation von Developer for System z verwendetes übergeordnetes Qualifikationsmerkmal. Die Standardeinstellung ist FEK.

TZ=time_zone

Zeitzonendifferenz. Die Standardeinstellung ist EST5DST.

CLIENT=Portnummer

Der für die externe Kommunikation (Client-Host-Kommunikation) verwendete Port. Der Standardwert lautet 5335.

HOST=Portnummer

Der für die interne, auf den Host begrenzte Kommunikation verwendete Port. Der Standardwert lautet 5336.

SVC=SVC-Nummer

Die für die Fehlerbehebung bei CICS-Transaktionen im Nur-Lese-Speicher verwendete SVC-Nummer. Der Standardwert lautet 251.

PRM=DEBUG

Aktivieren des ausführlichen Modus (Trace-Modus). Der Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

JES Job Monitor

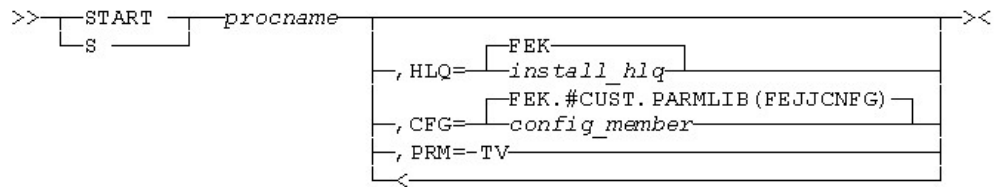


Abbildung 38. Bedienerbefehl 'START JMON'

procname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der bei der Hostsystemkonfiguration verwendete Standardname ist JMON.

HLQ=install_hlq

Für die Installation von Developer for System z verwendetes übergeordnetes Qualifikationsmerkmal. Die Standardeinstellung ist FEK.

CFG=config_member

Absoluter Name der Dateigruppe und des Members für die Konfigurationsdatei für JES Job Monitor. Die Standardeinstellung ist FEK.#CUST.PARMLIB(FEJJCNFG). Wenn diese Variable auf NULLFILE gesetzt ist, verwendet JES Job Monitor die Standardkonfigurationswerte.

PRM=-TV

Aktivieren des ausführlichen Modus (Trace-Modus). Der Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSE-Dämon

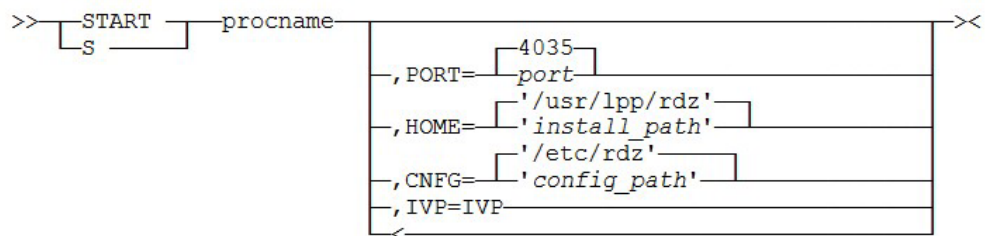


Abbildung 39. Bedienerbefehl 'START RSED'

procname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der bei der Hostsystemkonfiguration verwendete Standardname ist RSED.

PORT=port

Der Port des RSE-Dämons, zu dem Clients eine Verbindung herstellen. Wenn dieser Wert nicht angegeben ist, wird der in der Variablen _RSE_RSED_PORT in /etc/rdz/rsed.envvars definierte Port verwendet. Die Standardeinstellung ist 4035.

IVP=IVP

Server nicht starten, sondern das Installationsprüfprogramm für den RSE-Dämon ausführen

CNFG='config_path'

Absolute Position der unter z/OS UNIX gespeicherten Konfigurationsdateien. Die Standardeinstellung ist '/etc/rdz'. Beim z/OS UNIX-Pfad wird die Groß-/Kleinschreibung unterschieden, sodass der Pfad in einfache Anführungszeichen (') eingeschlossen werden muss, damit Kleinbuchstaben erhalten bleiben.

HOME='install_path'

Pfadpräfix und der obligatorische Pfad /usr/lpp/rdz für die Installation von Developer for System z. Die Standardeinstellung ist '/usr/lpp/rdz'. Beim z/OS UNIX-Pfad wird die Groß-/Kleinschreibung unterschieden, sodass der Pfad in einfache Anführungszeichen (') eingeschlossen werden muss, damit Kleinbuchstaben erhalten bleiben.

Modify (F)

Mit dem Befehl **MODIFY** können Kenndaten einer aktiven Task dynamisch abgefragt und geändert werden. Die abgekürzte Fassung des Befehls ist der Buchstabe F.

Integrated Debugger

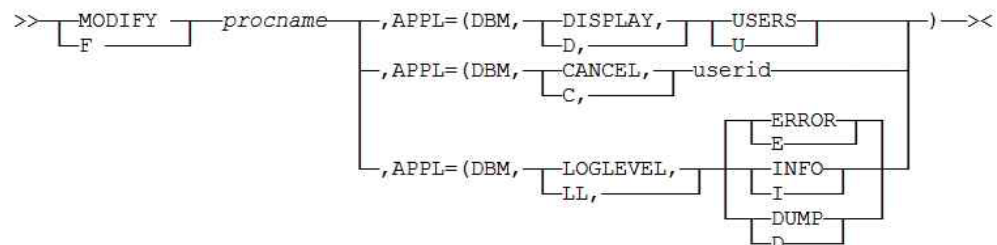


Abbildung 40. Bedienerbefehl 'MODIFY DBGMGR'

procname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der bei der Hostsystemkonfiguration verwendete Standardname ist DBGMGR.

DISPLAY,USERS

Zeigt die aktiven Benutzer mit AQECM104I-Konsolennachrichten an. Die Nachricht AQECM103I wird ausgegeben, wenn keine aktiven Benutzer vorhanden sind. In der Benutzerliste wird der Status des entsprechenden Benutzers im Server angezeigt. Ein Überblick über den Datenfluss von Integrated Debugger finden Sie im Abschnitt "Integrated Debugger" im Kapitel "Wissenswertes zu Developer for System z" in der *Hostkonfigurationsreferenz* (IBM Form SC12-4489).

```
AQECM104I User:IBMUSER RegisterSocket(2) waits for probe
connection
AQECM104I User:IBMUSR2 ProbeSocket(3) waits for register
connection
AQECM104I User:IBMUSR3 EngineSocket(4) connected to
ProbeSocket(8)
AQECM104I User:IBMUSR4 ProbeSocket(5) waits for engine
connection
AQECM103I There is no active user
```

CANCEL,userid

Bricht alle Debugsitzungen für die angegebene Benutzer-ID ab. Die Ergebnisse werden mit der AQECM110I- oder AQECM111I-Konsolennachricht angezeigt.

AQECM110I user(IBMUSER) canceled
AQECM111I user(IBMUSER) not connected

LOGLEVEL,{ERROR | INFO | DUMP}

Steuern des Detaillierungsgrads des Nachrichtenprotokolls von Debug Monitor (DD SYSPRINT). Der Standardwert ist E (Error=Fehler). Die Nachricht "LOGLEVEL command processed normally" wird mit der Nachrichten-ID AQECM101I an die Konsole geschrieben.

