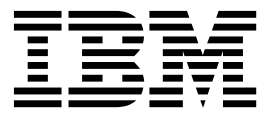


IBM Rational Developer for z Systems
Version 9.5.1

Konfiguration



IBM Rational Developer for z Systems
Version 9.5.1

Konfiguration



Hinweis

Vor Verwendung dieser Informationen sollten die allgemeinen Hinweise im Abschnitt „Bemerkungen“ auf Seite 155 gelesen werden.

Erste Ausgabe (Dezember 2015)

Diese Ausgabe bezieht sich auf IBM Rational Developer for z Systems Version 9.5.1 (Programmnummer 5724-T07 oder einen Teil von Programmnummer 5697-CDT) und - sofern in neuen Ausgaben nicht anders angegeben - auf alle nachfolgenden Releases und Modifikationen.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM Rational Developer for z Systems Version 9.5.1 Configuration Guide,
IBM Form SC27-8577-00,
herausgegeben von International Business Machines Corporation, USA

(C) Copyright International Business Machines Corporation 2000, 2015

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 2015

© Copyright IBM Corporation 2015, 2015.

Inhalt

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

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

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

Zielgruppe	xii
Zusammenfassung der Änderungen	xii
Beschreibung des Dokumentinhalts	xiv
Planung	xiv
Basisanpassung	xv
Common Access Repository Manager (CARMA).	xv
SCLM Developer Toolkit	xv
Integrated Debugger	xv
Hostbasierte Codeanalyse	xv
Weitere Anpassungstasks	xvi
Installationsprüfung	xvi
Sicherheitsdefinitionen	xvi
Leitfaden für die Migration.	xvi
Hostkonfigurationsreferenz	xvi

IBM Rational Developer for z Systems Hostkonfiguration	1
---	----------

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

Hinweise zur Migration	3
Hinweise zur Planung	3
Produktübersicht	3
Erforderliche Qualifikationen	4
Zeitbedarf	4
Hinweise zur Installationsvorbereitung	5
ID des Installationsbenutzers	5
Vorausgesetzte Produkte	6
Erforderliche Ressourcen	6
Hinweise zur Konfigurationsvorbereitung	9
Workload-Management	9
Ressourcennutzung und Systembegrenzungen	9
Erforderliche Konfiguration für vorausgesetzte Produkte	9
Hinweise zur Benutzer-ID	9
Hinweise zum Server	10
Hinweise zur Implementierungsvorbereitung	11
Clientprüfliste	12

Kapitel 2. Basisanpassung	15
--	-----------

Voraussetzungen und Prüfliste	15
Anpassungskonfiguration.	15
PARMLIB-Änderungen	16
z/OS UNIX-Grenzwerte in BPXPRMxx festlegen	17
Gestartete Tasks zu COMMNDxx hinzufügen	17
SVC-Definitionen in IEASVCxx.	18
Produktaktivierung in IFAPRDxx	18
LPA-Definitionen in LPALSTxx.	19
APF-Berechtigungen in PROGxx	19

LINKLIST-Definitionen in PROGxx	20
Vorausgesetzte LINKLIST- und LPA-Definitionen	22
LINKLIST-Definitionen für andere Produkte	23
PROCLIB-Änderungen	23
DBGMR, gestartete Task für Debug Manager	23
ELAXF* - ferne Buildprozeduren	24
Sicherheitsdefinitionen.	26
Die Umgebungsdatei 'rdz.env'	27
Bestimmte Komponenten.	31
Installationsprüfung	31

Kapitel 3. Common Access Repository Manager (CARMA).	33
---	-----------

Voraussetzungen und Prüfliste	33
Serverstartmethode und aktiven RAM auswählen	34
CARMA-Serverstart	34
CRASTART	34
Batchübergabe	34
Produktions-RAM	34
CA Endevor SCM-RAM	35
RAM für CA Endevor SCM-Pakete	35
Muster-RAM	35
PDS-RAM	35
Skeleton-RAM	35
SCLM-RAM	35
Kombinationen aus vorkonfiguriertem RAM und Serverstart.	35
CRASTART mit CA Endevor SCM-RAM.	35
CARMA-VSAM-Dateien erstellen	36
CRASRV.properties anpassen	36
crastart.endevor.conf anpassen	36
Zusätzliche Anpassung des CA Endevor SCM-RAM (optional)	37
CRASTART mit Muster-RAM	38
CARMA-VSAM-Dateien erstellen	38
CARMA	38
Muster-RAM	38
CRASRV.properties anpassen	38
crastart.conf anpassen	39
Zusätzliche benutzerdefinierte Anpassung des RAM (optional)	39
Batchübergabe mit CA Endevor SCM-RAM.	40
CARMA-VSAM-Dateien erstellen	40
CRASRV.properties anpassen	40
CRASUBCA anpassen	40
Zusätzliche Anpassung des CA Endevor SCM-RAM (optional)	42
Batchübergabe mit Muster-RAM	42
VSAM-Dateien erstellen	42
CARMA	42
Muster-RAM	42
CRASRV.properties anpassen	42
CRASUBMT anpassen	43
Zusätzliche benutzerdefinierte Anpassung des RAM (optional)	44

CARMA-Konfigurationsdetails	44	SCLM-Updates für SCLMDT	75
CRASRV.properties: RSE-Schnittstelle zu CARMA	45	Alte Dateien aus WORKAREA und /tmp entfernen	76
crastart*.conf - CRASTART-Serverstart	48		
CRASTART-Protokolldateien erfassen. . . .	51	Kapitel 5. Integrated Debugger	77
CRASUB* - Serverstart mit Batchübergabe . .	52	Voraussetzungen und Prüfliste	77
CARMA-VSAM-Dateien	54	Integrated Debugger und COBOL V4.	78
CRADEF - Konfigurationsdatei	54	Integrated Debugger und andere Language Environ-	
CRAMSG - Nachrichtendatei	54	ment-basierte Debugger	79
CRASTRS - Datei mit angepassten Zeichenfol-		Integrated Debugger und verschlüsselte Kommuni-	
gen	54	kation	79
CARMA-RAM (Repository Access Manager) ..	54	Integrated Debugger und SYSPLEX	79
CA Endeavor SCM-RAM	55	Konfigurationsparameter für Integrated Debugger	80
RAM für CA Endeavor SCM-Pakete	55	Parmlib-Aktualisierungen für Integrated Debugger	80
PDS-RAM	55	SVC-Aktualisierungen für Integrated Debugger ..	81
Skeleton-RAM	56	TCP/IP-Aktualisierungen für Integrated Debugger	81
SCLM-RAM	56	Sicherheitsupdates für Integrated Debugger . .	82
CRACFG, CRASCL, CRASHOW und CRATMAP		Language Environment-Aktualisierungen für Integ-	
- Konfigurationsdateien für den CA Endeavor		rated Debugger	82
SCM-RAM.	56	CICS-Aktualisierungen für Integrated Debugger ..	83
CRACFG, Interaktion von CA Endeavor SCM-		IMS-Aktualisierungen für Integrated Debugger ..	84
RAM mit dem SCM	57	Gespeicherte DB2-Prozeduraktualisierungen für In-	
CRASCL - Schablonen-SCL für den CA Ende-		tegrated Debugger	84
vor SCM-RAM	57	ELAXF*-Aktualisierungen für Integrated Debugger	85
CRASHOW - CA Endeavor SCM-RAM-Stan-			
dardfilter	57	Kapitel 6. Hostbasierte Codeanalyse ..	87
CRATMAP - Zuordnungen von CA Endeavor		Voraussetzungen und Prüfliste	87
SCM-RAM-Dateierweiterungen.	57	Codeüberprüfung	87
CRANDVRA und CRADYNDA - Zuordnungs-		Verarbeitung der Codeüberprüfung ändern. ..	88
Execs für den CA Endeavor SCM-RAM	58	Codeabdeckung	88
CRANDVRA - Anfangszuordnungs-Exec für		Einzelaufruf der Codeabdeckung	88
den CA Endeavor SCM-RAM.	58	Mehrfachaufrufe der Codeabdeckung. . . .	89
CRADYNDA - Neuzuordnungs-Exec für den		Ausgabe der Codeabdeckung	89
CA Endeavor SCM-RAM	59		
Batch-Aktionen für CA Endeavor SCM-RAM ..	59	Kapitel 7. Weitere Anpassungstasks ..	91
CRABCFG - Konfiguration der Batch-Aktion		include.conf: Erzwungene Includes für C/C++-	
für den CA Endeavor SCM-RAM	60	Content-Assist	91
CRABATCA - JCL für Batch-Aktionen des CA		z/OS UNIX-Unterprojekte	92
Endeavor SCM-RAM	61	REXEC- oder SSH-Konfiguration	93
CRABJOBC - Jobkarte für Batch-Aktionen des		Unterstützung für Include-Vorprozessor	93
CA Endeavor SCM-RAM	62	xUnit-Unterstützung für Enterprise COBOL und	
CRAALLOC - Angepasste Zuordnungs-Exec für		PL/I.	94
den RAM	62	Enterprise Service Tools-Unterstützung	95
CARMA-Rückkehrcodes	63	Unterstützung bidirektionaler Sprachen für CICS..	95
Unterstützung mehrerer RAM (optional) . . .	63	IRZ-Diagnosenachrichten für Enterprise Service	
Beispiel.	63	Tools	96
Angepasste Zuordnungs-Exec (optional). . .	65	FEKRNPILI - ferner PL/I-Vorprozessor	97
CARMA-Benutzerexit (optional)	65		
IRXJCL oder CRAXJCL (optional)	66	Kapitel 8. Installationsprüfung. . . .	99
CRAXJCL erstellen	67	Gestartete Tasks prüfen	99
		DBGMR, Debug Manager	99
Kapitel 4. SCLM Developer Toolkit	69	Services prüfen	99
Voraussetzungen und Prüfliste	69	Installationsprüfprogramm initialisieren . .	100
Voraussetzungen	70	CARMA-Verbindung	100
Aktualisierung von ISPF.conf für SCLMDT . .	70	SCLMDT-Verbindung.	101
Aktualisierung von 'rdz.env' für SCLMDT . .	71		
Umsetzung langer/kurzer Namen (optional) .	72	Kapitel 9. Sicherheitsdefinitionen . .	103
LSTRANS.FILE - VSAM-Datei für die Umsetzung		Voraussetzungen und Prüfliste	103
langer/kurzer Namen - erstellen	72	Sicherheitseinstellungen und -klassen aktivieren	104
Aktualisierung von 'rdz.env' für die Umsetzung		Gestartete Tasks für Developer for z Systems defi-	
langer/kurzer Namen	74	nieren	104
Ant installieren und anpassen (optional). .	74		

Debug Manager als sicheren z/OS UNIX-Server definieren	105	Konfigurierbare Dateien	138
Programmgesteuerte MVS-Bibliotheken für Debug Manager definieren	105	Migration von Version 8.5 auf Version 9.0	138
Zugriff auf Integrated Debugger definieren	106	IBM Rational Developer for z Systems - FMID HHOP900	138
Dateiprofile definieren	106	Konfigurierbare Dateien	140
Sicherheitseinstellungen prüfen	107	IBM Rational Developer for z Systems Host Utilities - FMID HAKG900	146
		Konfigurierbare Dateien	146
Kapitel 10. Leitfaden für die Migration	109	Kapitel 11. Hostkonfigurationsreferenz	149
Aspekte der Migration	109	Wissenswertes zu Developer for z Systems	149
Zuvor konfigurierte Dateien sichern	109	Sicherheitsaspekte	149
Migration von Version 9.5 auf Version 9.5.1	110	TCP/IP-Aspekte	149
IBM Rational Developer for z Systems - FMID HHOP951	111	Hinweise zu WLM	149
IBM Rational Developer for z Systems Host Utilities - FMID HAKG951	118	Hinweise zu Push-to-Client.	149
Konfigurierbare Dateien	118	CICSTS-Aspekte	149
Migration von Version 9.1 auf Version 9.5	119	AT-TLS konfigurieren.	149
IBM Rational Developer for z Systems - FMID HHOP950	119	Literaturübersicht	151
Konfigurierbare Dateien	122	Referenzierte Veröffentlichungen	151
IBM Rational Developer for z Systems Host Utilities - FMID HAKG950	127	Veröffentlichungen mit weiteren Informationen ..	152
Konfigurierbare Dateien	127	Bemerkungen	155
Migration von Version 9.0 auf Version 9.1	129	Informationen zu Programmierschnittstellen	157
IBM Rational Developer for z Systems - FMID HHOP910	129	Marken	157
Konfigurierbare Dateien	131	Nutzungsbedingungen für die Produktdokumentation.	157
IBM Rational Developer for z Systems Host Utilities - FMID HAKG910	137	Copyrightlizenz	158
		Marken	158

Abbildungen

1. DBGMGR: Gestartete Task des Debug Managers	24	13. CRASUB*: CARMA-Start mit Batchübergabe	53
2. Umgebungskonfigurationsdatei 'rdz.env'	27	14. CRACFG - Interaktion von CA Endeavor SCM-RAM mit dem SCM.	57
3. CRASRV.properties: CRASTART mit CA Endeavor SCM-RAM	36	15. CRASHOW - CA Endeavor SCM-RAM-Standardfilter	57
4. crastart.endeavor.conf: CRASTART mit CA Endeavor SCM-RAM	37	16. CRATMAP: CA Endeavor SCM-RAM-Standardfilter	58
5. CRASRV.properties: CRASTART mit Muster-RAM.	39	17. CRABCFG: Konfiguration der Batch-Aktion für den CA Endeavor SCM-RAM	60
6. crastart.conf: CRASTART mit Muster-RAM	39	18. CRABATCA: JCL für die Batch-Aktion des CA Endeavor SCM-RAM.	61
7. CRASRV.properties: Batchübergabe mit CA Endeavor SCM-RAM.	40	19. CRABJOBC: Jobkarte für Batch-Aktionen des CA Endeavor SCM-RAM	62
8. CRASUBCA: Batchübergabe mit CA Endeavor SCM-RAM.	41	20. Aktualisierung von ISPF.conf für SCLMDT	71
9. CRASRV.properties: Batchübergabe mit Muster-RAM	43	21. Aktualisierung von 'rdz.env' für SCLMDT	71
10. CRASUBMT: Batchübergabe mit Muster-RAM	44	22. FLM02LST: Konfigurations-JCL für Umsetzung langer/kurzer Namen	73
11. CARMA-Konfigurationsdatei CRASRV.properties	45	23. "include.conf": Erzwungene Includes für C/C++-Content-Assist	92
12. crastart*.conf: CARMA-Serverstart mit CRA-START	50		

Tabellen

1.	Erforderliche Ressourcen	6	15.	Installationsprüfprogramme für Services	99
2.	Optionale Ressourcen	6	16.	Variablen der Sicherheitskonfiguration	103
3.	Administratoren für erforderliche Tasks . ..	7	17.	Anpassungen in Version 9.5.1	113
4.	Administratoren für optionale Tasks.	7	18.	Anpassungen in Version 9.5.1 von z/OS Ex- plorer	117
5.	Clientprüfliste: obligatorischer Teil	12	19.	Anpassungen in Host Utilities Version 9.0	119
6.	Clientprüfliste: optionaler Teil	13	20.	Anpassungen in Version 9.5.	122
7.	Zuordnung von Lademodulen zu Funktionen	20	21.	Anpassungen in Host Utilities Version 9.0	127
8.	ELAXF*-Beispielprozeduren	24	22.	Anpassungen in Version 9.1.0	132
9.	Prüfliste der übergeordneten Qualifikations- merkmale in ELAXF*	26	23.	Anpassungen in Host Utilities Version 9.0	138
10.	ELAXF*	26	24.	Anpassungen in Version 9.0.	141
11.	Automatische Wiederherstellung der Verbin- dung zu Debug Manager	28	25.	Anpassungen in Host Utilities Version 9.0	146
12.	CARMA-Rückkehrcodes	63	26.	Referenzierte Veröffentlichungen	151
13.	Prüfliste für den SCLM-Administrator . . .	75	27.	Referenzierte Websites	152
14.	Dynamische SVC-Aktualisierung für Integrated Debugger	81	28.	Veröffentlichungen mit weiteren Informatio- nen	152

Zu diesem Handbuch

Dieses Handbuch beschäftigt sich mit der Konfiguration der Funktionen von IBM® Rational Developer for z Systems. Es enthält Konfigurationsanweisungen für IBM Rational Developer for z Systems Version 9.5.1 auf Ihrem z/OS-Hostsystem.

Im weiteren Verlauf dieses Handbuchs werden die folgenden Namen verwendet:

- *IBM Explorer for z/OS* wird als *z/OS Explorer* bezeichnet.
- *IBM Rational Developer for z Systems* wird als *Developer for z Systems* bezeichnet.
- *IBM Rational Developer for z Systems Integrated Debugger* wird als *Integrated Debugger* bezeichnet.
- *IBM Rational Developer for z Systems Interface for CA Endeavor® SCM* wird als *CA Endeavor® SCM-RAM* 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 z Systems beschrieben wird. Jedes dieser Dokumente hat eine spezielle Zielgruppe. Zur Durchführung der Developer for z Systems-Konfiguration müssen Sie nicht alle Dokumente lesen.

- In der *IBM Rational Developer for z Systems Hostkonfiguration* (IBM Form SC27-8577) werden alle Planungstasks, Konfigurationstasks und Optionen (einschließlich der optionalen) ausführlich beschrieben und alternative Szenarios bereitgestellt.
- In der *IBM Rational Developer for z Systems Hostkonfigurationsreferenz* (IBM Form SC27-8578) wird das Design von Developer for z Systems beschrieben. Das Handbuch enthält außerdem Hintergrundinformationen für verschiedene Konfigurationstasks von Developer for z Systems, z/OS-Komponenten und weitere Produkte (wie WLM und CICS) in Verbindung mit Developer for z Systems.

Die Informationen in diesem Dokument gelten für alle Pakete von IBM Rational Developer for z Systems Version 9.5.

Die aktuellsten Versionen dieses Dokuments finden Sie im Handbuch *IBM Rational Developer for z Systems Hostkonfiguration* (IBM Form SC12-4062) unter '<http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC27-8577>'.

Die aktuellsten Versionen der kompletten Dokumentation, einschließlich Installationsanweisungen, White Papers, Podcasts und Lernprogrammen, finden Sie auf der Bibliotheksseite der IBM Rational Developer for z Systems-Website (http://www-01.ibm.com/software/sw-library/en_US/products/Z964267S85716U24/).

Zielgruppe

Das vorliegende Dokument wendet sich an Systemprogrammierer, die IBM Rational Developer for z Systems Version 9.5.1 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 der *IBM Rational Developer for z Systems Hostkonfigurationsreferenz* (IBM Form SC27-8578). 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 z Systems Version 9.5.1 Hostkonfiguration* (IBM Form SC27-8577-00) zusammengefasst (aktualisiert im Dezember 2015).