E oder ERROR	Nur Fehlernachrichten (Standardwert)
I oder INFO	Fehler- und Informationsnachrichten
D oder DUMP	Fehler- und Informationsnachrichten sowie Debug- und Speicherauszugsnachrichten.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

JES Job Monitor

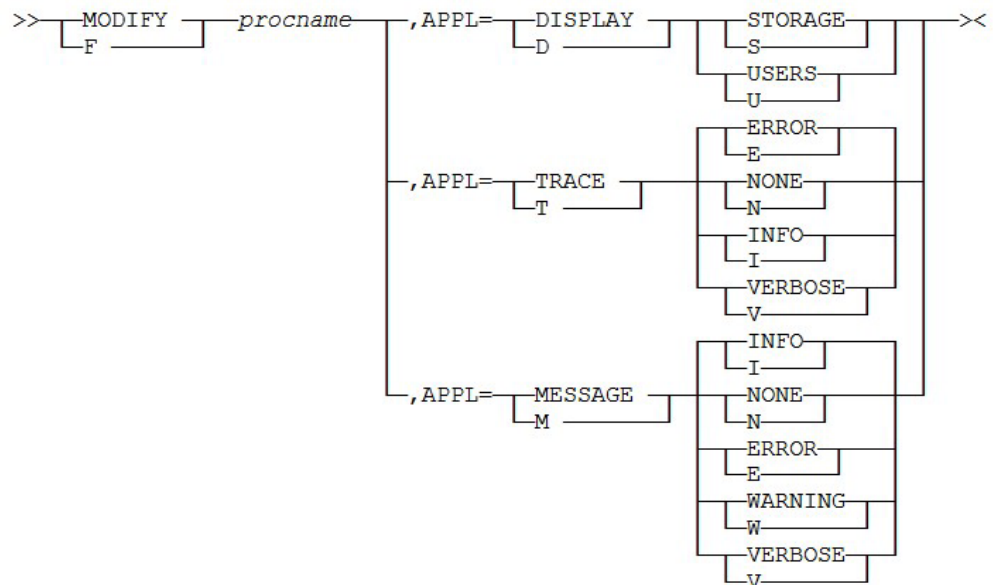


Abbildung 41. Bedienerbefehl 'MODIFY JMON'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wurde. Der bei der Hostsystemkonfiguration verwendete Standardname ist JMON.

DISPLAY STORAGE

Schreiben eines Speicherbelegungsberichts an die DD "SYSOUT". Die Nachricht "JMON storage information written to SYSOUT" wird mit der Nachrichten-ID BPXM023I an die Konsole geschrieben. Der Speicherbelegungsbericht zeigt verschiedene speicherbezogene Felder mit Größenangaben in Byte, Kilobyte und Megabyte an.


```
>>>STORAGE TRACE (console request)<<<
LDAREGRQ  0000000000 00000000K 00000M requested region size
  below 16M line
LDASIZE    00006266880 00006120K 00005M maximum region size
LDALIMIT   00006266880 00006120K 00005M limit
LDAVVRG    00006266880 00006120K 00005M getmain limit
LDALOAL    00000061440 00000060K 00000M in use
LDAHIAL    00000266240 00000260K 00000M LSQA/SWA/private subpools
  _GAP      00000000000 00000000K 00000M gaps in allocation
  _AVAIL     00005939200 00005800K 00005M available (including gaps)
  _MAX       00006000640 00005860K 00005M current limit
  above 16M line
LDAESIZE   01905262592 01860608K 01817M maximum region size
LDAELIM    01905262592 01860608K 01817M limit
LDAEVVRG    01905262592 01860608K 01817M getmain limit
LDAELOAL    00000937984 00000916K 00000M in use
LDAEHIAL    00012754944 00012456K 00012M ELSQA/ESWA/private subpools
  _EGAP      00000000000 00000000K 00000M gaps in allocation
  _EAVAIL    01891569664 01847236K 01803M available (including gaps)
  _EMAX      01892507648 01848152K 01804M current limit
```

DISPLAY USERS

Schreiben einer Liste der aktiven Benutzer an die DD "SYSOUT". Die Nachricht "JMON user list written to SYSOUT" wird mit der Nachrichten-ID BPXM023I an die Konsole geschrieben. Die Benutzerliste zeigt verschiedene benutzerbezogene Daten, einschließlich CPU-Auslastung, an.

```
S0  userid  USER      4:04(elapsed)    4:04(idle)
Users: 1
```

TRACE {NONE | ERROR | INFO | VERBOSE}

Steuern des Detaillierungsgrads des Traceprotokolls von JES Job Monitor (DD SYSOUT). Der Standardwert ist E (Error=Fehler). Die Nachricht "JMON TRACE LEVEL:{NONE | ERROR | INFO | VERBOSE}" wird mit der Nachrichten-ID BPXM023I an die Konsole geschrieben.

N oder NONE	Nur Startnachrichten
E oder ERROR	Nur Start- und Fehlernachrichten (Standardwert)
I oder INFO	Start-, Fehler- und Informationsnachrichten
V oder VERBOSE	Start-, Fehler- und Informationsnachrichten sowie ausführliche Nachrichten

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

MESSAGE {NONE | ERROR | WARNING | INFO | VERBOSE}

Steuern des Detaillierungsgrads des Nachrichtenprotokolls von JES Job Monitor (DD SYSPRINT). Der Standardwert ist I (Informational). Die Nachricht "JMON MESSAGE LEVEL:{NONE | ERROR | WARNING | INFO | VERBOSE}" wird mit der Nachrichten-ID BPXM023I an die Konsole geschrieben.

N oder NONE	Keine Nachrichten
E oder ERROR	Nur Fehlernachrichten

W oder WARNING	Fehlernachrichten und Warnungen
I oder INFO	Fehlernachrichten, Warnungen und Informationsnachrichten (Standardwert)
V oder VERBOSE	Fehlernachrichten, Warnungen, Informationsnachrichten sowie ausführliche Nachrichten

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSE-Dämon

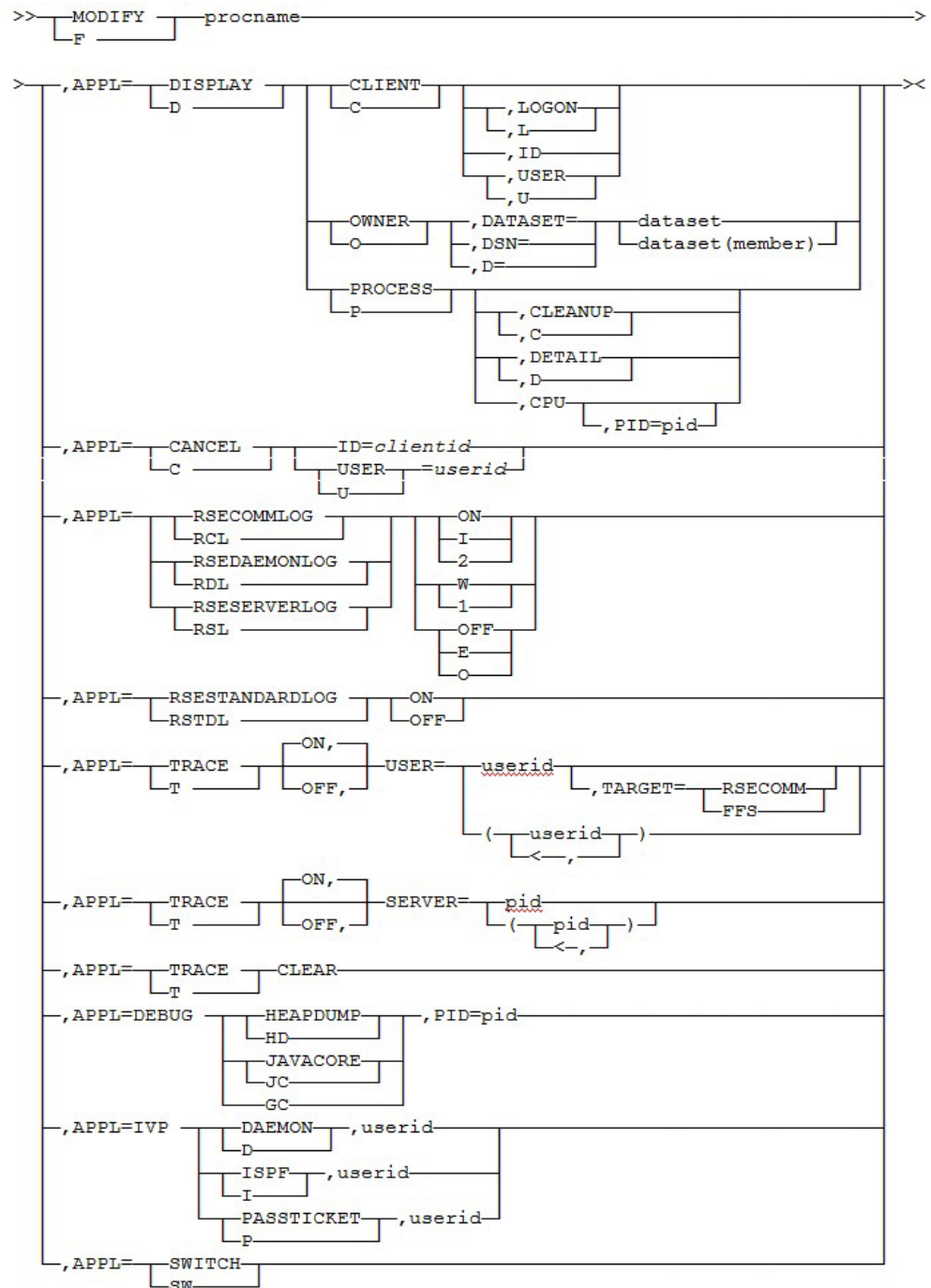


Abbildung 42. Bedienerbefehl 'MODIFY RSED'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wurde. Der bei der Hostsystemkonfiguration verwendete Standardname ist RSED.

DISPLAY CLIENT [{LOGON | ,ID | ,USER}]

Zeigt die aktiven Clients in einer einzelnen BPXM023I-Nachricht an. Das

Ergebnislayout hängt von der verwendeten Befehlsoption ab. Sie können die Sortierreihenfolge mit den optionalen Befehlsargumenten ändern.

- Keine Befehlsoption: Clients werden vom Thread-Pool-Prozess gruppiert, der diese bedient.

```
ProcessId(<Prozess-ID>) ASId(<AS-ID>) JobName(<Jobname>)
Clients(<lokal>/<Gesamt>) Order(<Startreihenfolge>)
<Client-ID><Benutzer-ID><verbunden seit>
```

- LOGON-Befehlsoption: Clients werden nach dem Anmeldezeitpunkt sortiert.

```
LOGON TIME----- ID----- USERID--
<verbunden seit>      <Client-ID> <Benutzer-ID>
```

- ID-Befehlsoption: Clients werden nach der Client-ID sortiert.

```
ID----- USERID-- LOGON TIME-----
<Client-ID> <Benutzer-ID> <verbunden seit>
```

- USER-Befehlsoption: Clients werden nach der Benutzer-ID sortiert.

```
USERID-- ID----- LOGON TIME-----
<Benutzer-ID> <Client-ID> <verbunden seit>
```

DISPLAY OWNER,DATASET={dataset | dataset(member)}

Zeigt den Dateiwarteschlangeneigner in einer einzelnen BPXM023I-Nachricht an.

```
FEK217I <Datei[(Member)]> is locked by <Benutzer-ID>
FEK218I <Datei[(Member)]> is not locked
FEK219E Failed to determine lock owner for <Datei[(Member)]>
```

- Der Server berichtet auch Sperren von anderen Produkten, wie z. B. ISPF.
- Der Bedienerbefehl **D GRS,RES=(*,Datei)** kann nicht erkennen, welcher Developer for System z-Benutzer der tatsächliche Inhaber der Warteschlange ist. Er kann lediglich den Thread-Pool benennen, in dem der Benutzer aktiv ist.

DISPLAY PROCESS[{,CLEANUP | ,CPU [,PID=pid] | ,DETAIL}]

Zeigt die RSE-Thread-Pool-Prozesse in einer oder mehreren BPXM023I-Nachrichten an. Für den Lastausgleich der verbundenen Benutzer kann es mehrere aktive Prozesse geben.

```
ProcessId(<Prozess-ID>) Memory Usage(<Belegung des Java-Heapspeichers>%)
Clients(<Anzahl der Clients>) Order(<Startreihenfolge>) <Fehlerstatus>
```

Anmerkung:

- <Prozess-ID> kann in prozessspezifischen z/OS UNIX-Bedienerbefehlen verwendet werden.
- Jeder Prozess hat einen eigenen Java-Heap-Speicher, dessen Größe in rsed.envvars festgelegt werden kann. Die berichtete Java-Heapspeicherbelegung schließt die von Developer for System z freigegebene, jedoch noch nicht durch den Java-Garbage-Collection-Prozess freigegebene Belegung ein.
- <Startreihenfolge> ist eine fortlaufende Zahl, die die Reihenfolge angibt, in der die Thread-Pools gestartet wurden. Die Zahl entspricht der Zahl im Dateinamen der Dateien stderr.*.log und stdout.*.log.

In normalen Situationen ist <Fehlerstatus> leer. In Tabelle 26 auf Seite 201 sind die möglichen, nicht leeren Werte für <Fehlerstatus> dokumentiert.

Tabelle 26. Fehlerstatus des Thread-Pools

Status	Beschreibung
severe error	Der Thread-Pool-Prozess hat einen nicht behebbaren Fehler festgestellt und die Operationen angehalten. In den anderen Statusfeldern werden die letzten bekannten Werte angezeigt. Verwenden Sie zum Entfernen dieses Eintrags aus der Tabelle die Option CLEANUP des Änderungsbefehls DISPLAY PROCESS .
killed process	Der Thread-Pool-Prozess wurde durch Java, z/OS UNIX oder einen Bedienerbefehl abgebrochen. In den anderen Statusfeldern werden die letzten bekannten Werte angezeigt. Verwenden Sie zum Entfernen dieses Eintrags aus der Tabelle die Option CLEANUP des Änderungsbefehls DISPLAY PROCESS .
timeout	Der Thread-Pool-Prozess hat dem RSE-Dämon während einer Clientverbindungsanforderung nicht zeitnah geantwortet. In den anderen Statusfeldern werden die aktuellen Werte angezeigt. Der Thread-Pool wird in zukünftigen Clientverbindungsanforderungen ausgeschlossen. Der Status *timeout* wird zurückgesetzt, wenn sich ein Client abmeldet, der von diesem Thread-Pool bereitgestellt wurde.