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 9.5.1
- Neue und entfernte Anweisungen in `rdz.env`

Entfernte Informationen:

In Version 9.5.1 wurden die zu RSE und JES Job Monitor gehörigen Funktionen von IBM Rational Developer for z Systems in ein anderes Produkt (IBM Explorer for z/OS) verschoben. Dies gilt auch für die zugehörige Dokumentation.

- Daten, die sich auf RSE beziehen, werden aus allen Kapiteln entfernt.
- Daten, die sich auf JES Job Monitor beziehen, werden aus allen Kapiteln entfernt.
- Daten, die sich auf den TSO-Befehlsservice beziehen, werden aus allen Kapiteln entfernt.
- "Push-to-Client"-Daten werden aus allen Kapiteln entfernt.
- Die Dokumentation zu Bedienerbefehlen wird entfernt.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for z Systems Version 9.5 Hostkonfiguration* (IBM Form SC23-7658-14) enthalten waren.

Neue Informationen:

- Migrationsinformationen für Version 9.5.1.
- Migrationsinformationen für Version 9.5.
- Neue und entfernte Anweisungen in `rdz.env`.
- Bedienerbefehle `DEBUG STORAGE` und `SEND`.
- Unterstützung für das Senden von Nachrichten.
- Unterstützung für Interactive ISPF Gateway.
- PL/I-Vorprozessor.

Informationen, die an eine andere Stelle versetzt wurden:

- Die Informationen zu Integrated Debugger wurden in ein anderes Kapitel versetzt.

Entfernte Informationen:

- Application Deployment Manager wird nicht mehr bereitgestellt. Daher wurden sämtliche Informationen bezüglich Application Deployment Manager entfernt.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for z Systems Version 9.1.1 Hostkonfiguration (SC12-4062)* enthalten waren.

Neue Informationen:

- Geänderte Definition der gestarteten DBGMGR-Task.
- Neue Sicherheitskonfiguration für Integrated Debugger.
- Neuer Status "rejectLogon" für RSE-Thread-Pool beim Befehl "Modify Display Process".
- Neue und entfernte Anweisungen in `rdz.env`.
- Geänderte Konfigurationsanweisungen für den optionalen Integrated Debugger.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for z Systems Version 9.1 Hostkonfiguration (IBM Form SC12-4062-12)* enthalten waren.

Neue Informationen:

- Migrationsinformationen für Version 9.1.0.
- Neue Optionen für die Definition der gestarteten RSED-Task.
- Neue und geänderte Anweisungen in '`rdz.env`'.
- Neue Bedienerbefehle.
- Neue Optionen für Integrated Debugger.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for z Systems Version 9.0.1 Hostkonfiguration (IBM Form SC12-4062-11)* enthalten waren.

Neue Informationen:

- Zusätzliche Konfigurationsanweisungen für den optionalen Integrated Debugger.

Dieses Dokument enthält Informationen, die zuvor im Handbuch *IBM Rational Developer for z Systems Version 9.0.1 Hostkonfiguration (IBM Form SC12-4062-10)* enthalten waren.

Neue Informationen:

- Migrationsinformationen für Version 9.0.1.
- Neue optionale Aktualisierungen von PARMLIB.
- Neue optionale gestartete Task.
- Neue Bedienerbefehle.
- Neue und geänderte Anweisungen in `rdz.env`
- Neue Anweisungen in "`CRASRV.properties`".
- Neue Konfigurationsdatei für CARMA CA Endeavor® SCM-RAM.
- Unterstützung eines beim CARMA-Start aufgerufenen Benutzerexits.

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

Neue Informationen:

- Migrationsinformationen für Version 9.0.
- Neue und geänderte Anweisungen in `rdz.env`.
- Neue und entfernte Anweisungen in FEJJCNFG.
- Neue ELAXF*-PROCLIB-Member.
- Neue JMON- und RSED-Bedienerbefehle.
- Hinzugefügte Informationen zur hostbasierten Codeanalyse.

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 Buildprozedur 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 z Systems Version 8.5.1 Hostkonfiguration* (IBM Form SC12-4062-08) enthalten waren.

Neue Informationen:

- Neue und geänderte Anweisungen in `'rdz.env'`.
- Erweiterte Unterstützung für Batch-Aktionen für CA Endevor® SCM-RAM.
- Anforderungen für die ID des Installationsbenutzers.

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

Neue Informationen:

- Neue optionale Anweisungen in FEJJCNFG. Weitere Informationen finden Sie in der Veröffentlichung FEJJCNFG - Konfigurationsdatei für JES Job Monitor.
- Neue optionale Anweisungen in `'rdz.env'`.
- Neue optionale Konfigurationsdatei.
- Neue optionale Komponente.
- Neue optionale Komponente.
- Neue optionale Komponente.
- Neue und erweiterte Bedienerbefehle.

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 z Systems die Informationen in diesem Kapitel.

Basisanpassung

Die folgenden Anpassungsschritte beziehen sich auf eine Basiskonfiguration von Developer for z Systems.

- „Anpassungskonfiguration“ auf Seite 15
- „PARMLIB-Änderungen“ auf Seite 16
- „PROCLIB-Änderungen“ auf Seite 23
- „Sicherheitsdefinitionen“ auf Seite 26
- „Die Umgebungskonfigurationsdatei 'rdz.env'“ auf Seite 27

Common Access Repository Manager (CARMA)

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 z Systems 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 z Systems gibt Developer for z Systems-Clients direkten Zugriff auf CA Endevor® SCM.

SCLM Developer Toolkit

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.

Integrated Debugger

Mit der Hostkomponente Developer for z Systems Integrated Debugger können Clients bei verschiedenen LE-basierten Anwendungen (LE - Language Environment) Fehler beheben.

Hostbasierte Codeanalyse

Ähnlich wie der Developer for z Systems-Client unterstützt der Developer for z Systems-Host die Ausführung von Codeanalysetools, die als separates Produkt unter der Bezeichnung IBM Rational Developer for z Systems 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

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 z Systems:

- include.conf: Erzwungene Includes für C/C++-Content-Assist

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

- 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 Enterprise Service Tools
- FEKRNPLI - ferner PL/I-Vorprozessor

Installationsprüfung

Nach Abschluss der Produktanpassung 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.

Hostkonfigurationsreferenz

In diesem Abschnitt werden die Informationen im Handbuch *Rational Developer for z Systems Hostkonfigurationsreferenz* (IBM Form SC27-857) zusammengefasst.

IBM Rational Developer for z Systems Hostkonfiguration

Kapitel 1. Planung

Nutzen Sie für die Planung der Installation und der Implementierung von Developer for z Systems die Informationen in diesem Kapitel sowie die Softwarevoraussetzungen von Developer for z Systems einschließlich der Angaben über vorausgesetzte und zusätzlich erforderliche Produkte. Die folgenden Themen werden behandelt:

- „Hinweise zur Migration“
- „Hinweise zur Planung“
- „Hinweise zur Installationsvorbereitung“ auf Seite 5
- „Hinweise zur Konfigurationsvorbereitung“ auf Seite 9
- „Hinweise zur Implementierungsvorbereitung“ auf Seite 11
- „Clientprüfliste“ auf Seite 12

Eine vollständige Liste der Hardware- und Softwarevoraussetzungen für Developer for z Systems einschließlich vorausgesetzter und zusätzlich erforderlicher Produkte finden Sie auf der Registerkarte **System Requirements** auf der Produktseite von Developer for z Systems (<http://www-03.ibm.com/software/products/en/developerforsystemz/>).

Hinweise zur Migration

Kapitel 10, „Leitfaden für die Migration“, auf Seite 109 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 z Systems.

Anmerkung:

- In Version 9.5.1 wurde Developer for z Systems als Produkterweiterung von IBM Explorer for z/OS installiert. Stellen Sie sicher, dass Sie über eine unterstützte Version von IBM Developer for z/OS verfügen.
- Wenn Sie mit einer früheren Version von IBM Rational Developer for System z gearbeitet haben, sichern Sie die zugehörigen angepassten Dateien, bevor Sie diese Version von IBM Rational Developer for z Systems installieren. Eine Übersicht über die Dateien, die eine Anpassung erforderten, finden Sie im Kapitel 10, „Leitfaden für die Migration“, auf Seite 109.
- Wenn Sie planen, mehrere Instanzen von Developer for z Systems auszuführen, lesen Sie sich die Informationen zur "Ausführung mehrerer Instanzen" in der *IBM Rational Developer for z Systems Hostkonfigurationsreferenz (IBM Form SC27-8578)* durch.

Hinweise zur Planung

Produktübersicht

Developer for z Systems besteht aus einem Client, der auf dem Personal Computer des Benutzers installiert ist, und einem Server, der auf mindestens einem Hostsys-

tem installiert ist. Sowohl Client als auch Host werden zusätzlich zu IBM Explorer for z/OS installiert. Diese Dokumentation enthält Informationen für ein z/OS-Hostsystem.

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 z Systems erweitert die von z/OS Explorer bereitgestellte grundlegende Zugriffsfunktionalität. Developer for z Systems kann beispielsweise mit Subsystemen und anderer Anwendungssoftware auf dem Hostsystem interagieren, wie CICS und Software Configuration Managers (SCMs), wenn Developer for z Systems entsprechend konfiguriert ist und wenn diese zusätzlich erforderlichen Produkte verfügbar sind.

Informationen zu Developer for z Systems und zur Interaktion des Produkts mit Ihrem System sowie mit den vorausgesetzten und zusätzlich erforderlichen Produkten finden Sie in der *Rational Developer for z Systems Hostkonfigurationsreferenz* (IBM Form SC27-8578). Das Handbuch *IBM Explorer for z/OS Host Configuration Reference* (IBM Form SC27-8438) enthält ähnliche Informationen für das Produkt z/OS Explorer, das eine Voraussetzung für Developer for z Systems ist.

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

Erforderliche Qualifikationen

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

Die Konfiguration von Developer for z Systems 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 7 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 z Systems benötigte Zeit hängt von verschiedenen Faktoren ab. Dazu gehören:

- Die aktuelle z/OS UNIX- und TCP/IP-Konfiguration
- Die Verfügbarkeit von Softwarevoraussetzungen und Wartung
- 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 z Systems-Hostsystem einen Zeitaufwand von ein bis zwei Tagen erfordert. Dabei ist die Installation und Konfiguration von IBM Explorer for z/OS, einem erforderlichen Produkt, nicht berücksichtigt. 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 z Systems* (GI11-8298).

Die Developer for z Systems-Server sind einzelsystemorientiert und wurden nicht für ein SYSPLEX konzipiert. Wenn Sie die Server in einem SYSPLEX verwenden, müssen Sie sicherstellen, dass die von den Endbenutzern angeforderten Daten (Dateien, Jobausgaben, z/OS UNIX-Dateien) auf dem System verfügbar sind, auf dem Developer for z Systems installiert ist. Informationen zum Klonen von Developer for z Systems für andere Systeme finden Sie unter „Hinweise zur Implementierungsvorbereitung“ auf Seite 11.

Informationen zur Ausführung mehrerer Instanzen von Developer for z Systems auf einem Einzelhostsystem finden Sie unter "Ausführung mehrerer Instanzen" in der *Rational Developer for z Systems Hostkonfigurationsreferenz* (IBM Form SC27-8578).

Das Dateisystem (HFS oder zFS), in dem Developer for z Systems 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 z Systems keine Sicherheitsumgebung für den Benutzer erstellen, sodass die Verbindungsanforderungen des Clients zurückgewiesen werden. Dasselbe gilt für die Dateisysteme, die das Hosting für z/OS Explorer-, Java- und z/OS UNIX-Binärprogramme übernehmen.

ID des Installationsbenutzers

Die Benutzer-ID, die zur Installation von Developer for z Systems oder zur Installation einer Wartung verwendet wird, 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 erforderlich, 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.
 - 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

Developer for z Systems enthält eine Liste der vorausgesetzten Software, die installiert und betriebsbereit sein muss, damit das Produkt ordnungsgemäß funktionieren kann. Außerdem gibt es eine Liste zusätzlich erforderlicher Software zur Unterstützung bestimmter Features von Developer for z Systems. Diese zusätzlich erforderliche Software muss zur Laufzeit installiert und betriebsbereit sein, damit die entsprechenden Features ordnungsgemäß funktionieren.

Eine vollständige Liste der Softwarevoraussetzungen für Developer for z Systems einschließlich vorausgesetzter und zusätzlich erforderlicher Produkte finden Sie auf der Registerkarte **System Requirements** auf der Website von Developer for z Systems (<http://www-03.ibm.com/software/products/en/developerforsystemz/>).

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 1.12 oder höher
- IBM Explorer for z/OS 3.0
- Neuestes Service-Release von Java ab Version 6.0 (31- oder 64-Bit)

Erforderliche Ressourcen

Developer for z Systems erfordert die Reservierung der in Tabelle 1 aufgelisteten Systemressourcen. Die in Tabelle 2 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
MVS-Buildprozeduren	ELAXF*	„PROCLIB-Änderungen“ auf Seite 23

Tabelle 2. Optionale Ressourcen

Ressource	Standardwert	Informationen
Einleitendes Programmladen mit CLPA	Nicht anwendbar	Kapitel 5, „Integrated Debugger“, auf Seite 77
Gestartete Task	DBGMGR	Kapitel 5, „Integrated Debugger“, auf Seite 77
LINKLIST-Dateigruppe	FEL.SFELAUTH und FEL.SFELLOAD	<ul style="list-style-type: none"> • Kapitel 4, „SCLM Developer Toolkit“, auf Seite 69 • Kapitel 5, „Integrated Debugger“, auf Seite 77

Tabelle 2. Optionale Ressourcen (Forts.)

Ressource	Standardwert	Informationen
LPA-Datei	FEL.SFELLPA	<ul style="list-style-type: none"> • „LPA-Definitionen in LPALSTxx“ auf Seite 19 • Kapitel 3, „Common Access Repository Manager (CARMA)“, auf Seite 33 • Kapitel 5, „Integrated Debugger“, auf Seite 77
Sicherheitsprofile	AQE.**	<ul style="list-style-type: none"> • Kapitel 5, „Integrated Debugger“, auf Seite 77
Portbereich für die hostinterne Verwendung	Jeder verfügbare Port kann verwendet werden.	<ul style="list-style-type: none"> • Kapitel 3, „Common Access Repository Manager (CARMA)“, auf Seite 33 • Kapitel 6, „Hostbasierte Codeanalyse“, auf Seite 87
Portbereich für die hostinterne Verwendung	5336	Kapitel 5, „Integrated Debugger“, auf Seite 77
Port für die Kommunikation zwischen Client und Host	<ul style="list-style-type: none"> • 5335 für Integrated Debugger 	<ul style="list-style-type: none"> • Kapitel 5, „Integrated Debugger“, auf Seite 77
Update für die CICS-Systemdefinition	Mehrere Werte	<ul style="list-style-type: none"> • Kapitel 5, „Integrated Debugger“, auf Seite 77
Update für CICS-JCL	<ul style="list-style-type: none"> • FEL.SFELLOAD • FEL.SFELAUTH 	<ul style="list-style-type: none"> • „Unterstützung bidirektionaler Sprachen für CICS“ auf Seite 95 • Kapitel 5, „Integrated Debugger“, auf Seite 77

Die Konfiguration von Developer for z Systems 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 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.	-

Tabelle 4. Administratoren für optionale Tasks

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

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-Profile für Debugging definieren • Zertifikat für verschlüsselte Kommunikation hinzufügen • Gestartete Tasks definieren • z/OS UNIX-Serverprofile definieren • Profile für Debugging definieren • Profile zum Senden von Nachrichten definieren 	<ul style="list-style-type: none"> • "Sicherheitsaspekte" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC12-4489-02)
TCP/IP-Administrator	Neue TCP/IP-Ports definieren	"TCP/IP-Ports" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC27-8578)
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“, auf Seite 69
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"> • „Unterstützung bidirektionaler Sprachen für CICS“ auf Seite 95 • Kapitel 5, „Integrated Debugger“, auf Seite 77
WLM-Administrator	<ul style="list-style-type: none"> • Ziele für Developer for z Systems-Tasks zuweisen 	<ul style="list-style-type: none"> • "Hinweise zu WLM" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC27-8578) (IBM Form SC12-4489-02)
LDAP	Gruppen für Push-to-Client definieren	"Hinweise zu Push-to-Client" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC27-8578)

Hinweise zur Konfigurationsvorbereitung

Informationen zu Developer for z Systems und zur Interaktion des Produkts mit Ihrem System sowie mit den vorausgesetzten und zusätzlich erforderlichen Produkten finden Sie im Handbuch *Rational Developer for z Systems Hostkonfigurationsreferenz* (IBM Form SC27-8578). Das Handbuch *IBM Explorer for z/OS Host Configuration Reference* (IBM Form SC27-8438) enthält ähnliche Informationen für das Produkt z/OS Explorer, das eine Voraussetzung für Developer for z Systems ist. 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 z Systems keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for z Systems 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 SC27-8578).

Anmerkung: Developer for z Systems 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 z Systems können Zeitlimitprobleme auftreten (weil während des Zeitlimitfensters zu wenig CPU-Zeit verfügbar ist).

Ressourcennutzung und Systembegrenzungen

Developer for z Systems 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 SC27-8578). Developer for z Systems 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 z Systems-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 z Systems-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 erforderlich, 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
FILEPROCMAX= NONE
PROCUSERMAX= 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 z/OS Explorer und Developer for z Systems (standardmäßig /usr/lpp/ibm/zexpl/*, /etc/zexpl/*, /var/zexpl/*, /usr/lpp/ibm/rdz/* und /var/rdz/*).
- Zugriffsrechte READ, WRITE und EXECUTE für das Verzeichnis WORKAREA von z/OS Explorer (Standard: /var/zexpl/WORKAREA) und für das Benutzerprotokollverzeichnis (Standard: /var/zexpl/logs).
- Zugriffsrecht READ für die Installationsdateien von z/OS Explorer (standardmäßig FEK.SFEK* und FEL.SFEL*).
- Zugriffsrechte READ, WRITE und EXECUTE für das Verzeichnis /tmp oder ein Verzeichnis, auf das in der Umgebungsvariablen TMPDIR verwiesen wird.

Hinweise zum Server

z/OS Explorer und Developer for z Systems bestehen 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 z/OS Explorer- und Developer for z Systems-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. JMON ist eine Komponente von z/OS Explorer.
- RSE (Remote Systems Explorer) stellt Kernservices wie den Verbindungsaufbau vom Client zum Hostsystem und das Starten anderer Server für bestimmte Services bereit. RSE ist eine Komponente von z/OS Explorer. 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 SC27-8578) 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 z Systems verwendeten Ports gibt es nur Hostdatenverkehr. Nachfolgend sind die Ports aufgelistet, die für die externe Kommunikation in einer Basiskonfiguration von Developer for z Systems 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.

Hinweise zur Implementierungsvorbereitung

Developer for z Systems 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 (einschließlich z/OS Explorer).

Developer for z Systems

- FEL.SFELAUTH(*)
- FEL.SFELLMOD(*)
- FEL.SFELLOAD(*)
- FEL.SFELLPA(*)
- FEL.SFELPROC(*)
- FEL.#CUST.CNTL(*)
- FEL.#CUST.PARMLIB(*)

- FEL.#CUST.PROCLIB(*)
- /usr/lpp/ibm/rdz/*
- /var/rdz/*
- Definitionen, Dateien und Verzeichnisse im Ergebnis der Anpassung von Jobs in 'FEL.#CUST.JCL'

Developer for z Systems Host Utilities

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

Anmerkung:

- FEK und /usr/lpp/ibm/rdz sind das während der Installation von Developer for z Systems verwendete übergeordnete Qualifikationsmerkmal und der Pfad. FEL.#CUST, /etc/zexpl und /var/rdz sind die während der Anpassung des Produkts verwendeten Standardpositionen, wobei /etc/zexpl das z/OS Explorer-Konfigurationsverzeichnis ist.
- AKG und /usr/lpp/ibm/rdzutil sind das während der Installation von Developer for z Systems Host Utilities verwendete übergeordnete Qualifikationsmerkmal und der Pfad.

- Sie sollten Developer for z Systems in einem privaten Dateisystem (HFS oder zFS) und optional mit z/OS Explorer kombiniert 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 z Systems beibehalten.

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

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

Clientprüfliste

Benutzer der Clientkomponente von Developer for z Systems 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 auf Seite 13 sind die erforderlichen Ergebnisse optionaler Anpassungsschritte aufgelistet.

Tabelle 5. Clientprüfliste: obligatorischer Teil

Anpassung	Wert
(Voraussetzung) TCP/IP-Portnummer des RSE-Dämons. Die Standardeinstellung ist 4035.	
Dieser Port wird während der Konfiguration von z/OS festgelegt.	

Tabelle 6. Clientprüfliste: optionaler Teil

Anpassung	Wert
Position der ELAXF*-Prozeduren, falls sie nicht in einer Systemprozedurbibliothek enthalten sind. Die Standardeinstellung ist FEL.#CUST.PROCLIB. Lesen Sie die Anmerkung zu JCLLIB in „ELAXF* - ferne Buildprozeduren“ auf Seite 24.	
Namen der ELAXF*-Prozeduren oder der zugehörigen Prozedurschritte, sofern sie geändert wurden. Lesen Sie die Anmerkung zur Änderung der Namen im Abschnitt „ELAXF* - ferne Buildprozeduren“ auf Seite 24.	
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 87.	
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 88.	
Position der Exec FEKRNPLI des Include-Vorprozessors. Die Standardeinstellung ist FEL.#CUST.CNTL. Lesen Sie hierzu „Unterstützung für Include-Vorprozessor“ auf Seite 93.	
Position der Debugger-Lademodule, falls diese nicht in LINKLIST enthalten sind. Die Standardeinstellung ist 'FEL.SFELAUTH'. Siehe Kapitel 5, „Integrated Debugger“, auf Seite 77	
Position der Lademodule für den Komponententext, falls diese nicht in LINKLIST oder STEPLIB von rdz.env enthalten sind. Die Standardeinstellung ist FEL.SFELLOAD. Lesen Sie hierzu „xUnit-Unterstützung für Enterprise COBOL und PL/I“ auf Seite 94.	
Position der AZUZUNIT-Prozedur, sofern sie nicht in einer Systemprozedurbibliothek enthalten ist. Die Standardeinstellung ist FEL.#CUST.PROCLIB. Lesen Sie die Anmerkung zu JCLLIB im Abschnitt „xUnit-Unterstützung für Enterprise COBOL und PL/I“ auf Seite 94.	
Position der Beispiel-XML-Dateien *.xsd und *.xsl, die für die Formatierung von Komponententestausgaben verwendet werden. Die Standardwerte sind /usr/lpp/ibm/rdz/samples/zunit/xsd und /usr/lpp/ibm/rdz/samples/zunit/xsl. Lesen Sie hierzu „xUnit-Unterstützung für Enterprise COBOL und PL/I“ auf Seite 94.	
TN3270-Portnummer für den Host-Connect-Emulator (zusätzliche Voraussetzung). Die Standardeinstellung ist 23. Siehe "TCP/IP-Ports" in der <i>Hostkonfigurationsreferenz</i> (IBM Form SC43-2898).	

Tabelle 6. Clientprüfliste: optionaler Teil (Forts.)

Anpassung	Wert
REXEC- oder SSH-Portnummer. Standardwert: 512 bzw. 22 (zusätzliche Voraussetzung). Lesen Sie hierzu „z/OS UNIX-Unterprojekte“ auf Seite 92.	
Portnummer des Debug Tool-Servers (zusätzliche Voraussetzung; ohne Standardwert). Siehe Unterstützung für DB2- und IMS-Debug (optional).	
Position der Beispielbibliothek SFELSAMP für CARMA-RAM-Muster. Der Standardwert ist FEL.SFELSAMP. Siehe <i>CARMA Developer's Guide</i> (IBM Form SC23-7660).	
Position der CRA#ASLM-JCL für das Zuordnen von CARMA-SCLM-RAM-Dateien. Der Standardwert ist FEL.#CUST.JCL. Lesen Sie die Anmerkung zu CRA#ASLM in „SCLM-RAM“ auf Seite 56.	

Kapitel 2. Basisanpassung

Die folgenden Anpassungsschritte beziehen sich einheitlich auf die verschiedenen Developer for z Systems-Services. Informationen zu den Voraussetzungen für die Anpassung bestimmter 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:

- LPA-Datei
- Datei mit APF-Berechtigung
- Verschiedene Aktualisierungen von PARMLIB
- Verschiedene Aktualisierungen der Sicherheitssoftware
- Unterschiedliche TCP/IP-Ports für interne und Client-Host-Kommunikation
- (Optional) IPL zum Aktivieren von SVCs

Führen Sie zur Überprüfung der Installation und zur Verwendung von Developer for z Systems 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 z Systems. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“.
2. Aktualisieren Sie die z/OS UNIX-Systembegrenzungen, aktualisieren Sie die Produktregistrierung, starten Sie gestartete Tasks und definieren Sie APF-autorisierte Dateien und LINKLIST-Dateien sowie optional SVC- und LPA-Dateien. Weitere Details enthält der Abschnitt „PARMLIB-Änderungen“ auf Seite 16.
3. Erstellen Sie gestartete Taskprozeduren sowie Kompilierungs- und Verknüpfungsprozeduren. Weitere Details enthält der Abschnitt „PROCLIB-Änderungen“ auf Seite 23.
4. Aktualisieren Sie die Sicherheitsdefinitionen. Weitere Details enthält der Abschnitt „Sicherheitsdefinitionen“ auf Seite 26.
5. Passen Sie die Konfigurationsdateien für Developer for z Systems an. Weitere Details enthalten die Abschnitte
 - „Die Umgebungskonfigurationsdatei 'rdz.env'“ auf Seite 27

Anpassungskonfiguration

In Developer for z Systems 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 z Systems 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 FELSETUP bereit, mit dem Sie die Kopien und die erforderlichen Verzeichnisse erstellen können.

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

Dieser Job führt die folgenden Tasks aus:

- FEL.#CUST.PARMLIB erstellen und mit Beispielkonfigurations-JCL füllen
- FEL.#CUST.PROCLIB erstellen und mit SYS1.PROCLIB-Beispiel-Membert füllen
- FEL.#CUST.JCL erstellen und mit Beispielkonfigurations-JCL füllen
- FEL.#CUST.CNTL erstellen und mit Beispiel-Scripts für den Serverstart füllen
- FEL.#CUST.ASM erstellen und mit Assemblerbeispielquellcode füllen
- FEL.#CUST.SQL erstellen und mit SQL-Beispielbefehlsdateien füllen
- z/OS Explorer-Konfigurationsverzeichnis /etc/zexpl/* mit Beispielkonfigurationsdateien füllen
- /var/rdz/* als Arbeitsverzeichnisse für verschiedene Funktionen von Developer for z Systems erstellen und mit Beispieldateien füllen

Anmerkung:

- Für die Konfigurationsschritte in dieser Veröffentlichung werden die vom Job FELSETUP erstellten Member-Positionen und Dateipositionen verwendet, sofern nichts anderes angegeben ist. Die Originalbeispiele, die nicht aktualisiert werden sollten, befinden sich in FEL.SFELSAMP und /usr/lpp/ibm/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 FEL.SFELSAMP(FELSETUP).
- Zur Unterstützung der Migration einer vorhandenen Konfiguration dokumentieren die Kommentare in FEL.SFELSAMP(FELSETUP) außerdem die Änderungen zwischen verschiedenen Versionen von Rational Developer for z Systems.
- Falls Sie alle z/OS UNIX-Dateien von z/OS Explorer und Developer for z Systems in demselben Dateisystem (HFS oder zFS) behalten wollen, die Konfigurationsdateien aber auch im Verzeichnis /etc/zexpl 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/ibm/rdz/cust) und definieren eine symbolische Verbindung (/etc/zexpl) zu diesem Verzeichnis:

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

PARMLIB-Änderungen

In diesem Abschnitt werden die folgenden PARMLIB-Änderungen dokumentiert:

- „z/OS UNIX-Grenzwerte in BPXPRMxx festlegen“ auf Seite 17
- „Gestartete Tasks zu COMMNDxx hinzufügen“ auf Seite 17
- „SVC-Definitionen in IEASVCxx“ auf Seite 18
- „Produktaktivierung in IFAPRDxx“ auf Seite 18
- „LPA-Definitionen in LPALSTxx“ auf Seite 19
- „APF-Berechtigungen in PROGxx“ auf Seite 19
- „LINKLIST-Definitionen in PROGxx“ auf Seite 20
- „Vorausgesetzte LINKLIST- und LPA-Definitionen“ auf Seite 22
- „LINKLIST-Definitionen für andere Produkte“ auf Seite 23

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

SCLM Developer Toolkit startet während bestimmten Benutzeraktivitäten mehrere benutzereigene Prozesse gleichzeitig.

Definieren Sie OMVS=xx im Parmlib-Member IEASYSxx, um festzulegen, welches Parmlib-Member BPXPRMxx beim einleitenden Programmladen verwendet werden soll.

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 z Systems verwenden.

Diese Werte können mit folgenden Konsolbefehlen überprüft und dynamisch (bis zum nächsten IPL) gesetzt werden:

- DISPLAY OMVS,0
- SETOMVS MAXPROCUSER=50

Anmerkung:

- 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.
- Bei der SMP/E-Installation von Developer for z Systems 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 z Systems 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 zu SYS1.PARMLIB(COMMANDxx) Startbefehle für die Developer for z Systems-Server hinzu, damit sie beim nächsten IPL automatisch gestartet werden. 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 z Systems 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 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

Für den optionalen Integrated Debugger muss ein Supervisoraufruf (SVC) für Developer for z Systems für Ihr System definiert sein.

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. Beachten Sie jedoch, dass mit Integrated Debugger eine dynamische Installation oder Aktualisierung mithilfe der gestarteten Task des Debug Managers möglich ist. 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 z Systems zu definieren:

```
SVCPARM 251,REPLACE,TYPE(3),EPNAME(AQESVC03) /* RDz debug */
```

Der Standardwert ist SVC-Nummer 251. Innerhalb des von z/OS vorgegebenen Bereichs von 200 bis 255 kann jedoch ein beliebiger Wert verwendet werden. Integrated Debugger erkennt die verwendete SVC-Nummer.

Anmerkung: Für Developer for z Systems-Versionen vor Version 9.1.1 wurde ein anderer SVC (AQESVC01) genutzt. Dieser SVC wird nicht mehr verwendet. Er kann gelöscht werden, wenn auf Ihrem Hostsystem nur noch Developer for z Systems ab Version 9.1.1 eingesetzt wird.

Produktaktivierung in IFAPRDxx

Falls Sie Developer for z Systems als Bestandteil von Produktcode 5697-CDT, IBM Enterprise COBOL Suite for z/OS, V1.1 (oder höher) erworben haben, fügen Sie dem PARMLIB-Member IFAPRDxx einen Eintrag hinzu, um die zugehörigen Programme zu aktivieren, sofern Sie dies nicht bereits für ein anderes Programm in Produktcode 5697-CDT getan haben. Definieren Sie PROD=xx im Parmlib-Member IEASYSxx, um anzugeben, welches Parmlib-Member IFAPRDxx beim einleitenden Programmladen verwendet werden soll.

Geben Sie in IFAPRDxx Folgendes an, um Enterprise COBOL Suite for z/OS (Produktcode 5697-CDT) zu definieren:

```
PRODUCT OWNER('IBM CORP')  
NAME('IBM COBOL SUITE')  
ID(5697-CDT)  
VERSION(*) RELEASE(*) MOD(*)  
FEATURENAME(*)  
STATE(ENABLED)
```

Alternativ und optional können Sie dem PARMLIB-Member IFAPRDxx für Developer for z Systems unter Verwendung des eigenständigen Produktcodes (5724-T07) einen Eintrag hinzufügen, falls Sie Developer for z Systems separat (NICHT als Bestandteil von Enterprise COBOL Suite for z/OS) erworben haben.

```
PRODUCT OWNER('IBM CORP')
NAME('IBM RDZ')
ID(5724-T07)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME(*)
STATE(ENABLED)
```

Nach dem Aktualisieren des Parmlib-Members IFAPRDxx kann es mit dem folgenden Konsolenbefehl dynamisch (bis zum nächsten IPL) gestartet werden:

```
SET PROD=xx
```

Anmerkung: Developer for z Systems registriert die folgenden Features:

- RDZ-RSED (für Developer for z Systems)
- RDZ-CC (für Codeabdeckung in Developer for z Systems Host Utilities)
- RDZ-CR (für Codeüberprüfung in Developer for z Systems Host Utilities)

LPA-Definitionen in LPALSTxx

Der optionale Service von Common Access Repository Manager (CARMA) unterstützt unterschiedliche Methoden für den Serverstart des CARMA-Servers. Für die Startmethode 'CRASTART' wird vorausgesetzt, dass sich die Module in der Ladebibliothek FEL.SFELLPA im Link-Pack-Bereich (LPA) befinden.

LPA-Dateien 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=FEL.SFELLPA,MASK=*

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

Zum Ausführen des optionalen Debug Managers müssen die Module in den Ladebibliotheken FEL.SFELAUTH und FEL.SFELLPA über eine APF-Berechtigung verfügen. Hinweis: Die Ladebibliothek FEL.SFELLPA muss sich für den Debug Manager in STEPLIB befinden und erfordert daher eine explizite APF-Berechtigung.

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

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 Programmladen verwendet werden soll.

APF-Berechtigungen können mit den folgenden Konsolenbefehlen dynamisch (bis zum nächsten einleitenden Programmladen) 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=FEL.SFELAUTH,SMS
- SETPROG APF,ADD,DSN=FEL.SFELLPA,SMS
- SETPROG APF,ADD,DSN=REXX.V1R4M0.SEAGLPA,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 vorausgesetzten und zusätzlich erforderlichen Produkte ist ebenfalls eine APF-Berechtigung erforderlich. Dies gilt beispielsweise für z/OS Explorer. Weitere Informationen hierzu finden Sie im Anpassungshandbuch zum jeweiligen Produkt.

LINKLIST-Definitionen in PROGxx

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

- Developer for z Systems-Ladebibliotheken, die für Developer for z Systems-Funktionen benötigt werden. Diese Definitionen werden in diesem Abschnitt beschrieben.
- Vorausgesetzte Ladebibliotheken, die für Developer for z Systems-Funktionen benötigt werden. Diese Definitionen werden im Abschnitt „Vorausgesetzte LINKLIST- und LPA-Definitionen“ auf Seite 22 beschrieben.
- Developer for z Systems-Ladebibliotheken, die von anderen Produkten benötigt werden. Diese Definitionen werden im Abschnitt „LINKLIST-Definitionen für andere Produkte“ auf Seite 23 beschrieben.

Tabelle 7. Zuordnung von Lademodulen zu Funktionen

Ladebibliothek	Lademodule	Verwendung	STEPLIB
FEL.SFELAUTH	AQE* und CEE*	Kapitel 5, „Integrated Debugger“, auf Seite 77	ELAXFGO-Prozedur oder CICS
FEL.SFELLMOD	IRZ* und IIRZ*	„IRZ-Diagnosenachrichten für Enterprise Service Tools“ auf Seite 96	CICS-, IMS- oder MVS-Batch
FEL.SFELLOAD	AZU* und IAZU*	„xUnit-Unterstützung für Enterprise COBOL und PL/I“ auf Seite 94	rdz.env oder MVS-Batch

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

Ladebibliothek	Lademodule	Verwendung	STEPLIB
	BWB*	Kapitel 4, „SCLM Developer Toolkit“, auf Seite 69	rdz.env
	CRA*	Kapitel 3, „Common Access Repository Manager (CARMA)“, auf Seite 33	CRASUB* oder crastart*.conf
	ELAX*	„ELAXF* - ferne Buildprozeduren“ auf Seite 24 (Fehlerrückmeldung und Include-Vorprozessor)	ELAXF*-Prozeduren
	FEJB*	„Unterstützung bidirektionaler Sprachen für CICS“ auf Seite 95	CICS
FEL.SFELLPA	CRA*	Kapitel 3, „Common Access Repository Manager (CARMA)“, auf Seite 33	CRASRV.properties
	AQE*	Kapitel 5, „Integrated Debugger“, auf Seite 77	nicht zutreffend (LPA erforderlich)

Damit die aufgelisteten Developer for z Systems-Services funktionieren, müssen alle in Tabelle 7 auf Seite 20 dokumentierten Module, die sich auf den Service beziehen, mithilfe von STEPLIB oder LINKLIST (oder LPA) verfügbar gemacht werden. Beachten Sie, dass die Bibliothek SFELLMOD nicht von Developer for z Systems selbst verwendet wird, sondern von Code, der von Developer for z Systems generiert wurde. In der Spalte STEPLIB (falls Sie STEPLIB auswählen) in Tabelle 7 auf Seite 20 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 Programmloaden 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) DSNAME(FEL.SFELAUTH) VOLUME(volser)
- LNKLIST ADD NAME(listname) DSNAME(FEL.SFELLOAD)

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=FEL.SFELAUTH,VOL=volser
3. LNKLIST ADD NAME=LLTMP,DSN=FEL.SFELLOAD
4. LNKLIST ACTIVATE,NAME=LLTMP

Vorausgesetzte LINKLIST- und LPA-Definitionen

Zur Unterstützung optionaler Services müssen die folgenden Bibliotheken über STEPLIB oder LINKLIST/LPALIB verfügbar sein. Diese Liste enthält keine Dateigruppen, die für ein Produkt spezifisch sind, mit dem Developer for z Systems interagiert, z. B. für z/OS Explorer.