Es werden weitere Informationen bereitgestellt, wenn die Option "DETAIL" des Änderungsbefehls **DISPLAY PROCESS** verwendet wird:

```
ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
PROCESS LIMITS:  CURRENT  HIGHWATER  LIMIT
  JAVA HEAP USAGE(%)    10         56         100
    CLIENTS                0         25         30
  MAXFILEPROC           83        103       64000
  MAXPROCUSER           97         99         200
  MAXTHREADS             9         14       1500
  MAXTHREADTASKS         9         14       1500
```

Das Feld 'ASId' ist die Adressraum-ID in Hexadezimalschreibweise. Die Tabelle zum Verarbeitungslimit zeigt die aktuelle Ressourcennutzung, die obere Grenze für die Ressourcennutzung und die Ressourcengrenze an. Die definierte Grenze wird aufgrund anderer Begrenzungsfaktoren möglicherweise nie erreicht.

Die CPU-Option des Änderungsbefehls **DISPLAY PROCESS** zeigt die kumulierte CPU-Auslastung in Millisekunden der einzelnen Threads in einem Thread-Pool an. Jeder Thread-Pool hat eine BPXM023I-Nachricht. Standardmäßig geben alle Thread-Pools die CPU-Belegung an. Sie können den Bereich jedoch auf einen einzelnen Thread-Pools beschränken, indem Sie im Bedienerbefehl PID=pid angeben, wobei pid die Prozess-ID des Ziel-Thread-Pools ist.

```
ProcessId(421 ) ASId(007D) JobName(RSED8) Order(1)
USERID  THREAD-ID  TCB#  ACC_TIME  TAG
STCRSE  0EDE54000000000  005E6B60  822  1/ThreadPoolProcess
STCRSE  0EDE870000000001  005E69C8  001
STCRSE  0EDE980000000002  005E6518  1814
```

STCRSE	0EDEBA0000000003	005E66B0	2305
STCRSE	0EDECB0000000004	005E62F8	001
STCRSE	0EDED00000000005	005E60D8	001
STCRSE	0EDF860000000006	005C2BF8	628 6/ThreadPoolMonitor\$Memory
UsageMonitor			
STCRSE	0EDF970000000007	005C2D90	003 7/ThreadPoolMonitor
STCRSE	0EDFDB0000000008	005C29D8	001
STCRSE	0EE22E000000000E	005C1BE0	070
IBMUSER	0EE0EB0000000011	005C22B8	276 20/ServerReceiver
IBMUSER	0EE2500000000012	005C19C0	137 16/ServerUpdateHandler
IBMUSER	0EE2610000000013	005C17A0	509 15/ServerCommandHandler
IBMUSER	0EE1840000000014	005C1E00	065 21/ZosSystemMiner
STCRSE	0EE1510000000016	005C2098	078
STCRSE	0EE1950000000017	005C1580	001
IBMUSER	0EE23F0000000018	005C1360	021 26/UniversalFileSystemMine
r			
IBMUSER	0EE2A5000000001C	005C0CF0	003 27/EnvironmentMiner
IBMUSER	0EE283000000001D	005C1140	002 31/CommandMiner
IBMUSER	0EE272000000001E	005C0E88	081 32/MVSFileSystemMiner
IBMUSER	0EE294000000001F	005C0AD0	002 33/MVSByteStreamHandler\$Op
enCloseThread			
STCRSE	0EE2E90000000023	005C0470	001
IBMUSER	0EE2C70000000024	005C08B0	050 38/JESMiner
IBMUSER	0EE2B60000000026	005C0690	004 40/FAMiner
IBMUSER	0EE30B0000000027	005C0250	002 41/LuceneMiner
IBMUSER	0EE31C0000000028	005C0030	002 42/CDTParserMiner
IBMUSER	0EE32D0000000029	005BDE00	002 43/MVSLuceneMiner
IBMUSER	0EE33E000000002A	005BDBE0	002 44/CDTMVSParserMiner

Wenn die Ausgabegröße die maximale Anzahl der Zeilen für eine Konso-
lennachricht überschreitet, wird die Ausgabe in mehrere BPXM023I-Nach-
richten aufgeteilt. Diese zusätzlichen Nachrichten haben denselben Header
wie die erste Nachricht, jedoch ist der ersten Zeile das Schlüsselwort
CONTINUATION hinzugefügt.

```
ProcessId(421      ) ASId(007D) JobName(RSED8) Order(1) CONTINUATION
USERID  THREAD-ID      TCB#      ACC_TIME TAG
```

Die Ausgabe ist auf die ersten 4000 Threads für jeden Threadpool begrenzt.

CANCEL ID=Client-ID

Bricht die Clientverbindung auf der Basis der Client-ID ab, die im Modifi-
zierungsbefehl **DISPLAY CLIENT** angegeben ist.

Beim Abbrechen einer Clientverbindung wird für die Hostsystemthreads
eine normale Beendigung ausgeführt, um von den Threads verwendete
Ressourcen zu bereinigen. Diese Aktion impliziert, dass einige Minuten
vergehen können, bevor die Threads beendet werden. Dies kann zum Bei-
spiel daran liegen, dass sie auf eine Zeitlimitüberschreitung des Keepalive-
Mechanismus warten.

CANCEL USER=Benutzer-ID

Bricht die Clientverbindung auf der Basis der Benutzer-ID des Clients ab,
die im Modifizierungsbefehl **DISPLAY CLIENT** angegeben ist.

Beim Abbrechen einer Clientverbindung wird für die Hostsystemthreads
eine normale Beendigung ausgeführt, um von den Threads verwendete
Ressourcen zu bereinigen. Diese Aktion impliziert, dass einige Minuten
vergehen können, bevor die Threads beendet werden. Dies kann zum Bei-
spiel daran liegen, dass sie auf eine Zeitlimitüberschreitung des Keepalive-
Mechanismus warten.

RSECOMMLOG {ON | OFF | I | W | E | 2 | 1 | 0}

Steuert die Tracedetailstufe für den RSE-Server (rsecomm.log) und die

MVS-Dateiservices (lock.log und ffs*.log). Die Standardeinstellung wird beim Start in rsecomm.properties definiert. Drei Detaillierungsgrade sind verfügbar:

E oder 0 oder OFF	Nur Fehlermeldungen
W oder 1	Fehlermeldungen und Warnungen. Dies ist die Standardeinstellung in rsecomm.properties.
I oder 2 oder ON	Fehlermeldungen, Warnungen und Informationsmeldungen.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSEDAEMONLOG {ON | OFF | I | W | E | 2 | 1 | 0}

Steuert die Tracedetailstufe für den RSE-Dämon (rsedaemon.log). Die Standardeinstellung wird beim Start in rsecomm.properties definiert. Drei Detaillierungsgrade sind verfügbar:

E oder 0 oder OFF	Nur Fehlermeldungen
W oder 1	Fehlermeldungen und Warnungen. Dies ist die Standardeinstellung in rsecomm.properties.
I oder 2 oder ON	Fehlermeldungen, Warnungen und Informationsmeldungen.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSESERVERLOG {ON | OFF | I | W | E | 2 | 1 | 0}