- Systemladebibliothek
 - SYS1.LINKLIB
- LE-Laufzeit (Language Environment)
 - CEE.SCEERUN
 - CEE.SCEERUN2
- DLL-Klassenbibliothek von C++'s
 - CBC.SCLBDLL
- TSO/ISPF-Client-Gateway von ISPF (für SCLMDT)
 - ISP.SISPLoad
 - ISP.SISPLPA
- REXX-Laufzeitbibliothek (für SCLMDT)
 - REXX.*.SEAGLPA
- Systemladebibliothek (für Integrated Debugger)
 - SYS1.MIGLIB
- Systemladebibliothek (für Integrated Debugger unter z/OS ab Version 1.13)
 - SYS1.SIEAMIGE
- Systemladebibliothek (für Enterprise COBOL- und PL/I-Einheitentest)
 - SYS1.CSSLIB
 - SYS1.SIXML0D1

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 vorausgesetzten und zusätzlich erforderlichen Produkte sind auch STEPLIB- oder LINKLIST/LPALIB-Definitionen erforderlich. Dies gilt beispielsweise für z/OS Explorer. 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 von rdz.env (der RSE-Konfigurationsdatei) 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 z Systems-Client verfügt über eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools'. Alle IRZM*- und IIRZ*-Module in der Ladebibliothek FEL.SFELMOD 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:

- „DBGMGR, gestartete Task für Debug Manager“
- „ELAXF* - ferne Buildprozeduren“ auf Seite 24

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.

DBGMGR, gestartete Task für Debug Manager

Passen Sie das Beispielmember FEL.#CUST.PROCLIB(DBGMGR) der gestarteten Task wie innerhalb des Members beschrieben an und kopieren Sie es in SYS1.PROCLIB. Die folgenden Informationen sind wie im nachfolgenden Codebeispiel anzugeben:

- 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 von Integrated Debugger verwendete SVC-Nummer (standardmäßig 251)
- Das übergeordnete Qualifikationsmerkmal der Ladebibliothek (standardmäßig FEL)

```

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

```

Abbildung 1. DBGMGR: Gestartete Task des Debug Managers

Anmerkung:

- Diese gestartete Task ist optional. Sie wird von der Funktion 'Integrated Debugger' von Developer for z Systems verwendet. Weitere Informationen finden Sie in Kapitel 5, „Integrated Debugger“, auf Seite 77.
- Die Beispiel-JCL hat ursprünglich den Namen FEL.SFELSAMP(AQEZJCL) und wird in „Anpassungskonfiguration“ auf Seite 15 in FEL.#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 SC43-2898).
- Falls der SVC von Integrated Debugger bereits geladen ist, wird die hier angegebene SVC-Nummer ignoriert und es wird die aktive SVC-Nummer verwendet.

ELAXF* - ferne Buildprozeduren

Developer for z Systems 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

Tabelle 8. ELAXF*-Beispielprozeduren (Forts.)

Member	Zweck
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)
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 z Systems-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 FEL.#CUST.PROCLIB(ELAXF*) wie in den Members beschrieben an und kopieren Sie sie in SYS1.PROCLIB. Geben Sie die korrekten übergeordneten Qualifikationsmerkmale für die verschiedenen Produktbibliotheken an, wie in Tabelle 9 auf Seite 26 beschrieben.

Tabelle 9. Prüfliste der übergeordneten Qualifikationsmerkmale in ELAXF*

Produkt	Standard-HLQ	Wert
Developer for z Systems	FEK	
CICS	CICSTS52.CICS	
DB2	DSNA11	
IMS	IMS	
COBOL	IGY.V5R2M0	
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 z Systems - SQL-Beispiele	FEL.#CUST.SQL	
DB2-Ausführungsbibliothek	DSNA11.RUNLIB.LOAD	

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

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

Sicherheitsdefinitionen

Zum Erstellen der Sicherheitsdefinitionen für Developer for z Systems passen Sie die Beispielmember FELRACF und AQERACF an und übergeben sie. Der Benutzer, der diesen Job übergibt, muss die Zugriffsrechte eines Sicherheitsadministrators haben, z. B. RACF SPECIAL.

FELRACF und AQERACF befinden sich in FEL.#CUST.JCL, sofern sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

Die folgende Liste von Sicherheitsdefinitionen für Developer for z Systems ist in Kapitel 9, „Sicherheitsdefinitionen“, auf Seite 103 ausführlich erläutert.

- Sicherheitseinstellungen und -klassen aktivieren
- Debug Manger als sicheren z/OS UNIX-Server definieren
- Zugriff auf Integrated Debugger definieren
- Dateiprofile definieren

Die Umgebungsconfigurationsdatei 'rdz.env'

Die RSE-Serverprozesse von z/OS Explorer (RSE-Dämon, RSE-Threadpool und RSE-Server) verwenden die Definitionen in `rdz.env`, um mehr zu Developer for z Systems-Umgebungsvariablen zu erfahren.

Die Datei `rdz.env` befindet sich in `/etc/zexpl`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEL.SFELSAMP(FELSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Die folgende Beispieldatei `rdz.env` muss an Ihre Systemumgebung angepasst werden. Für alle Variablen, die nicht explizit angegeben sind, werden Standardwerte bereitgestellt. Die Syntax der Datei entspricht den Standardsyntaxregeln der z/OS UNIX-Shell. Wenn eine US-Codepage verwendet wird, beginnen Kommentare mit dem Nummernzeichen und Leerzeichen um das Gleichzeichen (=) werden nicht unterstützt.

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

```
#RDZ_HOME=/usr/lpp/ibm/rdz
#RDZ_HLQ=FEL

## debug
# RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddebug.miner.autoreconnect=0"
# _RSE_JAVAOPTS="$_RSE_JAVAOPTS -Ddebug.miner.localhost=localhost"
## C/C++
# RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dinclude.c=/etc/zexpl/include.conf"
# _RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dinclude.cpp=/etc/zexpl/include.conf"
# _RSE_JAVAOPTS="$_RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"

## remote index search
# RSE_JAVAOPTS="$_RSE_JAVAOPTS -DRIS_BUFFER=8"
# _RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDISABLE_REMOTE_INDEX_SEARCH=true"

## system
# _RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDISABLE_DELETE_IN_SUBPROJECT=true"

## zUnit unit test
#STEPLIB=$STEPLIB:$RDZ_HLQ.SFELLOAD
#STEPLIB=$STEPLIB:SYS1.CSSLIB:SYS1.SIXMLOD1

## RTC user build
#RDZ_UBLD_DD=$CGI_ISPCONF/ISPF.conf
#RDZ_UBLD_STEPLIB=$STEPLIB

## SCLM Developer Toolkit
# SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:$RDZ_HLQ.SFELAUTH:$RDZ_HLQ.SFELLOAD
# SCLMDT_TRANTABLE=FEL.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1
```

Abbildung 2. Umgebungsconfigurationsdatei 'rdz.env'

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

RDZ_HOME

Developer for z Systems-Ausgangsverzeichnis. Das Standardverzeichnis ist `/usr/lpp/ibm/rdz`. Entfernen Sie die Kommentarsymbole und ändern Sie den Pfad so, dass er Ihrer Developer for z Systems-Installation entspricht.

RDZ_HLQ

Für die Installation von Developer for z Systems verwendetes übergeordnetes Qualifikationsmerkmal. Die Standardeinstellung ist FEK. Entfernen Sie die Kommentarzeichen und ändern Sie das Merkmal so, dass es Ihren Developer for z Systems-Dateien entspricht.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddebug.miner.autoreconnect=0"

Automatische Wiederherstellung der Verbindung zu Debug Manager. Die Standardeinstellung ist 1, die angibt, dass bei einer fehlenden oder fehlerhaften Verbindung mit dem Debug Manager-Server der Debug Miner ein Mal versucht, die Verbindung mit Debug Manager wiederherzustellen. Entfernen Sie die Kommentarzeichen und geben Sie einen anderen Wert an, um die Anzahl der Verbindungsversuche von Debug Miner mit Debug Manager zu begrenzen.

Tabelle 11. Automatische Wiederherstellung der Verbindung zu Debug Manager

debug.miner.autoreconnect	Verhalten beim Wiederherstellen der Verbindung
-1	Keine Wiederherstellung
0 (Standard)	Bis zur erfolgreichen Wiederherstellung der Verbindung jede Minute Versuch der Wiederherstellung
1-86400	Anzahl der Wiederherstellungsversuche bis zum angegebenen Wert. Der Maximalwert (86400) entspricht 24 Stunden.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Ddebug.miner.localhost=localhost"

Alternative für die Spezifikation localhost der TCP/IP-Definition. Debug Miner versucht, mithilfe der Spezifikation localhost eine Verbindung zu Debug Manager herzustellen. Dieser Versuch schlägt fehl, falls die Auflösung von localhost nicht die lokale Loopback-Adresse (127.0.0.1 für IVPv4, '::1' für IPv6) ergibt. Entfernen Sie die Kommentarzeichen und geben Sie bei Bedarf die lokale Loopback-Adresse an.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dinclude.c=/etc/zexpl/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/zexpl/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 Milli-
| sekunden. 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 Fernzu-
| griff 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**
| **-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.

| **#_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 Menü-
| punkt "Löschen" im Kontextmenü von z/OS-Unterprojekten für Benutzer
| inaktiviert werden soll, entfernen Sie das Kommentarzeichen und geben
| den Wert true an.

| **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 Bib-
| liotheken in der folgenden Liste finden Sie in „PARMLIB-Änderungen“ auf
| Seite 16:

| # SCLM Developer Toolkit
| STEPLIB=\$STEPLIB:\$RSE_HLQ.SFELAUTH:\$RSE_HLQ.SFELLOAD
| # zUnit, xUnit support for Enterprise COBOL and PL/I
| STEPLIB=\$STEPLIB:\$RSE_HLQ.SFELLOAD:SYS1.CSSLIB:SYS1.SIXMLOD1

| **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 Be-
| rechtigung auch für alle anderen STEPLIB-Bibliotheken erforderlich. Bib-
| liotheken verlieren ihre APF-Berechtigung, wenn sie mit STEPLIB-Biblio-
| theken ohne APF-Berechtigung gemischt werden.
- | • Bibliotheken, die in den Link-Pack-Bereich (LPA) gestellt werden müs-
| sen, erfordern unter Umständen zusätzliche Programmsteuerberechti-

gungen 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_UBLD_DD

Gibt die Datendefinitionsanweisungen an, die beim Generieren von JCL für IBM Rational Team Concert-Benutzerbuilds über einen Developer for z Systems-Client verwendet werden, die TSO- oder ISPF-Befehle aufrufen. Developer for z Systems verwendet standardmäßig die Definitionen in der Datei ISPF.conf, die durch die Umgebungsvariable CGI_ISPCONF in rdz.env referenziert wird. Entfernen Sie die Kommentarsymbole 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 z Systems-Client verwendet werden, die TSO- oder ISPF-Befehle aufrufen. Developer for z Systems verwendet standardmäßig die STEPLIB-Definition in der Datei rdz.env. Entfernen Sie die Kommentarsymbole und ändern Sie den Wert, um die angegebene STEPLIB-Definition zu verwenden. Diese Anweisung ist optional.

_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 wird nur im Rahmen von SCLMDT verwendet.

Anmerkung: SCLMDT fügt dem in SCLMDT_CONF_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. Der Standardname lautet FEL.#CUST.LSTRANS.FILE. Entfernen Sie das Kommentarsymbol und ändern Sie den Namen in den Namen, der im SCLM-Beispieljob ISP.SISPSAMP (FLM02LST) verwendet wird. Diese Anweisung wird nur verwendet, 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 wird nur verwendet, wenn das SCLM Developer Toolkit mit der Java EE-Build-Unterstützung verwendet wird.

Bestimmte Komponenten

Developer for z Systems setzt sich aus verschiedenen unabhängigen Funktionen mit jeweils eigenen Anpassungstasks zusammen. Konfigurieren Sie den erforderlichen Service gemäß den Anweisungen im jeweiligen Abschnitt.

Anpassungen in eigenständigen Komponenten von Developer for z Systems:

- Kapitel 3, „Common Access Repository Manager (CARMA)“, auf Seite 33
- Kapitel 4, „SCLM Developer Toolkit“, auf Seite 69
- Kapitel 5, „Integrated Debugger“, auf Seite 77
- Kapitel 6, „Hostbasierte Codeanalyse“, auf Seite 87

Anpassungen in Konfigurationsdateien von Developer for z Systems:

- „include.conf: Erzwungene Includes für C/C++-Content-Assist“ auf Seite 91

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

- „z/OS UNIX-Unterprojekte“ auf Seite 92
- „Unterstützung für Include-Vorprozessor“ auf Seite 93
- „xUnit-Unterstützung für Enterprise COBOL und PL/I“ auf Seite 94
- „Enterprise Service Tools-Unterstützung“ auf Seite 95
- „Unterstützung bidirektionaler Sprachen für CICS“ auf Seite 95
- „IRZ-Diagnosenachrichten für Enterprise Service Tools“ auf Seite 96
- „FEKRNPLI - ferner PL/I-Vorprozessor“ auf Seite 97

Installationsprüfung

Eine ausführliche Beschreibung der verschiedenen Installationsprüfprogramme (IVPs) finden Sie in Kapitel 8, „Installationsprüfung“, auf Seite 99.

Kapitel 3. Common Access Repository Manager (CARMA)

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 z Systems 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 34.
2. Erstellen Sie CARMA-VSAM-Dateien. Weitere Details enthalten die Abschnitte „CARMA-VSAM-Dateien“ auf Seite 54 und „CARMA-RAM (Repository Access Manager)“ auf Seite 54.
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 45.
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 48
 - „CRASUB* - Serverstart mit Batchübergabe“ auf Seite 52
5. Optional: Passen Sie die CA Endeavor[®] SCM-spezifischen Konfigurationsmember an. Details finden Sie in den Abschnitten „CRACFG, CRASCL, CRASHOW und

CRATMAP - Konfigurationsdateien für den CA Endevor® SCM-RAM" auf Seite 56 und „Batch-Aktionen für CA Endevor® SCM-RAM" auf Seite 59.

6. Optional: Aktualisieren Sie die Dateizuordnungs-Exec. Details finden Sie in den Abschnitten „CRANDVRA - Anfangszuordnungs-Exec für den CA Endevor® SCM-RAM" auf Seite 58, „CRAALLOC - Angepasste Zuordnungs-Exec für den RAM" auf Seite 62 und „Angepasste Zuordnungs-Exec (optional)" auf Seite 65.
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 66.

Anmerkung: Die Beispielmembers aus diesem Kapitel sind in FEL.#CUST.* und /etc/zexp1 enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration" auf Seite 15.

Serverstartmethode und aktiven RAM auswählen

Developer for z Systems unterstützt mehrere Methoden für den Start eines CARMA-Servers. Außerdem stellt Developer for z Systems 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 z Systems 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.

Produktions-RAM

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

CA Endeavor® SCM-RAM

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

RAM für CA Endeavor® SCM-Pakete

RAM für CA Endeavor® SCM-Pakete ermöglicht Clients von Developer for z Systems den direkten Zugriff auf CA Endeavor® 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 Endeavor® SCM-RAM“
- „CRASTART mit Muster-RAM“ auf Seite 38
- „Batchübergabe mit CA Endeavor® SCM-RAM“ auf Seite 40
- „Batchübergabe mit Muster-RAM“ auf Seite 42

Ausführliche Informationen zu den verschiedenen Schritten der Szenarios finden Sie in „CARMA-Konfigurationsdetails“ auf Seite 44.

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 „Unterstützung mehrerer RAM (optional)“ auf Seite 63.

CRASTART mit CA Endeavor® 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 Endeavor® 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 54.

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

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in `/etc/zexpl/CRASRV.properties` zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Sie können die Datei mit dem TSO-Befehl **OEDIT** 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 `crastart.endevor.conf` ändern. Weitere Details zu den verschiedenen Anweisungen finden Sie in „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 45.

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

Abbildung 3. CRASRV.properties: CRASTART mit CA Endevor® SCM-RAM

crastart.endevor.conf anpassen

CRASTART verwendet die Definitionen in `/etc/zexpl/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 **OEDIT** 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 48.

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 = FEL.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEL.#CUST.PARMLIB(CRASHOW)
CRACFG = FEL.#CUST.PARMLIB(CRACFG)
* Kommentarzeichen für CRABCFG znd CRABSKEL für die Verwendung von Batch-Aktionen entfernen
*CRABCFG = FEL.#CUST.PARMLIB(CRABCFG)
*CRABSKEL= FEL.#CUST.CNTL
* Kommentarzeichen entfernen und richtige DSN für die Verwendung von Package Ship angeben
*APIHJC = #shiphjc
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 = FEL.SFELLOAD,CA.NDVR.CSIQAUTH,CA.NDVR.CSIQAUTU
CRADEF = FEL.#CUST.CRADEF
CRAMSG = FEL.#CUST.CRAMSG
CRASTRS = FEL.#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
ISPLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = FEL.SFELPROC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

* CRANDVRA

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

```

Abbildung 4. *crastart.endevor.conf*: CRASTART 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 CARMA-Startprozess verfügt über einen optionalen Benutzerexit. Weitere Informationen finden Sie unter CARMA-Benutzerexit (optional).
- Der CA Endevor® SCM-RAM beinhaltet viele Konfigurationsdateien (FEL.#CUST.PARMLIB(CRA*)), die angepasst werden können. Weitere Informatio-

nen finden Sie in „CRACFG, CRASCL, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endevor® SCM-RAM“ auf Seite 56.

- Der CA Endevor® SCM-RAM verwendet eine Zuordnungs-Exec (FEL.SFELPROC-(CRANDVRA)), die angepasst werden kann. Weitere Informationen finden Sie in „CRANDVRA - Anfangszuordnungs-Exec für den CA Endevor® SCM-RAM“ auf Seite 58.
- Der CA Endevor® SCM-RAM unterstützt die Ausführung von CA Endevor® SCM-Aktionen im Batch-Modus. Für Batch-Aktionen ist eine Konfigurationsdatei (FEL.#CUST.PARMLIB(CRABCFG)) und eine Skeleton-JCL (FEL.#CUST.CNTL(CRABATCA)) erforderlich, die angepasst werden müssen. Weitere Informationen finden Sie in „Batch-Aktionen für CA Endevor® SCM-RAM“ auf Seite 59.

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 54 und „CARMA-RAM (Repository Access Manager)“ auf Seite 54.

CARMA

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

Muster-RAM

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

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in /etc/zexpl/CRASRV.properties zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Sie können die Datei mit dem TSO-Befehl **0EDIT** 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 45.

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

Abbildung 5. CRASRV.properties: CRASTART mit Muster-RAM

crastart.conf anpassen

CRASTART verwendet die Definitionen in /etc/zexpl/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 48.

```
* DD used by RAM
CRARAM1 = FEL.#CUST.CRARAM1                                * PDS RAM
* DD used by CARMA server (CRASERV)
TASKLIB = FEL.SFELLOAD
CRADEF = FEL.#CUST.CRADEF
CRAMSG = FEL.#CUST.CRAMSG
CRASTRS = FEL.#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.SFELPROC                                      * CRAALLOC
SYSTSIN = DUMMY
SYTSPRT= SYSOUT(H)

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

Abbildung 6. 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, FEL.SFELPROC-(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 54.

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

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in /etc/zexpl/CRASRV.properties zum Starten eines CARMA-Servers und für den Zugriff auf diesen 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 Standarddateipositionen verwenden, müssen Sie nur den Wert der Anweisung `clist.dsname` in FEL.#CUST.CNTL(CRASUBCA) ändern. Weitere Details zu den verschiedenen Anweisungen finden Sie in „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 45.

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

Abbildung 7. CRASRV.properties: Batchübergabe mit CA Endevor® SCM-RAM

CRASUBCA anpassen

Mit der CLIST FEL.#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 52.

```

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=FEL.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRASHOW)
//CRACFG DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRACFG)
//* Kommentarzeichen für CRABCFG und CRABSKEL für die Verwendung von Batch-Aktionen entfernen
//*CRABCFG DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRABCFG)
//*CRABSKEL DD DISP=SHR,DSN=FEL.#CUST.CNTL
//* Kommentarzeichen entfernen und richtige DSN für die Verwendung von Package Ship angeben
//*APIHJC DD DISP=SHR,DSN=#shipjhc
//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))
//C1MSG1 DD SYSOUT(H)
//C1EXMSG1 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=FEL.SFELLOAD
//      DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//      DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
//CRADEF DD DISP=SHR,DSN=FEL.#CUST.CRADEF
//CRAMSG DD DISP=SHR,DSN=FEL.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEL.#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
//ISPLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=FEL.SFELPROC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &STR(+))
&CRAPRM3 &STR(+)
&CRAPRM4 &STR(+)
&CRAPRM5 &STR(+)
&CRAPRM6 &STR(+)
&CRAPRM7 &STR(+)
&CRAPRM8 &STR(+) )
$$
EXIT CODE(0)

```

Abbildung 8. CRASUBCA: Batchübergabe 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 in „CARMA-Benutzerexit (optional)“ auf Seite 65.
- Der CA Endeavor® SCM-RAM verwendet mehrere Konfigurationsdateien (FEL.#CUST.PARMLIB(CRACFG), FEL.#CUST.PARMLIB(CRASHOW) und FEL.#CUST.PARMLIB(CRATMAP)), die angepasst werden können. Weitere Informationen finden Sie in „CRACFG, CRASCL, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endeavor® SCM-RAM“ auf Seite 56.
- Der CA Endeavor® SCM-RAM verwendet eine Zuordnungs-Exec (FEL.SFELPROC-(CRANDVRA)), die angepasst werden kann. Weitere Informationen finden Sie in „CRANDVRA - Anfangszuordnungs-Exec für den CA Endeavor® SCM-RAM“ auf Seite 58.
- Der CA Endeavor® SCM-RAM unterstützt die Ausführung von CA Endeavor® SCM-Aktionen im Batch-Modus. Für Batch-Aktionen ist eine Konfigurationsdatei (FEL.#CUST.PARMLIB(CRABCFG)), eine Skeleton-JCL FEL.#CUST.CNTL(CRABATCA) und eine optionale Standard-Jobkarte FEL.#CUST.CNTL(CRABJOB), erforderlich, die angepasst werden müssen. Weitere Informationen finden Sie in „Batch-Aktionen für CA Endeavor® SCM-RAM“ auf Seite 59.

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 54 und „CARMA-RAM (Repository Access Manager)“ auf Seite 54.

CARMA

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

Muster-RAM

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

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in /etc/zexpl/CRASRV.properties zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Sie kön-

nen die Datei mit dem TSO-Befehl **0EDIT** bearbeiten. Damit die Änderungen wirksam werden, muss die gestartete RSED-Task erneut gestartet werden.

Wenn Sie Standarddateipositionen verwenden, müssen Sie nur den Wert der Anweisung 'clist.dsname' in FEL.#CUST.CNTL(CRASUBMT) ändern. Weitere Details zu den verschiedenen Anweisungen finden Sie in „CRASRV.properties: RSE-Schnittstelle zu CARMA“ auf Seite 45.

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

Abbildung 9. CRASRV.properties: Batchübergabe mit Muster-RAM

CRASUBMT anpassen

Mit der CLIST FEL.#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 52.

```

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

```

Abbildung 10. 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, FEL.SFELPROC-(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/zexpl, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15. 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
| #
| clist.dsname=''
| crastart.configuration.file=crastart.conf
| #port.start=0
| #port.range=100
| #user.exit='FEL.SFELSAMP(CRAEXIT) '
| #startup.script.name=carma.startup.rex
| #crastart.stub=CRASTART
| #crastart.syslog=Partial
| #crastart.timeout=420
| #crastart.steplib=FEL.SFELLPA
| #crastart.tasklib=TASKLIB
```

Abbildung 11. CARMA-Konfigurationsdatei CRASRV.properties

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 34.

- *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. Ist der Dateiname in einfache Anführungszeichen (') gesetzt, handelt es sich um einen absoluten Verweis. Bei Angabe ohne einfache 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.

Der Standardwert ist eine leere Zeichenfolge, die angibt, dass CARMA nicht konfiguriert ist.

crastart.configuration.file

Gibt den Namen der CRA-START-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/zexpl`), um die Datei zu finden.
- Relativer Pfad: Dies ist das Verzeichnis und der Dateiname ohne einen führenden Schrägstrich (/). CARMA fügt Ihr Konfigurationsverzeichnis (standardmäßig `/etc/zexpl/`) 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.

#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. Zur Angabe des Anfangs des Portbereichs entfernen Sie das Kommentarzeichen und passen die Option an. 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 SC43-2898).

#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 FEL.SFELSAMP(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 z Systems-Dateien enthalten sind (standardmäßig `/usr/lpp/ibm/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/zexpl/`) 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 z Systems-Dateien enthalten sind (standardmäßig `/usr/lpp/ibm/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/zexpl/`) 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 `rdz.env` erfolgt. Die Standardeinstellung ist `FEL.SFELLPA`. 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. Rational Developer for z Systems führt in dieser Umgebung den CARMA-Server CRASERV aus.

Die Datei `crastart*.conf` befindet sich in `/etc/zexpl/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEL.SFELSAMP(FELSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15. 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 z Systems 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 Zuordnungsstrukture.

- 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:

&CRAUSER.	Client-Benutzer-ID
&CRADATE.	Aktuelles Datum im (siebenstelligen julianischen) Format Djjjjttt
&CRATIME.	Aktuelle Uhrzeit im Format Thhmmss (Stunden, Minuten, Sekunden)
&CRAPRM1.	Portnummer
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 65.

Abb. 12 zeigt ein grundlegendes crastart*.conf-Skeleton, das ISPF-Services einschließt.

```
* DD used by RAM

* DD used by CARMA server (CRASERV)
TASKLIB = FEL.SFELLOAD
CRADEF  = FEL.#CUST.CRADEF
CRAMSG  = FEL.#CUST.CRAMSG
CRASTRS = FEL.#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.SFELPROC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

* CRAALLOC

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

Abbildung 12. 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 45 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 im Handbuch *IBM Explorer for z/OS Host Configuration Guide (IBM Form SC27-8437)*.

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 65.

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 im Handbuch *IBM Explorer for z/OS Host Configuration Guide (IBM Form SC27-8437)*.

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. Rational Developer for z Systems führt in dieser Umgebung den CARMA-Server CRASERV aus.

CRASUB* befindet sich in FEL.#CUST.CNTL, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

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

Developer for z Systems stellt mehrere JCL-Jobs (CRASUB*) 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. 13 auf Seite 53 zeigt ein grundlegendes "CRASUB*" -Skeleton, das ISPF-Services einschließt.

```

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

```

Abbildung 13. 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 im Handbuch *IBM Explorer for z/OS Host Configuration Guide (IBM Form SC27-8437)*.

CARMA-VSAM-Dateien

Der CARMA-Server erfordert Lesezugriff auf drei VSAM-Dateien. Die Beispielmuster zum Erstellen und Füllen dieser VSAM-Dateien befinden sich in `FEL.#CUST.JCL`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEL.SFELSAMP(FELSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

Anmerkung:

- Wenn Sie die Definitionen für einen (benutzerdefinierten) RAM mit einer vorhandenen VSAM-Konfiguration zusammenführen müssen, finden Sie entsprechende Informationen im Beispielmuster `FEL.#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 Beispielmuster `FEL.#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 Beispielmuster 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.

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 z Systems 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 z Systems bereitgestellt werden, befinden sich in FEL.SFELLOAD und der Beispiel Quellcode und die Kompilierjobs befinden sich in FEL.SFELSAMP, sofern Sie bei der SMP/E-Installation von Developer for z Systems kein anderes übergeordnetes Qualifikationsmerkmal verwendet haben.

Die folgenden Abschnitte enthalten Anpassungshinweise für die RAM, die in Developer for z Systems verfügbar sind. Die Beispielmuster, auf die verwiesen wird, befinden sich in FEL.#CUST.*, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

Eingehende Informationen zu CARMA sowie weitere Informationen zu den bereitgestellten Muster-RAM und zum bereitgestellten Musterquellcode 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 z Systems den direkten Zugriff auf CA Endeavor® SCM-Elemente.
- Der Name des Lademoduls lautet CRARNVDR.
- 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.
- Der CA Endeavor® SCM-RAM beinhaltet viele Konfigurationsdateien, die angepasst werden können. Weitere Informationen finden Sie in „CRACFG, CRASCL, CRASHOW und CRATMAP - Konfigurationsdateien für den CA Endeavor® SCM-RAM“ auf Seite 56.
- Der CA Endeavor® SCM-RAM verwendet eine Zuordnungs-Exec (FEL.SFELPROC-(CRANDVRA)), die angepasst werden kann. Weitere Informationen hierzu enthält der Abschnitt „CRANDVRA - Anfangszuordnungs-Exec für den CA Endeavor® SCM-RAM“ auf Seite 58.
- Der CA Endeavor® SCM-RAM unterstützt die Ausführung von CA Endeavor® SCM-Aktionen im Batch-Modus im Hintergrund. Weitere Informationen finden Sie in „Batch-Aktionen für CA Endeavor® SCM-RAM“ auf Seite 59.

RAM für CA Endeavor® SCM-Pakete

- Der RAM für CA Endeavor® SCM-Pakete ist ein Produktions-RAM.
- RAM für CA Endeavor® SCM-Pakete ermöglicht Clients von Developer for z Systems 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 FEL.#CUST.JCL(CRA#VPDS) zugeordnet und vorbereitet werden. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.
- Quellcode und Kompilierjobs sind in FEL.SFELSAMP verfügbar. Weitere Informationen finden Sie im Handbuch *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 FEL.SFELSAMP 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 FEL.#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 FEL.#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 FEL.#CUST.JCL(CRA#UADD), um FEL.SFELVSM2(CRA0SLMD) in CRADEF und FEL.SFELVSM2(CRA0SLMS) in CRASTRS zusammenzuführen. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation.
- Quellcode und Kompilierjobs sind in FEL.SFELSAMP verfügbar. Weitere Informationen finden Sie im Handbuch *Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).

CRACFG, CRASCL, 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 FEL.#CUST.PARMLIB, sofern

Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie im Abschnitt „Anpassungskonfiguration“ auf Seite 15.

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

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

```
# ENTRY-STAGE-COPY-MODE = RETRIEVE-ADD
# ALTERNATIVE-ALLOC = YES
# PACKAGE-EDITING-OPTION = {READONLY | DISABLED}
# SCL-REQUIRED = YES
# SCL-DATASET-TEMPLATE = FEL.#CUST.PARMLIB(CRASCL)
# DYNAMIC-VB-DATASET-ALLOC = YES
# DATASET-ALLOC-OVERRIDE = SPACE(5,30) TRACK UNIT(SYSALLDA)
```

Abbildung 14. CRACFG - Interaktion von CA Endeavor® SCM-RAM mit dem SCM

CRASCL - Schablonen-SCL für den CA Endeavor® SCM-RAM

CRASCL ist eine Schablonen-SCL (Software Control Language), die vorgeben kann, welche Aktionen und Optionen beim Verarbeiten von Packages von CA Endeavor® SCM zulässig sind.

Wenn Sie verwendet wird, können nur Aktionen und Optionen, die explizit in der Schablonen-SCL aufgelistet sind, im Editor für Developer for z Systems-Packages angegeben werden. Wenn Sie die Standardeinstellungen ändern möchten, finden Sie Anpassungsanweisungen in der im Member enthaltenen Dokumentation.

CRASHOW - CA Endeavor® SCM-RAM-Standardfilter

CRASHOW definiert Standardfilter für CA Endeavor® 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 15. CRASHOW - CA Endeavor® SCM-RAM-Standardfilter

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

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

CRATMAP überschreibt den CA Endeavor® 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 16. CRATMAP: CA Endevor® SCM-RAM-Standardfilter

CRANDVRA und CRADYNDA - Zuordnungs-Execs 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.

Sie können eine Kopie dieser Zuordnungs-REXX-Execs anpassen, falls bestimmte Standardwerte, wie der Dateiname, nicht den Standards Ihres Standorts entsprechen. Die Execs befinden sich in FEL.SFELPROC, sofern Sie bei der SMP/E-Installation von Developer for z Systems 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 65.

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 SFELPROC in der DD-Anweisung 'SYSEXEC' Ihrer gewählten CARMA-Startmethode aktualisieren, damit sie mit Ihrem neuen Dateinamen übereinstimmt.

CRANDVRA - Anfangszuordnungs-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
BROWSEV	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.BROWSEV	Temporär
ENHCEDIT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDIT	Temporär
ENHCEDITV	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDITV	Temporär
C1PRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	Temporär

DD-Anweisung	Dateiname	Typ
SPCLLIST	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.SPCLLIST	Temporär
PKGSCLS	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.PKGSCLS	Temporär
CRABJCLO	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRABJCLO	Temporär
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRAPARM	Temporär

CRADYNDA - Neuzordnungs-Exec für den CA Endeavor® SCM-RAM

CA Endeavor® SCM kann mit variablen geblockten Dateien mit unterschiedlichen Satzlängen arbeiten und setzt voraus, dass Developer for z Systems für bestimmte Aktionen identische Zuordnungen verwendet. Die Startmethode mit Batchübergabe und die Startmethode CRASTART rufen beide die REXX-Exec CRADYNDA auf, um eine benutzerdefinierte Arbeitsdatei den richtigen Attributen zuzuordnen.

Batch-Aktionen für CA Endeavor® SCM-RAM

Normalerweise werden CA Endeavor® SCM-Aktionen wie "Generate Element" (Element generieren) online im Serveradressraum von CARMA ausgeführt. Dieses Verhalten führt zu Problemen, wenn Ihre CA Endeavor® 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 Endeavor® SCM-RAM Batch-Aktionen. Bei aktivierten Batch-Aktionen übergibt der CA Endeavor® 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 Endeavor® SCM-Prozeduren zugeordnet werden, da TSO für die übergebene JCL nicht aktiv sein muss.

CA Endeavor® SCM-RAM-Batch-Aktionen entsprechen den CA Endeavor® SCM-Hintergrundaktionen von Developer for z Systems.

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.endeavor.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. Developer for z Systems-Clients können dieses Verhalten zu überschreiben.

Beispiel:

```
/* Kommentarzeichen für CRABCFG und CRABSKEL für die Verwendung von Batch-Aktionen entfernen
/*CRABCFG DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRABCFG)
/*CRABSKEL DD DISP=SHR,DSN=FEL.#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 FEL.#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 Developer for z Systems, befindet, damit der CA Endeavor® SCM-RAM den Status prüfen kann.
- Der Developer for z Systems-Client kann 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 FEL.#CUST.PARMLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

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
DELETE-ELEMENT  = CRABATCA
GENERATE-ELEMENT = CRABATCA
MOVE-ELEMENT    = CRABATCA
RETRIEVE-ELEMENT = CRABATCA
PRINT-ELEMENT   = CRABATCA
PRINT-MEMBER    = CRABATCA
SIGNIN-ELEMENT  = CRABATCA
TRANSFER-ELEMENT = CRABATCA
#
# Befehlssubstitutionsschlüssel in Batch-Aktions-JCL
BSTIPT01-KEY = <CRA_BSTIPT01>
```

Abbildung 17. CRABCFG: Konfiguration der Batch-Aktion für den CA Endeavor® 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.endevor.conf definiert werden.

DEFAULT-JOBCARD

Name des Members, das eine Standardjobkarte enthält. Falls nicht durch eine benutzerspezifische Jobkarte, die auf dem Developer for z Systems-Client 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 Endevor® SCM-Aktionen

Die Schlüsselnamen stellen die CA Endevor® 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 Endevor® 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 Endevor® SCM angewiesen wird, die angeforderte Aktion für das angeforderte Element auszuführen.