Steuert die Tracedetailstufe für die RSE-Thread-Pools (rseserver.log). Die Standardeinstellung wird beim Start in rsecomm.properties definiert. Drei Detaillierungsgrade sind verfügbar:

E oder 0 oder OFF	Nur Fehlermeldungen
W oder 1	Fehlermeldungen und Warnungen. Dies ist die Standardeinstellung in rsecomm.properties.
I oder 2 oder ON	Fehlermeldungen, Warnungen und Informationsmeldungen.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSESTANDARDLOG {ON | OFF}

Inaktiviert (OFF) oder aktiviert (ON) die Aktualisierung der Protokolldateien mit den Datenströmen 'stdout' und 'stderr' der Thread-Pools stdout*.log und stderr*.log. Die Standardeinstellung wird beim Start durch die Anweisung enable.standard.log in rsd.envvars definiert.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

TRACE [{ON, | OFF,}]USER=userid[,TARGET={FFS | RSECOMM}]

Aktiviert (ON) oder inaktiviert (OFF) die Traceerstellung für die angegebenen Benutzer-IDs. Die Standardeinstellung ist ON. Diese Einstellung setzt die durch den Bedienerbefehl **MODIFY RSECOMMLOG** gesteuerte Standardeinstellung außer Kraft. Zwei Detaillierungsgrade sind verfügbar:

OFF	Nur Fehlermeldungen
ON (Standard)	Fehlermeldungen, Warnungen und Informationsmeldungen.

Der Befehl ändert die Tracedetailstufe für den RSE-Server (rsecomm.log) und die MVS-Dateiservices (lock.log und ffs*.log). Dies kann durch das Schlüsselwort TARGET eingeschränkt werden, welches zwei Werte akzeptiert:

FFS	Legt die angegebene Protokollstufe nur für MVS-Dateiservices (lock.log und ffs*.log) fest
RSECOMM	Legt die angegebene Protokollstufe nur für den RSE-Server (rsecomm.log) fest

Der Befehl kann für Benutzer ausgegeben werden, die derzeit nicht angemeldet sind. Die Einstellung bleibt aktiv, wenn sich ein Benutzer abmeldet, und wird erneut verwendet, wenn der Benutzer sich anmeldet.

Verwenden Sie die Anweisung USER in der Datei rsecomm.properties, um beim Serverstart die Ausgabe des Befehls **MODIFY TRACE USER** zu simulieren. Vorhandene Einstellungen von vorherigen Bedienerbefehlen vom Typ **MODIFY TRACE USER** oder **MODIFY TRACE SERVER** oder von der Anweisung USER in rsecomm.properties werden durch die Einstellung dieses Befehls ersetzt.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

TRACE [{ON, | OFF,}]USER=(userid,userid,...)

Aktiviert (ON) oder inaktiviert (OFF) die Traceerstellung für die angegebenen Benutzer-IDs. Die Standardeinstellung ist ON. Diese Einstellung setzt die durch den Bedienerbefehl **MODIFY RSECOMMLOG** gesteuerte Standardeinstellung außer Kraft. Zwei Detaillierungsgrade sind verfügbar:

OFF	Nur Fehlermeldungen
ON (Standard)	Fehlermeldungen, Warnungen und Informationsmeldungen.

Der Befehl ändert die Tracedetailstufe für den RSE-Server (rsecomm.log) und die MVS-Dateiservices (lock.log und ffs*.log). Der Befehl kann für Benutzer ausgegeben werden, die derzeit nicht angemeldet sind. Die Einstellung bleibt aktiv, wenn sich ein Benutzer abmeldet, und wird erneut verwendet, wenn der Benutzer sich anmeldet. Verwenden Sie die Anweisung USER in der Datei rsecomm.properties, um beim Serverstart die Ausgabe des Befehls **MODIFY TRACE USER** zu simulieren. Vorhandene Einstellungen von vorherigen Bedienerbefehlen vom Typ **MODIFY TRACE USER** oder **MODIFY TRACE SERVER** oder von der Anweisung USER in rsecomm.properties werden durch die Einstellung dieses Befehls ersetzt.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

TRACE [{ON, | OFF,}] SERVER={pid | (pid,pid,...)}

Aktiviert (ON) oder inaktiviert (OFF) die Traceerstellung für alle Benutzer im angegebenen Thread-Pool. Dabei steht pid für die Prozess-ID eines RSE-Thread-Pools. Die Standardeinstellung ist ON. Diese Einstellung setzt die durch den Bedienerbefehl **MODIFY RSECOMMLOG** gesteuerte Standardeinstellung außer Kraft. Zwei Detaillierungsgrade sind verfügbar:

OFF	Nur Fehlermeldungen
ON (Standard)	Fehlermeldungen, Warnungen und Informationsmeldungen.

Der Befehl ändert die Tracedetailstufe für den RSE-Server (rsecomm.log) und die MVS-Dateiservices (lock.log und ffs*.log). Vorhandene Einstellungen von vorherigen Bedienerbefehlen vom Typ **MODIFY TRACE USER** oder **MODIFY TRACE SERVER** oder von der Anweisung **USER** in rsecomm.properties werden durch die Einstellung dieses Befehls ersetzt.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

TRACE CLEAR

Entfernt alle Traceüberschreibungen, die durch die Bedienerbefehle **MODIFY TRACE USER** und **MODIFY TRACE SERVER** und die Anweisung **USER** in rsecomm.properties festgelegt wurden.

DEBUG HEAPDUMP,PID=pid

Fordert einen Auszug des Java-Heapspeichers für einen angegebenen Thread-Pool an, wobei pid die Prozess-ID eines RSE-Thread-Pools ist. Der Speicherauszug wird in das durch **_CEE_DUMPTARG** in rsed.envvars angegebene Verzeichnis geschrieben. Standardwert: /tmp. Die Ergebnisse werden in einer einzigen BPXM023I-Konsolennachricht angezeigt.

```
JVMDUMP034I User requested Heap dump using '/tmp/heapdump.20120223.211'
430.16777590.0001.phd' through JVMRI
```

DEBUG JAVACORE,PID=pid

Fordert einen Auszug des Java-Kernspeichers für einen angegebenen Thread-Pool an, wobei pid die Prozess-ID eines RSE-Thread-Pools ist. Der Speicherauszug wird in das durch **_CEE_DUMPTARG** in rsed.envvars angegebene Verzeichnis geschrieben. Standardwert: /tmp. Die Ergebnisse werden in einer einzigen BPXM023I-Konsolennachricht angezeigt.

```
JVMDUMP034I User requested Java dump using '/tmp/javacore.20120223.214
244.16777590.0002.phd' through JVMRI
```

DEBUG GC,PID=pid

Fordert eine Java-Garbage-Collection für einen angegebenen Thread-Pool an, wobei pid die Prozess-ID eines RSE-Thread-Pools ist.

IVP DAEMON,userid

Meldet die Benutzer-ID userid beim RSE-Dämon an, um einen Verbindungstest durchzuführen. Die Ergebnisse werden in einer oder mehreren FEK900I-Konsolennachrichten angezeigt. Der Rückkehrcode wird mit der Konsolennachricht FEK901I angezeigt.

```
+FEK900I DAEMON IVP: SSL is disabled
+FEK900I DAEMON IVP: connected
+FEK900I DAEMON IVP: 1977
+FEK900I DAEMON IVP: 6902918
+FEK900I DAEMON IVP: Success
+FEK901I DAEMON IVP Exit code = 0
```

Anmerkung:

- Diese Funktion ist mit dem Installationsprüfprogramm (Installation Verification Program, IVP) fekfivpd vergleichbar.
- Der RSE-Dämon generiert ein PassTicket, das als Kennwort für IVP verwendet wird. Daher erfolgt keine WTOR-Operation (Write To Operator with Reply) mit Anforderung eines Kennworts.

IVP ISPF,userid

Ruft das Client-Gateway der Interactive System Productivity Facility (ISPF) mit der Benutzer-ID userid auf. Die Ergebnisse werden in einer oder mehreren FEK900I-Konsolennachrichten angezeigt. Der Rückkehrcode wird mit der Konsolennachricht FEK901I angezeigt.

```
+FEK900I ISPF IVP: executed on CDFMVS08 -- Tue Sep 13 22:29:28 EDT 2011
+FEK900I ISPF IVP: executed by uid=1(IBMUSER) gid=0(SYS1)
+FEK900I ISPF IVP: using /etc/rdz/rsed.envvars
+FEK900I ISPF IVP: current address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: maximum address space size limit is 2147483647
(2048.0 MB)
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: /etc/rdz/ISPF.conf content:
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: isplib=ISP.SISPLOAD
+FEK900I ISPF IVP: ispmllib=ISP.SISPMENU
+FEK900I ISPF IVP: isptlib=ISP.SISPTENU
+FEK900I ISPF IVP: ispplib=ISP.SISPPENU
+FEK900I ISPF IVP: ispslib=ISP.SISPSLIB
+FEK900I ISPF IVP: sysproc=ISP.SISPCLIB,FEK.SFEKPROC
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host install verification for RSE
+FEK900I ISPF IVP: Review IVP log messages from HOST below :
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Service level 22Feb2011
+FEK900I ISPF IVP: RSE connection and base TSO/ISPF session initializati
on check only
+FEK900I ISPF IVP: *** CHECK : ENVIRONMENT VARIABLES - key variables
displayed below :
+FEK900I ISPF IVP: Server PATH = ./usr/lpp/java/J6.0/bin:/usr/l
pp/rdz/bin:/usr/lpp/ispf/bin:/bin:/usr/sbin
+FEK900I ISPF IVP: STEPLIB = NONE
+FEK900I ISPF IVP: Temporary directory = /tmp
+FEK900I ISPF IVP: CGI_ISPHOME = /usr/lpp/ispf
+FEK900I ISPF IVP: CGI_ISPCONF = /etc/rdz
+FEK900I ISPF IVP: CGI_ISPWORK = /var/rdz
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : USS MODULES
+FEK900I ISPF IVP: Checking ISPF Directory : /usr/lpp/ispf
+FEK900I ISPF IVP: Checking modules in /usr/lpp/ispf/bin directory
+FEK900I ISPF IVP: Checking for ISPF configuration file ISPF.conf
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK : TSO/ISPF INITIALIZATION
+FEK900I ISPF IVP: ( TSO/ISPF session will be initialized )
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: *** CHECK: Shutting down TSO/ISPF IVP session
+FEK900I ISPF IVP: RC=0
+FEK900I ISPF IVP: MSG: SUCCESSFUL
+FEK900I ISPF IVP: -----
-----
+FEK900I ISPF IVP: Host installation verification completed successfully
+FEK900I ISPF IVP: -----
-----
+FEK901I ISPF IVP Exit code = 0
```

- Diese Funktion ist mit dem Installationsprüfprogramm (Installation Verification Program, IVP) `fekfivpi` vergleichbar.
- Der RSE-Dämon generiert ein PassTicket, das als Kennwort für IVP verwendet wird. Daher erfolgt keine WTOR-Operation (Write To Operator with Reply) mit Anforderung eines Kennworts.

IVP PASSTICKET,userid

Testet die Wiederverwendbarkeit eines PassTickets, das für die Benutzer-ID Benutzer-ID erstellt wurde. Die Ergebnisse werden in einer oder mehreren FEK900I-Konsolennachrichten angezeigt. Der Rückkehrcode wird mit der Konsolennachricht FEK901I angezeigt.

```
+FEK900I PASSTICKET IVP: the default applid=FEKAPPL
+FEK900I PASSTICKET IVP: Success, PassTicket IVP finished normally
+FEK901I PASSTICKET IVP Exit code = 0
```

Anmerkung:

- Bei der Verwendung von RACF als Sicherheitsprodukt muss sich für wiederverwendbare PassTickets das Schlüsselwort "NO REPLAY PROTECTION" in den Sicherheitsdefinitionen befinden.
- Für diesen Test gibt es kein funktional entsprechendes IVP (Installation Verification Program, Installationsprüfprogramm). Durch den Start des RSE-Dämons mit dem Argument IVP=IVP wird ein PassTicket-IVP aufgerufen, das die PassTicket-Generierung prüft, die PassTicket-Wiederverwendbarkeit jedoch nicht prüfen kann.
- Der RSE-Dämon generiert ein PassTicket, das als Kennwort für IVP verwendet wird. Daher erfolgt keine WTOR-Operation (Write To Operator with Reply) mit Anforderung eines Kennworts.

SWITCH

Wechselt zu einer neuen Prüfprotokolldatei.

Stop (P)

Mit dem Befehl **STOP** können Sie eine aktive Task stoppen. Die abgekürzte Fassung des Befehls ist der Buchstabe P.

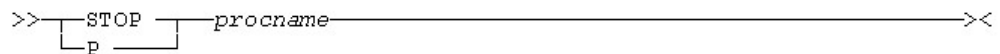


Abbildung 43. Bedienerbefehl 'STOP'

procname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wurde. Die bei der Hostsystemkonfiguration verwendeten Standardnamen sind DBGMGR, JMON und RSED für Integrated Debugger, JES Job Monitor bzw. für den RSE-Dämon.

Hinweise zum Lesen eines Syntaxdiagramms

Das Syntaxdiagramm zeigt Ihnen, wie ein Befehl angegeben werden muss, damit das Betriebssystem Ihre Eingabe ordnungsgemäß interpretieren kann. Das Syntaxdiagramm wird von links nach rechts und von oben nach unten gelesen. Folgen Sie dabei der horizontalen Linie, die den Hauptpfad darstellt.