CRABATCA - JCL für Batch-Aktionen des CA Endevor® 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 FEL.#CUST.CNTL, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

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

```
//<JOBCARD>
//*
//CRABATCA EXEC PGM=NDVRC1,DYNAMNBR=1500,REGION=4096K,PARM='C1BM3000'
//STEPLIB DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
// DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD
//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 18. CRABATCA: JCL für die Batch-Aktion des CA Endevor® SCM-RAM

CRABJOBC - Jobkarte für Batch-Aktionen des CA Endevor® SCM-RAM

CRABJOBC ist eine standardmäßige Beispieljobkarte, die für Skeleton-JCL mit Batch-Aktionen verwendet wird, die den Schlüssel <JOBCARD> angibt. Wenn Sie die Standardwerte ändern wollen, finden Sie entsprechende Anpassungsanweisungen in der im Member enthaltenen Dokumentation.

CRABJOBC befindet sich in FEL.#CUST.CNTL, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

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

```
I //<USERID>B JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
I //*PROCS JCLLIB ORDER=(COBOL.V5R1M0.SIGYPROC,CBC.SCCNPRC)
```

Abbildung 19. CRABJOBC: Jobkarte für Batch-Aktionen des CA Endevor® 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 FEL.SFELPROC, sofern Sie bei der SMP/E-Installation von Rational Developer for z Systems 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 SFELPROC 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.

Unterstützung mehrerer RAM (optional)

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 Adressraum des CARMA-Servers verfügbar sind. Diese Anforderung kann Änderungen an den CARMA-Konfigurationsbeispielen wie CRASUBMT oder crastart.conf erfordern, die in Developer for z Systems 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:

- FEL.SFELVSM2(CRA0VCAD): CRADEF-Definitionen
- FEL.SFELVSM2(CRA0VCAS): CRASTRS-Definitionen
- /etc/zexpl/crastart.endevor.conf: CRASTART-Konfigurationsdatei

Definitionen für den PDS-RAM:

- FEL.SFELVSM2(CRA0VDEF): CRADEF-Definitionen
- FEL.SFELVSM2(CRA0VSTR): CRASTRS-Definitionen
- FEL.#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 SFELVSM2-Mitgliedern. Diese Mitglieder enthalten Definitionen für alle Muster-RAM, nicht nur für den PDS-RAM.
2. Führen Sie diese Daten mit den SFELVSM2-Mitgliedern des Endeavor® SCM-RAMs zusammen.
3. Erstellen Sie eine Liste mit den Voraussetzungen für den PDS-RAM:
 - DD-Anweisung 'CRARAM1', verweist auf FEL.#CUST.CRARAM1
 - 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.endeavor.conf eine CRARAM1-Definition hinzu:
CRARAM1 = FEL.#CUST.CRARAM1
3. Überprüfen Sie die PROGRAM-Anweisung in crastart.endeavor.conf, um sicherzustellen, dass sie die Umgebung bereitstellen kann, die von beiden RAM benötigt wird.
4. 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 Endeavor® SCM-RAM-Vorzuordnungs-Exec verwendet.
 - %CRANDVRA: Vorzuordnungs-Exec des CA Endeavor® SCM-RAMs, die sich in FEL.SFELPROC 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.

Rational Developer for z Systems verwendet in den einzelnen vorkonfigurierten Konfigurationen, die zuvor in diesem Kapitel beschrieben wurden, eine Zuordnungs-Exec. FEL.SFELPROC(CRANDVRA), die Zuordnungs-Exec für CA Endevor® SCM-RAM, und FEL.SFELPROC(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 in 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)“ 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 Initialisierungsaufwurf findet nach der Zuordnung der temporären Dateien und vor dem Aufrufen des CARMA-Servers statt. Der Bereinigungsaufwurf 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 FEL.SFELSAMP(CRAEXIT) zur Verfügung gestellt, sofern Sie bei der SMP/E-Installation von Developer for z Systems 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 Exitaufwurf 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 FEL.#CUST.ASM(CRAXJCL) und FEL.#CUST.JCL(CRA#CIRX) verfügbar, sofern Sie beim Anpassen und Übergeben des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

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

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 70.
2. Passen Sie die Konfigurationsdateien für Developer for z Systems an. Weitere Details enthalten die folgenden Abschnitte:
 - „Aktualisierung von ISPF.conf für SCLMDT“ auf Seite 70
 - „Aktualisierung von 'rdz.env' für SCLMDT“ auf Seite 71
3. Definieren Sie die Unterstützung für die Umsetzung langer/kurzer Namen (optional). Weitere Details enthält der Abschnitt „Umsetzung langer/kurzer Namen (optional)“ auf Seite 72.
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 74.
5. Aktualisieren Sie SCLM, um spezifische Komponenten für SCLMDT zu definieren. Weitere Details enthält der Abschnitt „SCLM-Updates für SCLMDT“ auf Seite 75.
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 76.

Voraussetzungen

Eine vollständige Liste der Softwarevoraussetzungen für Developer for z Systems einschließlich vorausgesetzter und zusätzlich erforderlicher Produkte finden Sie auf der Registerkarte **System Requirements** auf der Produktseite von Developer for z Systems (<http://www-03.ibm.com/software/products/en/developerforsystemz/>). Dort sind auf die Ant-Spezifikationen aufgeführt, die für Java EE-Builds im SCLM Developer Toolkit benötigt werden.

Achtung: Das SCLM Developer Toolkit arbeitet mit dem Legacy ISPF Gateway, sodass z/OS ab Version 1.8 erforderlich ist. Falls Interactive ISPF Gateway für den TSO-Befehlsservice verwendet wird, müssen sowohl das Legacy als auch das Interactive ISPF Gateway konfiguriert sein.

Das SCLM Developer Toolkit erfordert zusätzliche Anpassungsschritte für System-einstellungen. Lesen Sie hierzu die Beschreibung im Abschnitt „PARMLIB-Änderungen“ auf Seite 16. 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.SISPLOAD, 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. .
- 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/zexp1/WORKAREA/ haben. Das Verzeichnis WORKAREA/ befindet sich in /var/zexp1/, sofern Sie bei der Anpassung von IBM Explorer for z/OS keine andere Position angegeben haben.

Aktualisierung von ISPF.conf für SCLMDT

SCLM Developer Toolkit verwendet die ISPF/SCLM-Standard-Skeletons, um sicherzustellen, dass die Skeleton-Bibliothek ISP.SISPSLIB der ISPSLIB-Verkettung in ISPF.conf zugeordnet wird. Die Verwendung der Datei ISP.SISPSENU ist optional.

Die Datei ISPF.conf befindet sich in /etc/zexp1/, sofern Sie bei der Anpassung von IBM Explorer for z/OS keine andere Position angegeben haben. 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. Kommentarseiten beginnen mit einem Stern (*). Fügen Sie Dateien zur Verkettung in derselben Zeile hinzu und trennen Sie die einzelnen Namen jeweils durch ein Komma (,).

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

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

Abbildung 20. 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" im Handbuch *IBM Explorer for z/OS Host Configuration Reference (IBM Form SC27-8438)*.
- 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 '`rdz.env`' für `SCLMDT`

`SCLM Developer Toolkit` verwendet einige Anweisungen in `rdz.env`, um Dateien und Verzeichnisse zu finden.

Die Datei `rdz.env` befindet sich in `/etc/zexpl`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEL.SFELSAMP(FELSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **EDIT** bearbeiten.

Anmerkung: Starten Sie die gestartete `RSED`-Task von *IBM Explorer for z/OS* erneut, damit die Änderungen wirksam werden.

Das folgende Codebeispiel zeigt die `SCLMDT`-Anweisungen in der Datei `rdz.env`, die Sie an Ihre Systemumgebung anpassen müssen. Weitere Informationen zum Anpassen der Datei `rdz.env` finden Sie in „Die Umgebungsconfigurationsdatei '`rdz.env`'“ auf Seite 27.

```
_SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:FEL.SFELAUTH:FEL.SFELLOAD
# SCLMDT_TRANTABLE=FEL.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/apache/Ant/apache-ant-1.7.1
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CGI_ISPHOME
CGI_DTWÖRK=$_SCLMDT_WORK_HOME
```

Abbildung 21. Aktualisierung von '`rdz.env`' 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 FEL.#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. Ändern Sie alle Verweise auf FEL.#CUST.LSTRANS.FILE durch
/* Ihre Namenskonvention für die SCLM-Umsetzungs-VSAM.
/*
//CREATE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE FEL.#CUST.LSTRANS.FILE
SET MAXCC=0
DEFINE CLUSTER(NAME(FEL.#CUST.LSTRANS.FILE) -
              VOLUMES(*****)) -
              RECORDSIZE(58 2048) -
              SHAREOPTIONS(3 3) -
              CYLINDERS(1 1) -
              KEYS(8 0) -
              INDEXED) -
DATA (NAME(FEL.#CUST.LSTRANS.FILE.DATA)) -
INDEX (NAME(FEL.#CUST.LSTRANS.FILE.INDEX))

/* DEFINE ALTERNATE INDEX WITH NONUNIQUE KEYS -> ESDS */

DEFINE ALTERNATEINDEX(-
              NAME(FEL.#CUST.LSTRANS.FILE.AIX) -
              RELATE(FEL.#CUST.LSTRANS.FILE) -
              RECORDSIZE(58 2048) -
              VOLUMES(*****)) -
              CYLINDERS(1 1) -
              KEYS(50 8) -
              UPGRADE -
              NONUNIQUEKEY) -
DATA (NAME(FEL.#CUST.LSTRANS.FILE.AIX.DATA)) -
INDEX (NAME(FEL.#CUST.LSTRANS.FILE.AIX.INDEX))

/*
/*
//PRIME EXEC PGM=IDCAMS,COND=(0,LT)
//SYSPRINT DD SYSOUT=*
//INITREC DD *
INITREC1
/*
//SYSIN DD *
REPRO INFILE(INITREC) -
      OUTDATASET(FEL.#CUST.LSTRANS.FILE)
IF LASTCC = 4 THEN SET MAXCC=0

BLDINDEX IDS(FEL.#CUST.LSTRANS.FILE) -
          ODS(FEL.#CUST.LSTRANS.FILE.AIX)

IF LASTCC = 0 THEN -
  DEFINE PATH (NAME(FEL.#CUST.LSTRANS.FILE.PATH) -
              PATHENTRY (FEL.#CUST.LSTRANS.FILE.AIX))
/*

```

Abbildung 22. FLM02LST: Konfigurations-JCL für Umsetzung langer/kurzer Namen

Anmerkung: Benutzer benötigen für diese VSAM-Dateigruppe die Zugriffsberechtigung UPDATE.

Aktualisierung von 'rdz.env' 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 `rdz.env`, damit der Name der VSAM für die Umsetzung langer Namen in Kurznamen übereinstimmt.

Die Datei `rdz.env` befindet sich in `/etc/zexpl`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEL.SFELSAMP(FELSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15. 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 z Systems 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 Beispielskript `/usr/lpp/ibm/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.
- Setzen Sie vor der Verwendung von Ant die Umgebungsvariablen `JAVA_HOME` und `ANT_HOME`.

- JAVA_HOME (in 'rse.env') muss auf das Java-Ausgangsverzeichnis zeigen.
Beispiel:
JAVA_HOME=/usr/lpp/java/IBM/J6.0
- ANT_HOME (in 'rse.env') muss auf das Ant-Ausgangsverzeichnis zeigen. 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/ibm/rdz/samples/BWBTRANT
- cat ./apache-ant-1.7.1/README
- chmod -R 755 *
- oedit /etc/rdz.env

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 z Systems 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 z Systems 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 z Systems	<ul style="list-style-type: none"> • FEL.SFELSAMV • SMP/E-Installation 	

Tabelle 13. Prüfliste für den SCLM-Administrator (Forts.)

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Beispielverzeichnis von Developer for z Systems	<ul style="list-style-type: none"> • /usr/lpp/ibm/rdz/samples • SMP/E-Installation 	
Java-Verzeichnis bin	<ul style="list-style-type: none"> • /usr/lpp/java/J6.0/bin • rse.env - \$JAVA_HOME/bin 	
Ant-Verzeichnis bin	<ul style="list-style-type: none"> • /usr/lpp/Apache/Ant/apache-ant-1.7.1/bin • rdz.env - \$ANT_HOME/bin 	
WORKAREA-Ausgangsverzeichnis	<ul style="list-style-type: none"> • /var/zexpl • rse.env - \$CGI_ISPWORK 	
Ausgangsverzeichnis für SCLMDT-Projektkonfigurationen	<ul style="list-style-type: none"> • /var/rdz/sclmdt • rdz.env - \$_SCLMDT_CONF_HOME 	
VSAM für Umsetzung langer/kurzer Namen	<ul style="list-style-type: none"> • FEL.#CUST.LSTRANS.FILE • rdz.env - \$_SCLMDT_TRANTABLE 	

Alte Dateien aus WORKAREA und /tmp entfernen

SCLM Developer Toolkit nutzt die Verzeichnisse 'WORKAREA' und /tmp gemeinsam mit dem Legacy ISPF Gateway. Für diese Verzeichnisse könnte eine regelmäßige Bereinigung erforderlich sein. z/OS UNIX stellt das Shell-Skript '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-Cron-Dämon, der Befehle zu angegebenen Zeitpunkten (Datum/Uhrzeit) ausführt, können Sie ein automatisiertes Tool einrichten, das bestimmte Zielverzeichnisse in regelmäßigen Abständen bereinigt. Weitere Informationen zum Skript 'skulker' und dem Cron-Dämon finden Sie in der Veröffentlichung 'UNIX System Services Command Reference (IBM Form FSA22-7802)'.

Kapitel 5. Integrated Debugger

Mit der Hostkomponente Developer for z Systems Integrated Debugger können Clients ab Version 9.0.1 bei verschiedenen LE-basierten Anwendungen (LE - Language Environment) Fehler beheben. Für Integrated Debugger ist z/OS ab Version 1.10 erforderlich.

Einen Überblick über den Datenfluss von Integrated Debugger finden Sie im Abschnitt "Integrated Debugger" des Kapitels "Wissenswertes zu Developer for z Systems" in der *Hostkonfigurationsreferenz* (IBM Form SC27-8578).

Voraussetzungen und Prüfliste

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:

- SVC hinzufügen (IPL optional)
- Ggf. Link-Pack-Bereich für SVC aktualisieren
- LINKLIST aktualisieren
- APF-Autorisierung
- Gestartete Task definieren
- Sicherheitsprofile und Zugriffslisten definieren
- TCP/IP-Ports für Client-Host und hostinterne Kommunikation reservieren
- (Optional) JCL für CICS-Regionen aktualisieren
- (Optional) CICS-Systemdefinition aktualisieren
- (Optional) JCL für IMS-Nachrichtenregion aktualisieren
- (Optional) JCL für den Adressraum der gespeicherten DB2-Prozedur aktualisieren
- (Optional) ELAXF*-Prozeduren für ferne Builderstellung aktualisieren

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 Task DBGMGR aktiv sein (zusammen mit der obligatorisch gestarteten Task RSED von IBM Explorer for z/OS). Informationen zur DBGMGR-Start-JCL finden Sie im Abschnitt „DBGMGR, gestartete Task für Debug Manager“ auf Seite 23.
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 80.
3. Für die gestartete DBGMGR-Task muss die Bibliothek FEL.SFELAUTH APF-autorisiert sein. Lesen Sie hierzu „Parmlib-Aktualisierungen für Integrated Debugger“ auf Seite 80.
4. Integrated Debugger greift auf einen Developer for z Systems-Supervisoraufruf (SVC) zurück. Das SVC-Lademodul muss im Link-Pack-Bereich (LPA) geladen werden und kann dynamisch geladen werden. Details hierzu finden Sie im Abschnitt „Parmlib-Aktualisierungen für Integrated Debugger“ auf Seite 80.

5. 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 80.
6. 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 82.
7. Für Integrated Debugger sind verschiedene Sicherheitsberechtigungen erforderlich. Details hierzu finden Sie im Abschnitt „Sicherheitsupdates für Integrated Debugger“ auf Seite 82.
8. Für die gestartete DBGMR-Task sind einige Sicherheitsberechtigungen erforderlich. Details hierzu finden Sie im Abschnitt „Sicherheitsupdates für Integrated Debugger“ auf Seite 82.

Der folgende Schritt ist 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 und IP CICS Sockets für CICS definiert sind. Details hierzu finden Sie im Abschnitt „CICS-Aktualisierungen für Integrated Debugger“ auf Seite 83.

Mit Integrated Debugger kann ein Language Environment-Benutzerexit (LE) verwendet werden, der TEST-Laufzeitoptionen aus einer Datei für ein gegebenes Modul liest. Dies ist nützlich für das Debugging von Code in Subsystemen, z. B. IMS-Transaktionen und gespeicherten DB2-Prozeduren, da es nicht möglich ist, in diesem Code dynamisch TEST-Laufzeitoptionen bereitzustellen. Die folgenden Schritte sind nur erforderlich, wenn der LE-Benutzerexit verwendet wird.

1. Erstellen Sie den LE-Benutzerexit und eine angepasste LE-Umgebung, die den Benutzerexit aufruft. Details hierzu finden Sie im Abschnitt „Language Environment-Aktualisierungen für Integrated Debugger“ auf Seite 82.
2. Definieren Sie die angepasste LE-Umgebung für Ihre IMS-Transaktion. Details hierzu finden Sie im Abschnitt „IMS-Aktualisierungen für Integrated Debugger“ auf Seite 84.
3. Definieren Sie die angepasste LE-Umgebung für Ihre gespeicherte DB2-Prozedur. Details hierzu finden Sie im Abschnitt „Gespeicherte DB2-Prozeduraktualisierungen für Integrated Debugger“ auf Seite 84.

Anmerkung:

- Integrated Debugger verwendet CICS CADP, um TEST-Laufzeitoptionen für CICS-Transaktionen bereitzustellen. Weitere Informationen zu CADP finden Sie in der CICS TS-Dokumentation. Sie können bei Bedarf jedoch auch den LE-Benutzerexitmechanismus verwenden.
- Der LE-Benutzerexitmechanismus kann auch für reguläre Anwendungen verwendet werden. Weitere Details enthält der Abschnitt „ELAXF*-Aktualisierungen für Integrated Debugger“ auf Seite 85.

Integrated Debugger und COBOL V4

Für das Debugging von Programmen, die in COBOL V3.4 und V4 geschrieben sind, muss Integrated Debugger über Zugriff auf eine Listendatei (PDS oder PDS/E) verfügen. Der Dateiname kann über die Umgebungsvariable AQE_DBG_V4LIST oder die DD-Anweisung AQEV4LST angegeben werden. Ist keine dieser Angaben vorhanden, erstellt Integrated Debugger den Dateinamen durch das Ersetzen des letzten Qualifikationsmerkmals für die Datei der ausführbaren Funktion (z. B.

| .LOAD) durch die Angabe .LISTING. Erkundigen Sie sich bei den verantwortlichen
| Entwicklern, welche Methode an Ihrem Standort verwendet wird.

| **Integrated Debugger und andere Language Environment-basierte Debugger**

| Normalerweise kann nur ein LE-basierter Debugger (LE - Language Environment),
| wie beispielsweise 'Integrated Debugger', in einer bestimmten Anwendung, CICS-
| Region, gespeicherten DB2-Prozedur oder IMS-Transaktion aktiv sein. Ob es sich
| bei einem Debugger um einen LE-basierten Debugger handelt, können Sie gut dar-
| an erkennen, dass er ein CEEVDBG-Lademodul oder ein Alias bereitstellt, das für
| die Anwendung verfügbar sein muss.

| Integrated Debugger kann jedoch in Verbindung mit IBM Debug Tool for z/OS ko-
| existieren, wenn Integrated Debugger von der Anwendung zuerst geladen wird.

| **Integrated Debugger und verschlüsselte Kommunikation**

| Verwendet der Developer for z Systems-Client eine Verschlüsselung für die Kom-
| munikation mit dem RSE-Dämon, verwendet die (clientbasierte) Debug-Engine
| standardmäßig ebenfalls eine Verschlüsselung für die Kommunikation mit dem
| (hostbasierten) Debug Manager. Die Debug-Engine nutzt standardmäßig dieselben
| Zertifikate wie der Developer for z Systems-Client.

| Dabei wird davon ausgegangen, dass die Konfiguration für die Verschlüsselung
| der Kommunikation bei den gestarteten Tasks für den RSE-Dämon und den Debug
| Manager übereinstimmt. In Situationen, in denen die Verschlüsselungseinstellun-
| gen für den RSE-Dämon und den Debug Manager voneinander abweichen, gilt
| Folgendes:

- Die Debug-Engine kann nach einer entsprechenden Bestätigung versuchen, einen nicht verschlüsselten Datenaustausch vorzunehmen, wenn die Einrichtung einer verschlüsselten Sitzung fehlschlägt.
- Die Debug-Engine kann andere Authentifizierungszertifikate verwenden als der Developer for z Systems-Client.

| Anders als der RSE-Dämon verfügt der Debug Manager nicht über eine native Un-
| terstützung für verschlüsselte Kommunikation. Der Debug Manager greift bei der
| verschlüsselten Kommunikation auf einen TCP/IP-Service mit der Bezeichnung
| AT-TLS (Application Transparent Transport Layer Security) zurück. Schrittweise
| Anleitungen zum Einrichten von AT-TLS finden Sie im Abschnitt "AT-TLS konfigu-
| rieren" im *IBM Rational Developer for z Systems Referenzhandbuch (IBM Form (SC27-
| 8578))*.

| **Integrated Debugger und SYSPLEX**

| Developer for z Systems ist nicht SYSPLEX-fähig. Daher muss das Programm, bei
| dem ein Debugging ausgeführt werden soll, auf dem System aktiv sein, mit dem
| der Developer for z Systems-Client verbunden ist. Dies beinhaltet das Starten von
| Debugsitzungen über einen Batch-Job.

| In z/OS 2.1 können Sie auf der Jobkarte SYSAFF=* oder SYSTEM=* angeben um zu
| erzwingen, dass der Job auf dem System, an das er übergeben wurde, auch ausge-
| führt wird. Auf älteren Systemen müssen Sie explizit den richtigen Systemnamen
| angeben, wenn Sie ein JESPLEX zum Verbinden mehrerer JES-Subsysteme in einem
| SYSPLEX verwenden.

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 Debug Manager“ auf Seite 23.

- 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 von Integrated Debugger verwendete SVC-Nummer (standardmäßig 251)
- Das übergeordnete Qualifikationsmerkmal der Ladebibliothek (standardmäßig FEL)

Parmlib-Aktualisierungen für Integrated Debugger

- Die gestartete DBGMR-Task muss aktiv sein, bevor Developer for z Systems-Clients die Verbindung zum Host herstellen. Weitere Informationen zum automatischen Starten des Servers während des einleitenden Programmladens (IPL) finden Sie im Abschnitt „Gestartete Tasks zu COMMNDxx hinzufügen“ auf Seite 17.
- Für die gestartete DBGMR-Task muss die Bibliothek FEL.SFELAUTH APF-autorisiert sein. Details hierzu finden Sie im Abschnitt „APF-Berechtigungen in PROGxx“ auf Seite 19.
- Integrated Debugger muss in LE (Language Environment) aufgerufen werden können. Die Bibliothek FEL.SFELAUTH 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 20.

Anmerkung:

- Stellen Sie bei Verwendung von LINKLIST sicher, dass FEL.SFELAUTH vor den Bibliotheken anderer LE-basierter Debugger mit dem Lademodul CEEVDBG aufgeführt wird. Dies ist z. B. bei IBM Debug Tool for z/OS zu beachten, das hlq.SEQA*-Bibliotheken verwendet.
- Zur Vermeidung von Konflikten sollte nur ein LE-basierter Debugger in LINKLIST definiert sein.
- Integrated Debugger kann in Verbindung mit IBM Debug Tool for z/OS eingesetzt werden, wenn Integrated Debugger von der Anwendung zuerst geladen wird.
- Von der gestarteten Task für Debug Manager abgesehen ist für die Lademodule für Integrated Debugger in FEL.SFELAUTH keine autorisierte Ausführung erforderlich. Da sich die Lademodule hier befinden, können sie in einer autorisierten Umgebung verwendet werden.
- Integrated Debugger verwendet den z/OS-Binder. Dies impliziert, dass SYS1.MIGLIB in LINKLIST (oder STEPLIB) enthalten sein sollte. Details hierzu finden Sie im Abschnitt „Vorausgesetzte LINKLIST- und LPA-Definitionen“ auf Seite 22.
- 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 22.

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 //IEWBNDD was not found

- Für Integrated Debugger muss ein Supervisoraufruf (SVC) für Developer for z Systems für Ihr System definiert sein. Die SVC-Standardnummer lautet 251. Das zugehörige Lademodul FEL.SFELLPA(AQESVC03) muss in den Link-Pack-Bereich (LPA) geladen werden und kann dynamisch aktiviert werden. Details hierzu finden Sie in den Abschnitten „SVC-Definitionen in IEASVCxx“ auf Seite 18 und „LPA-Definitionen in LPALSTxx“ auf Seite 19.

Anmerkung: Für Developer for z Systems-Versionen vor Version 9.1.1 wurde ein anderer SVC (AQESVC01) genutzt. Dieser SVC wird nicht mehr verwendet. Er kann gelöscht werden, wenn auf Ihrem Hostsystem nur noch Developer for z Systems ab Version 9.1.1 eingesetzt wird.

SVC-Aktualisierungen für Integrated Debugger

Für Integrated Debugger muss der Supervisoraufruf (SVC) 'FEL.SFELLPA(AQESVC03)' für Developer for z Systems für Ihr System definiert sein. Unter idealen Bedingungen wird dieser SVC beim einleitenden Programmladen des Systems aktiviert oder aktualisiert. Da einleitendes Programmladen nicht immer möglich ist, führt Integrated Debugger mithilfe der gestarteten Task des Debug Managers eine dynamische Installation oder Aktualisierung durch.

Beim Starten überprüft die gestartete Task des Debug Managers, DBGMGR, die folgenden Gegebenheiten und führt dann die entsprechende Aktion aus, wie in Tabelle 14 beschrieben.

- Ist der SVC definiert?
- Version des aktiven SVC (falls definiert) und neuen SVC (in SFELLPA)
- Ist das Startargument SVC=svc_number angegeben? Dabei ist svc_number die gewünschte SVC-Nummer

Tabelle 14. Dynamische SVC-Aktualisierung für Integrated Debugger

	SVC definiert	SVC nicht definiert
Versionen von aktivem und neuem SVC stimmen überein	<ul style="list-style-type: none"> • Keine Aktion, SVC ist bereits aktiv und aktuell • SVC-Startargument wird ignoriert 	<ul style="list-style-type: none"> • SVC mithilfe des Lademoduls SFELLPA dynamisch definieren • SVC-Startargument ist erforderlich
Versionen von aktivem und neuem SVC stimmen nicht überein	<ul style="list-style-type: none"> • SVC mithilfe des Lademoduls SFELLPA aktualisieren • SVC-Startargument wird ignoriert 	<ul style="list-style-type: none"> • SVC mithilfe des Lademoduls SFELLPA dynamisch definieren • SVC-Startargument ist erforderlich

TCP/IP-Aktualisierungen für Integrated Debugger

Integrated Debugger verwendet zwei TCP/IP-Ports.

- 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 Kapitel 9, „Sicherheitsdefinitionen“, auf Seite 103.

- OMVS-Segment für die Benutzer-ID, mit der die Anwendung ausgeführt wird, bei der Fehler behoben werden sollen (einschließlich der Benutzer-ID der CICS-Region)
- Gestartete DBGMR-Task
- BPX.SERVER-Berechtigung für die Benutzer-ID der gestarteten Task
- Programmsteuerung für die Ladebibliothek der gestarteten Task
- AQE.AUTHDEBUG.STDPGM-Berechtigung für Benutzer, die für die Fehlerbehebung bei Programmen mit Problemstatus zuständig sind
- AQE.AUTHDEBUG.AUTHPGM-Berechtigung für Benutzer, die für die Fehlerbehebung bei Programmen mit Problemstatus und berechtigten Programmen zuständig sind

Anmerkung: Um die Migration von einer vorhandenen Developer for z Systems-Konfiguration ohne Integrated Debugger zu erleichtern, wird die Beispiel-JCL `FEL.SFELSAMP(AQERACF)` mit RACF-Befehlen bereitgestellt, sodass nur die Sicherheitsdefinitionen für Integrated Debugger definiert werden.

Language Environment-Aktualisierungen für Integrated Debugger

Mit Integrated Debugger kann ein Language Environment-Benutzerexit (LE) verwendet werden, der TEST-Laufzeitooptionen aus einer Datei für ein gegebenes Modul liest. Dies ist nützlich für das Debugging von Code in Subsystemen, z. B. IMS-Transaktionen und gespeicherten DB2-Prozeduren, da es nicht möglich ist, in diesem Code dynamisch TEST-Laufzeitooptionen bereitzustellen.

Die nächsten Schritte beschreiben den Prozessfluss:

1. LE wird aufgefordert, ein Modul auszuführen.
2. LE ruft den Benutzerexit mit dem Modulnamen als Argument auf.
3. Der Benutzerexit öffnet die Datei mit den TEST-Laufzeitooptionen und sucht nach den TEST-Laufzeitooptionen für das Modul.
4. Der Benutzerexit gibt die TEST-Laufzeitooptionen an LE zurück.
5. LE führt das Modul mit den angegebenen TEST-Laufzeitooptionen aus.

Anmerkung:

- IBM Debug Tool for z/OS verwendet eine erweiterte Version des hier beschriebenen Benutzerexitmechanismus. Um die Koexistenz zu unterstützen und doppelten Aufwand für die Verwaltung der TEST-Laufzeitooptionen zu vermeiden, verwendet Integrated Debugger dieselben Standardwerte wie Debug Tool für die Namen der Dateien mit den Benutzerexitoptionen und den TEST-Laufzeitooptionen.
- Der Benutzerexit unterstützt Variablensubstitution für Benutzer-ID und Modulnamen im Namen der Datei mit den TEST-Laufzeitooptionen.

- Host und Client von Developer for z Systems sind erforderlich, um die Datei mit den TEST-Laufzeitoptionen mithilfe der grafischen Benutzerschnittstelle des Developer for z Systems-Clients zu verwalten.
- Integrated Debugger verwendet CICS CADP, um TEST-Laufzeitoptionen für CICS-Transaktionen bereitzustellen. Weitere Informationen zu CADP finden Sie in der CICS TS-Dokumentation.

Die folgenden Schritte sind nur erforderlich, wenn der LE-Benutzerexit verwendet wird.

- Passen Sie den Job AQED3CXT an und übergeben Sie ihn, um den Benutzerexit zu erstellen. AQED3CXT befindet sich in FEL.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben.
- Passen Sie den Job AQED3CEE an und übergeben Sie ihn, um die angepassten LE-Ladmodule zu erstellen, die den Benutzerexit aufrufen. AQED3CEE befindet sich in FEL.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Standardmäßig werden die angepassten LE-Ladmodule (CEE*) in FEL.#CUST.LOAD gespeichert.

Anmerkung: Speichern Sie die Ladebibliothek mit den angepassten LE-Ladmodulen aus Leistungsgründen nicht in LINKLIST. Der Benutzerexit würde für jedes LE-aktivierte Lademodul aufgerufen, einschließlich erworbener Software.

CICS-Aktualisierungen für Integrated Debugger

Um Fehler bei CICS-Transaktionen zu beheben, sind die folgenden CICS-Aktualisierungen für Integrated Debugger erforderlich.

- Aktualisierungen für Parameter der CICS-Systeminitialisierung (SIT):
 - Geben Sie DEBUGTOOL=YES an.
 - Geben Sie TCPIP=YES an.
 - Geben Sie LLACOPY=YES an, wenn LINKLIST ein Lademodul aus der DD-Verkettung DFHRPL abrufen muss.
- Aktualisierungen für die CICS-JCL:
 - Geben Sie für die Anweisung EXEC den Wert REGION=0M an.
 - Definieren Sie die Ladebibliothek FEL.SFELAUTH in der DD-Anweisung DFHRPL der Region. Wenn der SIT-Parameter LLACOPY=YES angegeben wird, kann sich die Bibliothek auch in LINKLIST befinden.
 - Definieren Sie die Ladebibliothek SYS1.MIGLIB in der DD-Anweisung DFHRPL der Region. Wenn der SIT-Parameter LLACOPY=YES angegeben wird, kann sich die Bibliothek auch in LINKLIST befinden.
 - Definieren Sie für z/OS ab Version 1.13 die Ladebibliothek SYS1.SIEAMIGE in der DD-Anweisung DFHRPL der Region. Wenn der SIT-Parameter LLACOPY=YES angegeben wird, kann sich die Bibliothek auch in LINKLIST befinden. Nähere Erläuterungen hierzu finden Sie in den Informationen zur z/OS-Binder-API im Abschnitt „Parmlib-Aktualisierungen für Integrated Debugger“ auf Seite 80.

Anmerkung:

- Für die Benutzer-ID der CICS-Region ist die Berechtigung UPDATE für das Profil CSVLLA.dataset in der Klasse FACILITY erforderlich, damit der SIT-Parameter LLACOPY=YES ordnungsgemäß funktioniert.

- Damit Integrated Debugger Programme debuggen kann, die in COBOL V4 geschrieben sind, muss Integrated Debugger über Zugriff auf eine Listendatei (PDS oder PDS/E) verfügen. Der Dateiname kann über die Umgebungsvariable AQE_DBG_V4LIST oder die DD-Anweisung AQEV4LST angegeben werden. Ist keine dieser Angaben vorhanden, erstellt Integrated Debugger den Dateinamen durch das Ersetzen des letzten Qualifikationsmerkmals für die Datei der ausführbaren Funktion (z. B. .LOAD) durch die Angabe .LISTING. Erkundigen Sie sich bei den verantwortlichen Entwicklern, welche Methode an Ihrem Standort verwendet wird.
- 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 FEL.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Anmerkung:

- Integrated Debugger setzt voraus, dass TCP/IP-basierte Kommunikation in Ihrer CICS-Region aktiviert ist. z/OS Communication Server IP CICS Sockets ist die bevorzugte Methode hierfür. Informationen zur Konfiguration dieses Features enthält das Handbuch *Communication Server IP CICS Sockets Guide (SC31-8807)*.
- Integrated Debugger verwendet CICS CADP, um TEST-Laufzeitoptionen für CICS-Transaktionen bereitzustellen. Weitere Informationen zu CADP enthält die CICSTS-Dokumentation.

IMS-Aktualisierungen für Integrated Debugger

Um Fehler bei IMS-Transaktionen zu beheben, sind die folgenden IMS-Aktualisierungen für Integrated Debugger erforderlich:

- Aktualisierungen für die JCL der IMS-Nachrichtenregion (DFSMPR):
 - Geben Sie für die Anweisung EXEC der Region den Wert REGION=0M an.
 - Fügen Sie die Ladebibliothek FEL.SFELAUTH zur STEPLIB hinzu, falls sie sich nicht in LINKLIST befindet. Die APF-Autorisierung muss nur für alle STEPLIST-Dateien beibehalten werden, wenn sie für andere Produkte als Integrated Debugger erforderlich ist.
 - Wenn Sie den Language Environment-Benutzerexit (LE) verwenden, fügen Sie die Ladebibliothek mit den angepassten LE-Ladmodulen (standardmäßig FEL.#CUST.LOAD) zur STEPLIB hinzu.

Gespeicherte DB2-Prozeduraktualisierungen für Integrated Debugger

Um Fehler bei gespeicherten DB2-Prozeduren zu beheben, sind die folgenden DB2-bezogenen Aktualisierungen für Integrated Debugger erforderlich.

- Aktualisierungen für die JCL der gestarteten Task für den Adressraum der gespeicherten DB2-Prozedur:
 - Geben Sie für die Anweisung EXEC den Wert REGION=0M an.
 - Fügen Sie die Ladebibliothek FEL.SFELAUTH zur STEPLIB hinzu, falls sie sich nicht in LINKLIST befindet. Die APF-Autorisierung muss nur für alle STEPLIST-Dateien beibehalten werden, wenn sie für andere Produkte als Integrated Debugger erforderlich ist.

- Wenn Sie den Language Environment-Benutzerexit (LE) verwenden, fügen Sie die Ladebibliothek mit den angepassten LE-Lademodulen (standardmäßig FEL.#CUST.LOAD) zur STEPLIB hinzu.

ELAXF*-Aktualisierungen für Integrated Debugger

Die folgenden Aktualisierungen sind für die Verwendung von Integrated Debugger in den von Developer for z Systems bereitgestellten fernen Buildprozeduren erforderlich. Nähere Informationen zur Funktionsweise und Position dieser JCL-Prozeduren finden Sie im Abschnitt „ELAXF* - ferne Buildprozeduren“ auf Seite 24.

- Aktualisierungen der JCL-Prozeduren ELAXFDCL, ELAXFG0 und ELAXFTS0:
 - Geben Sie für die Anweisung EXEC den Wert REGION=0M an.
 - Fügen Sie die Ladebibliothek FEL.SFELAUTH zur STEPLIB hinzu, falls sie sich nicht in LINKLIST befindet. Die APF-Autorisierung muss nur für alle STEP-LIST-Dateien beibehalten werden, wenn sie für andere Produkte als Integrated Debugger erforderlich ist.
 - Wenn Sie den Language Environment-Benutzerexit (LE) verwenden, fügen Sie die Ladebibliothek mit den angepassten LE-Lademodulen (standardmäßig FEL.#CUST.LOAD) zur STEPLIB hinzu.

Kapitel 6. Hostbasierte Codeanalyse

Ähnlich wie der Developer for z Systems-Client unterstützt der Developer for z Systems-Host die Ausführung von Codeanalysetools, die als separates Produkt unter der Bezeichnung IBM Rational Developer for z Systems 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 IBM Rational Developer for z Systems Host Utilities, wie in *Program Directory for IBM Rational Developer for z Systems 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

Anmerkung: IBM Debug Tool for z/OS ist eine Voraussetzung für die Codeabdeckungskomponente.