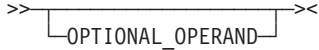
Symbole

In Syntaxdiagrammen werden die folgenden Symbole verwendet:

Symbol	Beschreibung
>>	Markiert den Anfang des Syntaxdiagramms
>	Zeigt an, dass das Syntaxdiagramm fortgesetzt wird
	Markiert Anfang und Ende eines Fragments oder Abschnitts des Syntaxdiagramms
<<	Markiert das Ende des Syntaxdiagramms

Operanden

In Syntaxdiagrammen werden die folgenden Arten von Operanden verwendet:

- Erforderliche Operanden werden auf der Linie des Hauptpfads angezeigt:

- Optionale Operanden werden unterhalb der Linie des Hauptpfads angezeigt:


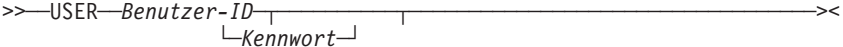
- Standardoperanden werden oberhalb der Linie des Hauptpfads angezeigt:


Operanden werden als Schlüsselwörter oder Variablen klassifiziert:

- Schlüsselwörter sind Konstanten, die angegeben werden müssen. Erscheint das Schlüsselwort im Syntaxdiagramm in gemischter Groß- und Kleinschreibung, gibt der Abschnitt in Großschreibung die Abkürzung für das Schlüsselwort an. Beispiel: KEYword. Bei Schlüsselwörtern wird die Groß-/Kleinschreibung nicht unterschieden.
- Variablen sind kursiv angegeben. Sie erscheinen in Kleinbuchstaben und repräsentieren Namen oder Werte, die Sie angeben müssen. Ein Dateigruppenname ist beispielsweise eine Variable. Bei Variablen muss unter Umständen die Groß-/Kleinschreibung beachtet werden.


Syntaxbeispiel

Im folgenden Beispiel ist der Befehl `USER` ein Schlüsselwort. Der erforderliche variable Parameter ist `Benutzer-ID`, und der optionale variable Parameter ist `Kennwort`. Ersetzen Sie die variablen Parameter durch Ihre eigenen Werte:



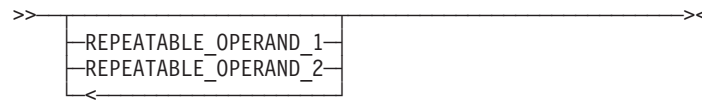
Nicht alphanumerische Zeichen und Leerzeichen

Wenn ein Diagramm ein Zeichen enthält, das kein alphanumerisches Zeichen ist, z. B. Klammern, Punkte, Kommata, Gleichheitszeichen und Leerzeichen, müssen Sie das Zeichen als Teil der Syntax eingeben. In diesem Beispiel muss die Eingabe `OPERAND=(001 0.001)` lauten:



Mehrere Operanden auswählen

Ein nach links weisender Pfeil in einer Gruppe von Operanden bedeutet, dass mehr als ein Operand ausgewählt oder ein einzelner Operand wiederholt werden kann:



Mehrere Zeilen

Wenn ein Diagramm mehr als eine Zeile umfasst, endet die erste Zeile mit einer einzelnen Pfeilspitze und die zweite Zeile beginnt mit einer einzelnen Pfeilspitze:

>>—| Die erste Zeile des Syntaxdiagramms, das mehr als eine Zeile umfasst |—>
>—| Die Fortsetzung der Unterbefehle und/oder Parameter |————><

Syntaxfragmente

Einige Diagramme können Syntaxfragmente enthalten, die zur Unterteilung zu langer oder zu komplexer Diagramme bzw. von Diagrammen mit zu vielen Wiederholungen dienen. Die Namen von Syntaxfragmenten sind in gemischter Groß-/Kleinschreibung angegeben und erscheinen im Diagramm sowie in der Überschrift des Diagramms. Das Fragment ist unterhalb des Hauptdiagramms dargestellt:

>>—| Syntaxfragment |————><

Syntaxfragment:

|—1ST_OPERAND—,—2ND_OPERAND—,—3RD_OPERAND—|

Kapitel 12. Hostkonfigurationsreferenz

In diesem Abschnitt werden die Informationen im Handbuch *Rational Developer for System z Hostkonfigurationsreferenz* (IBM Form SC12-4489) zusammengefasst. Weitere Details finden Sie in dieser Veröffentlichung.

Wissenswertes zu Developer for System z

Das Developer for System z-Hostsystem besteht aus mehreren Komponenten, die interagieren, um den Zugriff auf die Hostsystemservices und Hostsystemdaten für den Client bereitzustellen. Wenn Sie mit dem Design dieser Komponenten vertraut sind, können Sie die richtigen Konfigurationsentscheidungen treffen.

Sicherheitsaspekte

Developer for System z ermöglicht Benutzern einer Workstation den Zugriff auf Mainframe-Computer, wenn diese selbst kein Mainframe-Computer ist. Wichtige Aspekte bei der Produktkonfiguration sind deshalb das Prüfen von Verbindungsanforderungen, das Bereitstellen von sicherer Kommunikation zwischen dem Hostsystem und der Workstation sowie das Autorisieren und Protokollieren der Aktivitäten.

TCP/IP-Aspekte

Developer for System z stellt Benutzern einer Workstation den Zugriff auf Mainframe-Computer über TCP/IP bereit, wenn diese selbst kein Mainframe-Computer ist. Außerdem wird TCP/IP für die Kommunikation zwischen verschiedenen Komponenten und anderen Produkten verwendet.

Hinweise zu WLM

Im Gegensatz zu herkömmlichen z/OS-Anwendungen ist Developer for System z keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for System z besteht aus mehreren Komponenten, die interagieren, um dem Client den Zugriff auf die Hostsystemservices und -daten zu ermöglichen. Einige dieser Services sind in verschiedenen Adressräumen aktiv und werden somit verschiedenen WLM-Klassifikationen zugeordnet.

Optimierungsaspekte

RSE (Remote Systems Explorer) ist der Kern von Developer for System z. Zur Verwaltung der Verbindungen und Arbeitslasten von den Clients verfügt RSE über einen Dämonadressbereich, der Thread-Pool-Adressräume steuert. Der Dämon wird als Sammelpunkt für Verbindungs- und Verwaltungszwecke eingesetzt, während die Thread-Pools die Clientarbeitslast verarbeiten.

Diese Konfiguration macht RSE zu einem Hauptziel für die Optimierung der Einrichtung von Developer for System z. Wenn Sie allerdings Hunderte von Benutzern verwalten, die jeweils 17 oder mehr Threads, eine bestimmte Speichermenge und möglicherweise einen oder mehrere Adressräume verwenden, müssen Developer for System z und z/OS richtig konfiguriert sein.

Leistungsaspekte

z/OS ist ein sehr anpassungsfähiges Betriebssystem, bei dem (manchmal kleine) Systemänderungen eine enorme Auswirkung auf die Gesamtleistung haben können. Dieses Kapitel hebt einige der Änderungen hervor, die zu einer Verbesserung der Leistung von Developer for System z führen können.

Hinweise zu Push-to-Client

Push-to-Client (oder hostbasierte Clientsteuerung) unterstützt die zentrale Verwaltung folgender Elemente:

- Clientkonfigurationsdateien
- Clientproduktversion
- Projektdefinitionen

CICSTS-Aspekte

Dieses Kapitel enthält nützliche Informationen für CICS Transaction Server-Administratoren.

Hinweise zum Benutzerexit

In diesem Kapitel finden Sie Hilfen für die Erweiterung von Developer for System z durch das Schreiben von Exitroutinen.

Anpassung der TSO-Umgebung

Dieses Kapitel soll Sie beim Imitieren einer TSO-Anmeldeprozedur durch das Hinzufügen von DD-Anweisungen und Dateien zur TSO-Umgebung in Developer for System z unterstützen.

Ausführung mehrerer Instanzen

In bestimmten Situationen, z. B. beim Testen eines Upgrades, kann die Ausführung mehrerer aktiver Instanzen von Developer for System z auf demselben System erwünscht sein. Manche Ressourcen können jedoch nicht gemeinsam genutzt werden, z. B. TCP/IP-Ports, sodass die Standardeinstellungen nicht immer anwendbar sind. Anhand der Informationen in diesem Kapitel können Sie die Koexistenz der verschiedenen Instanzen von Developer for System z planen, um sie dann mithilfe der Informationen in diesem Konfigurationshandbuch anzupassen.

Konfigurationsprobleme lösen

Dieses Kapitel soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von Developer for System z auftreten könnten. Es enthält die folgenden Abschnitte:

- Protokoll- und Konfigurationsanalyse mit FEKLOGS
- Protokolldateien
- Speicherauszugsdateien
- Trace-Erstellung
- z/OS UNIX-Berechtigungsbits
- Reservierte TCP/IP-Ports
- Adressbereich, Größe

- APPC-Transaktion und TSO Commands Service
- Weitere Informationen

SSL- und X.509-Authentifizierung konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von SSL (Secure Sockets Layer) oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten. Dieser Anhang stellt auch eine Beispielkonfiguration zur Verfügung, um Benutzer zu unterstützen, die sich mit einem X.509-Zertifikat selbst authentifizieren.

TCP/IP konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von TCP/IP oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten können.

Literaturübersicht

Referenzierte Veröffentlichungen

In diesem Dokument werden die folgenden Veröffentlichungen referenziert:

Tabelle 27. Referenzierte Veröffentlichungen

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
Program Directory for IBM Rational Developer for System z	GI11-8298	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Program Directory for IBM Rational Developer for System z Host Utilities	GI13-2864	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Prerequisites	SC23-7659	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Host Configuration Quick Start	GI11-3191	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Hostkonfiguration	SC12-4062	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Hostkonfigurationsreferenz	SC12-4489	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Host Configuration Utility	SC12-4472	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Messages and Codes	SC14-7497	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z - Antworten auf gängige Fragen der Hostkonfiguration und -wartung	SC12-4724	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Common Access Repository Manager Developer's Guide	SC23-7660	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Prerequisites	SC23-7659	Developer for System z	http://www.ibm.com/software/rational/products/developer/systemz/library/index.html
Rational Developer for System z Host Configuration Quick Start	GI11-3191	Developer for System z	http://www.ibm.com/software/rational/products/developer/systemz/library/index.html
SCLM Developer Toolkit Administrator's Guide	SC23-9801	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Using APPC to provide TSO command services	SC14-7291	White paper	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Using ISPF Client Gateway to provide CARMA services	SC14-7292	White paper	http://www-01.ibm.com/support/docview.wss?uid=swg27038517

Tabelle 27. Referenzierte Veröffentlichungen (Forts.)

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
Communications Server IP Configuration Guide	SC31-8775	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Configuration Reference	SC31-8776	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Diagnosis Guide	GC31-8782	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP System Administrator's Commands	SC31-8781	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA Network Implementation Guide	SC31-8777	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA Operations	SC31-8779	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Cryptographic Services System SSL Programming	SC24-5901	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
DFSMS Macro Instructions for Data Sets	SC26-7408	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
DFSMS Using data sets	SC26-7410	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Language Environment Customization	SA22-7564	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Language Environment Debugging Guide	GA22-7560	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Diagnosis: Tools and Service Aids	GA22-7589	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL Reference	SA22-7597	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning APPC/MVS Management	SA22-7599	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning Workload Management	SA22-7602	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS System Commands	SA22-7627	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E Customization	SA22-7783	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E REXX Reference	SA22-7790	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

Tabelle 27. Referenzierte Veröffentlichungen (Forts.)

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
UNIX System Services Command Reference	SA22-7802	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Planning	GA22-7800	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services User's Guide	SA22-7801	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Using REXX and z/OS UNIX System Services	SA22-7806	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Java™ Diagnostic Guide	SC34-6650	Java 6.0	http://www.ibm.com/developerworks/java/jdk/diagnosis/
Java SDK and Runtime Environment User Guide	/	Java 6.0	http://www-03.ibm.com/servers/eserver/zseries/software/java/
Resource Definition Guide	SC34-6430	CICS TS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
Resource Definition Guide	SC34-6815	CICS TS 3.2	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
Resource Definition Guide	SC34-7000	CICSTS 4.1	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
Resource Definition Guide	SC34-7181	CICSTS 4.2	https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
RACF Security Guide	SC34-6454	CICS TS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
RACF Security Guide	SC34-6835	CICS TS 3.2	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
RACF Security Guide	SC34-7003	CICSTS 4.1	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
RACF Security Guide	SC34-7179	CICSTS 4.2	https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
Language Reference	SC27-1408	Enterprise COBOL für z/OS	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html

In diesem Dokument werden die folgenden Websites referenziert:

Tabelle 28. Referenzierte Websites

Beschreibung	Referenzwebsite
Developer for System z Information Center	http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp
Developer for System z-Bibliothek	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Developer for System z-Homepage	http://www-03.ibm.com/software/products/us/en/developerforsystemz/
Empfohlener Service für Developer for System z	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335
Verbesserungsvorschlag für Developer for System z	https://www.ibm.com/developerworks/support/rational/rfe/

Tabelle 28. Referenzierte Websites (Forts.)

Beschreibung	Referenzwebsite
z/OS-Internetbibliothek	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Information Center für CICSTS	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp
IBM Tivoli Directory Server	http://www-01.ibm.com/software/tivoli/products/directory-server/
Tool-Plug-ins für die Problembestimmung	http://www-01.ibm.com/software/awdtools/deployment/pdtplugins/
Download von Apache Ant	http://ant.apache.org/
Java-Keytool-Dokumentation	http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html
Homepage der CA-Unterstützung	https://support.ca.com/

Veröffentlichungen mit weiteren Informationen

Die folgenden Veröffentlichungen können Antworten auf Fragen enthalten, die vielleicht bei der Konfiguration der erforderlichen Hostsystemkomponenten auftreten.

Tabelle 29. Veröffentlichungen mit weiteren Informationen

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	http://www.redbooks.ibm.com/
System Programmer's Guide to: Workload Manager	SG24-6472	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	http://www.redbooks.ibm.com/
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	http://www.redbooks.ibm.com/
Tivoli Directory Server for z/OS	SG24-7849	Redbook	http://www.redbooks.ibm.com/

Bemerkungen zur Dokumentation für IBM Rational Developer for System z

© Copyright IBM Corporation 2009, 2013.

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. An Stelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene 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:

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
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 im Dokument aufgeführten 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.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

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.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

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; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

Copyrightlizenz

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmier-techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme in beliebiger Form kopieren, ändern und verteilen, ohne dass dafür Zahlungen an IBM anfallen, 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 Musterprogramme 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 Musterprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM haftet nicht für Schäden, die durch Verwendung oder im Zusammenhang mit den Beispielprogrammen entstehen.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. 2009, 2013.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farabbildungen.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der International Business Machines Corp. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie im Web 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.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Windows ist eine Marke der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein.

Antwort

IBM Rational Developer for System z
Version 9.0.1
Konfiguration

IBM Form SC12-4062-10

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 0180 3 313233) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: translation@de.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

IBM Deutschland GmbH
TSC Germany

71083 Herrenberg



SC12-4062-10