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 z Systems 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 z Systems auf, ihrem Aufrufjob eine JCLLIB-Karte nach der Jobkarte hinzuzufügen.

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

Verarbeitung der Codeüberprüfung ändern

Die Codeüberprüfung von Developer for z Systems 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 z Systems-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 über einen ephemeren Port mit IBM Debug Tool for z/OS oder mit der Debug-Engine von Integrated Debugger eines Clients von Developer for z Systems einrichtet.

Developer for z Systems Host Utilities bietet zwei Möglichkeiten, Codeabdeckung im Batch-Modus aufzurufen: eine JCL-Beispielprozedur, um eine einzelne Programmausführung zu verarbeiten, und einen Satz von Scripts zum Starten und Stoppen eines dauerhaft aktiven Kollektors für Codeabdeckung, der mehrere Programmausführungen verarbeiten kann.

Einzelaufruf der Codeabdeckung

Die Beispielprozedur AKGCC stellt eine Methode bereit, einen Kollektor für Codeabdeckung zu starten, eine einzelne Programmausführung damit zu analysieren, den Kollektor zu stoppen und die Ergebnisse für eine spätere Verwendung zu archivieren.

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 z Systems auf, ihrem Aufrufjob eine JCLLIB-Karte nach der Jobkarte hinzuzufügen.

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

Mehrfachaufrufe der Codeabdeckung

Abhängig davon, wie Ihr Softwareentwicklungsprozess konfiguriert ist, wäre es zwar praktisch, wenn sich die Prozedur AKGCC um alles kümmert, letztlich ist aber der bedeutende Ressourcen- und Zeitaufwand für das Starten eines Kollektors für Codeabdeckung bei jeder Programmanalyse das entscheidendere Argument.

Developer for z Systems Host Utilities stellt das Script `ccstart` bereit, um einen dauerhaft aktiven Kollektor für Codeabdeckung zu starten. Dieser Kollektor kann dann in Mehrfachaufrufen der Codeabdeckung verwendet werden. Mit dem Script `ccstop` kann der Kollektor gestoppt werden.

Diese Scripts (`ccstart`, `ccstop` und `codecov`) befinden sich im Verzeichnis `/usr/lpp/rdzutil/bin/`, wenn Sie Developer for z Systems Host Utilities an der Standardposition installiert haben. Die Verwendung dieser Scripts wird im Developer for z Systems IBM Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html) erläutert.

Im Folgenden finden Sie ein generisches Verwendungsszenario:

1. Rufen Sie `ccstart` mit der Option für die Verwendung einer festen Portnummer und der Option für die Weiterleitung der Ausgabe an eine bekannte Speicherposition auf.
2. Rufen Sie so oft wie erforderlich zu analysierende Programme mit der Startoption `TEST(,,,TCPIP<hostip>:<port>)` auf.
3. Rufen Sie `ccstop` auf.
4. Richten Sie den Developer for z Systems-Client auf die Ausgabeposition, um die Berichte anzuzeigen.

Ausgabe der Codeabdeckung

Die Ausgabe der Codeabdeckung ist für den Import in einen Developer for z Systems-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 z Systems Host Utilities 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

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 z Systems:

- „include.conf: Erzwungene Includes für C/C++-Content-Assist“

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

- „z/OS UNIX-Unterprojekte“ auf Seite 92
- „Unterstützung für Include-Vorprozessor“ auf Seite 93
- „xUnit-Unterstützung für Enterprise COBOL und PL/I“ auf Seite 94
- „Enterprise Service Tools-Unterstützung“ auf Seite 95
- „Unterstützung bidirektionaler Sprachen für CICS“ auf Seite 95
- „IRZ-Diagnosenachrichten für Enterprise Service Tools“ auf Seite 96
- „FEKRNPLI - ferner PL/I-Vorprozessor“ auf Seite 97

include.conf: 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 rdz.env mithilfe der Variablen include.c oder include.cpp auf die Datei verwiesen werden. Dieser Verweis in rdz.env bedeutet, dass Sie für die Verwendung durch C und C++ eine andere Datei angeben können. Die Variablen in rdz.env sind standardmäßig inaktiviert.

Die Beispieldatei include.conf befindet sich in /etc/zexpl/, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. 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 in derselben Zeile sind nicht 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 23. "include.conf": Erzwungene Includes für C/C++-Content-Assist

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 vergleichbarer Service, allerdings ist hier die gesamte Kommunikation verschlüsselt. Developer for z Systems nutzt beide Services für ferne (hostbasierte) Aktionen in z/OS UNIX-Unterprojekten.

Anmerkung:

- Developer for z Systems 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 z Systems 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 93.

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 Developer for System z (SC14-7301)*, das in der Bibliothek von Developer for z Systems 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

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Developer for z Systems unterstützt die Interpretation und Erweiterung von Include-Anweisungen von COBOL und PL/I, einschließlich ausgewählter Include-Anweisungen von Fremdanbietern. Developer for z Systems stellt auch die Beispiel-REXX-Exec FEKRNPLI bereit, die von dem Developer for z Systems-Client aufgerufen werden kann, um die PL/I-Quelle durch Aufrufen des PL/I-Compilers zu erweitern.

FEKRNPLI befindet sich in `FEL.#CUST.CNTL`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEL.SFELSAMP(FELSETUP)` keine andere Position angegeben haben. Weitere Details finden Sie in Kapitel 2, „Basisanpassung“, auf Seite 15.

Passen Sie die Beispiel-Exec `FEL.#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 z Systems-Client verwendet zum Ausführen der Exec den TSO-Befehlsservice. 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-Befehlsservice platziert wird. Der Benutzer muss nur den Namen kennen. Bei Verwendung des Legacy ISPF Gateway wird die Verknüpfung `SYSPROC` bzw. `SYSEXEC` in `ISPF.conf` definiert. Informationen zum Anpassen dieser Datei finden Sie im Handbuch *IBM Explorer for z/OS Host Configuration Guide* (IBM Form SC27-8437).

xUnit-Unterstützung für Enterprise COBOL und PL/I

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 z Systems 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 FEL.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 z Systems-Client im Batch-Modus, über die TSO-Befehlszeile und über die z/OS UNIX-Befehlszeile aufgerufen werden.

- Developer for z Systems stellt die Beispielprozedur AZUZUNIT bereit, um das Aufrufen des zUnit-Test Runners im Batch-Modus zu vereinfachen. AZUZUNIT befindet sich in FEL.#CUST.PROCLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie im Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Passen Sie die Beispielprozedur FEL.#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 z Systems-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 z Systems auf, ihrem Aufrufjob eine JCLLIB-Karte nach der JOB-Karte hinzuzufügen.

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

- Zum Aufrufen des zUnit-Test Runners von z/OS UNIX (mithilfe des Scripts /usr/lpp/ibm/rdz/bin/zunit) können Sie die erforderlichen Dateien, die keine LINKLIST-Dateien sind, in der Anweisung STEPLIB von rdz.env angeben. Dadurch vereinfacht sich die Konfiguration für den Entwickler.

Die Datei rdz.env befindet sich in /etc/zexpl, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **EDIT** 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 FEL.SFELPROC(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 z Systems stellt Beispielkonvertierungen (z. B. die Konvertierung in das Ant- oder jUnit-Format) bereit, die sich in /usr/lpp/ibm/rdz/samples/zunit/xsd und /usr/lpp/ibm/rdz/samples/zunit/xsl befinden, wenn Sie Developer for z Systems an der Standardposition /usr/lpp/ibm/rdz installiert haben.

Enterprise Service Tools-Unterstützung

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind auch keine speziellen Ressourcen oder Anpassungstasks erforderlich.

Der Developer for z Systems-Client verfügt über eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools'. Abhängig vom generierten Code-Typ liegen diesem Code Funktionen zugrunde, die durch die Hostsysteminstallation von Developer for z Systems bereitgestellt werden. In den folgenden Abschnitten wird beschrieben, wie Sie diese Hostsystemfunktionen verfügbar machen:

- „Unterstützung bidirektionaler Sprachen für CICS“
- „IRZ-Diagnosenachrichten für Enterprise Service Tools“ auf Seite 96

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

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 z Systems 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 (FEL.SFELLOAD) verknüpfen.

Führen Sie die folgenden Tasks aus, um die CICS-Unterstützung für bidirektionale Sprachen zu aktivieren:

1. Stellen Sie die FEL.SFELLOAD-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.

```
CEDA DEF PROG(FEJBDCMP) LANG(LE) G(XXX)
CEDA DEF PROG(FEJBDTRX) LANG(LE) G(XXX)
```

IRZ-Diagnosenachrichten für Enterprise Service Tools

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
 - CSD für die CICS-Region aktualisieren
-

Der Developer for z Systems-Client verfügt über eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools'. Alle IRZM*- und IIRZ*-Module in der Ladebibliothek FEL.SFELMOD 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

Verwenden Sie die folgenden Anweisungen, wenn der generierte Code in einer CICS-Transaktion ausgeführt wird.

- Fügen Sie alle IRZM*- und IIRZ*-Module in FEL.SFELLMOD zur DD "DFHRPL" der CICS-Region hinzu. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch verfügbar sind.
- Passen Sie den IRZCSD-Job an und übergeben Sie ihn, um die CICS-Systemdefinition (CSD) für die CICS-Region zu aktualisieren. Anweisungen zur Anpassung finden Sie in der im Member enthaltenen Dokumentation. IRZCSD befindet sich in FEL.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie im Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Machen Sie in allen anderen Situationen alle Module IRZM* und IIRZ* in der Ladebibliothek FEL.SFELLMOD 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 FEL.SFELLMOD(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.
- In Version 9.0.1 enthalten FEL.SFELLMOD(IRZPWSIO) und das zugehörige PL/I-Beispiel FEL.SFELSAMP(IRZPWSH) Member, die von Developer for z Systems nach IMS Version 12 verschoben wurden. Die Module werden entsprechend in IMS.SDFSRESL(DFSPWSIO) und IMS.SDFSMSPL(DFSPWSH) umbenannt.

FEKRNPLI - ferner PL/I-Vorprozessor

Für diese Anpassungstask benötigen Sie keine Unterstützung. Es sind aber die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich.

Developer for z Systems stellt einen Beispiel-PL/I-Vorprozessor bereit, der vom Developer for z Systems-Client aufgerufen werden kann.

Das Beispielmember FEKRNPLI befindet sich in FEL.#CUST.CNTL, sofern Sie beim Anpassen und Übergeben des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15. Passen Sie die (REXX)-Exec FEL.#CUST.CNTL(FEKRNPLI) für den PL/I-Vorprozessor wie im Member beschrieben an und informieren Sie die Benutzer von Developer for z Systems über die Position der Exec.

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

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 `FEL.SFELAUTH` nicht über eine APF-Berechtigung.

Anmerkung: Starten Sie DBGMGR, bevor Sie mit den anderen auf das Debugging bezogenen IVP-Tests fortfahren.

Services prüfen

Die Installation von Developer for z Systems 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/ibm/rdz/bin/`).

Tabelle 15. Installationsprüfprogramme für Services

<code>fekfivpc</code>	„CARMA-Verbindung“ auf Seite 100
<code>fekfivps</code>	„SCLMDT-Verbindung“ auf Seite 101

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 z/OS Explorer und Developer for z Systems 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-Scripts 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-Script **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/zexpl
$ ./ivpinit
-- RSE_CFG set to /etc/zexpl -- based on current location
-- RSE_HOME set to /usr/lpp/ibm/zexpl -- defined in $RSE_CFG/rse.env
-- added product $PATH to PATH
-- PATH=/etc/zexpl/bin:/usr/lpp/java/J6.0/bin:/usr/lpp/ibm/zexpl/bin:/usr/lpp/i
spf/bin:/usr/lpp/ibm/rdz/bin:/bin
```

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/ibm/rdz/bin/fekfivpc
```

Wenn **./ivpinit** nicht zuerst ausgeführt wird, fordern alle **fekfivp***-Scripts außerdem die Position des Verzeichnisses mit den angepassten Konfigurationsdateien an.

CARMA-Verbindung

Überprüfen Sie die Verbindung zu CARMA, indem Sie den folgenden Befehl ausführen:

```
fekfivpc
```

Der Befehl sollte eine Nachricht über eine erfolgreiche Ausführung zurückgeben.

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 des Installationsprüfverfahrens zu überspringen.

SCLMDT-Verbindung

Überprüfen Sie die Verbindung zum SCLM Developer Toolkit, indem Sie den folgenden Befehl ausführen:

```
fekfivps
```

Der Befehl sollte eine Nachricht über eine erfolgreiche Ausführung zurückgeben.

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 `TEMPDIR` in `rse.env` (Standardwert: `/tmp`).

-debug

Der Parameter `-debug` erstellt eine detaillierte Testausgabe. Verwenden Sie diese Option nur auf Anweisung des IBM Support Center.

Kapitel 9. Sicherheitsdefinitionen

Passen Sie den Beispieljob FELRACF an, der RACF-Beispielbefehle enthält, und übergeben Sie ihn, um die Basissicherheitsdefinitionen für Developer for z Systems zu erstellen. Passen Sie den Beispieljob AQERACF an, der RACF-Beispielbefehle enthält, und übergeben Sie ihn, um die Sicherheitsdefinitionen für Integrated Debugger zu erstellen.

FELRACF und AQERACF befinden sich in FEL.#CUST.JCL, sofern sie bei der Anpassung und Übergabe des Jobs FEL.SFELSAMP(FELSETUP) keine andere Position angegeben haben. Weitere Details finden Sie in „Anpassungskonfiguration“ auf Seite 15.

Weitere Informationen zu RACF-Befehlen finden Sie in der Veröffentlichung *RACF Command Language Reference* (IBM Form SA22-7687).

Voraussetzungen und Prüfliste

Der Sicherheitsadministrator muss die in Tabelle 16 aufgelisteten Werte kennen, um die Sicherheitskonfiguration durchzuführen. Diese Werte wurden in früheren Schritten der Installation und Anpassung von Rational Developer for z Systems definiert.

Tabelle 16. Variablen der Sicherheitskonfiguration

Beschreibung	<ul style="list-style-type: none">StandardwertEntsprechende Quelle	Wert
Übergeordnetes Qualifikationsmerkmal für das Developer for z Systems-Produkt	<ul style="list-style-type: none">FELSMP/E-Installation	
Übergeordnetes Qualifikationsmerkmal für die Developer for z Systems-Anpassung	<ul style="list-style-type: none">FEL.#CUSTFEL.SFELSAMP(FELSETUP), wie in „Anpassungskonfiguration“ auf Seite 15 beschrieben.	
Name der gestarteten Task für Integrated Debugger	<ul style="list-style-type: none">DBGMGRFEL.#CUST.PROCLIB(DBGMGR), wie in „PROCLIB-Änderungen“ auf Seite 23 beschrieben	

Die folgende Liste enthält eine Übersicht über die Aktionen, die zur Durchführung der Basissicherheitskonfiguration von Developer for z Systems erforderlich sind. Um diese Anforderungen zu erfüllen, können je nach erforderlicher Sicherheitsstufe verschiedene Methoden wie in den folgenden Abschnitten dokumentiert verwendet werden.

- „Sicherheitseinstellungen und -klassen aktivieren“ auf Seite 104
- „Gestartete Tasks für Developer for z Systems definieren“ auf Seite 104
- „Debug Manager als sicheren z/OS UNIX-Server definieren“ auf Seite 105

- „Programmgesteuerte MVS-Bibliotheken für Debug Manager definieren“ auf Seite 105
- „Zugriff auf Integrated Debugger definieren“ auf Seite 106
- „Dateiprofile definieren“ auf Seite 106
- „Sicherheitseinstellungen prüfen“ auf Seite 107

Sicherheitseinstellungen und -klassen aktivieren

Developer for z Systems 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 Integrated Debugger
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- Aktivieren von Definitionen gestarteter Tasks für Integrated Debugger
 - SETROPTS GENERIC(STARTED)
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- Aktivieren der Programmsteuerung für Integrated Debugger
 - 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.

Gestartete Tasks für Developer for z Systems definieren

Die folgenden RACF-Beispielbefehle erstellen die gestartete DBGMGR-Task mit der zugeordneten geschützten Benutzer-ID (STCDBM) und der Gruppe STCGROUP.

- ```
ADDGROUP STCGROUP OMVS(AUTOGID)
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
```
- ```
ADDUSER STCDBM DFLTGRP(STCGROUP) NOPASSWORD NAME('DEBUG MANAGER')
OMVS(AUTOUID HOME(/tmp) PROGRAM(/bin/sh) )
DATA('Rational Developer for z Systems')
```
- ```
RDEFINE STARTED DBGMGR.* DATA('DEBUG MANAGER')
STDATA(USER(STCDBM) 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.
- Die gestartete Task von Debug Manager (DBGMR) wird nur von der Funktion 'Integrated Debugger' verwendet.

---

## Debug Manger als sicheren z/OS UNIX-Server definieren

Integrated Debugger benötigt die Zugriffsberechtigung UPDATE für das Profil BPX.SERVER, um die Sicherheitsumgebung für den Debug-Thread erstellen oder löschen zu können. Beachten Sie, dass die Verwendung von UID(0) zum Umgehen dieser Anforderung nicht unterstützt wird. 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(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 Debug Manager definieren

Server mit der Berechtigung für BPX.SERVER müssen in einer sauberen, programm-gesteuerten Umgebung ausgeführt werden. Diese Voraussetzung impliziert, dass alle von Debug Manager aufgerufenen Programme ebenfalls programmgesteuert sein müssen. Die Programmsteuerung von MVS-Ladebibliotheken wird von Ihrer Sicherheitssoftware verwaltet.

Debug Manager verwendet Systembibliotheken, die Language Environment-Laufzeit und die Ladebibliothek (ISP.SISPLoad) von Developer for z Systems.

- RALTER PROGRAM \*\* UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('SYS1.CSSLIB'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)
- RALTER PROGRAM \*\* UACC(READ) ADDMEM('FEL.SFELAUTH'//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 z Systems interagiert, beispielsweise IBM Explorer for z/OS.

- Alternative REXX-Laufzeitbibliothek für SCLM Developer Toolkit
  - REXX.\*.SEAGALT

**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:

- REXX-Laufzeitbibliothek für SCLM Developer Toolkit
  - REXX.\*.SEAGLPA
- Developer for z Systems für CARMA
  - FEL.SFELLPA

---

## Zugriff auf Integrated Debugger definieren

Benutzer müssen über einen Lesezugriff auf eines der aufgelisteten AQE.AUTHDEBUG.\*-Profile verfügen, um Integrated Debugger für die Fehlerbehebung bei Programmen mit Problemstatus einsetzen zu können. Benutzer mit einer Berechtigung für das Profil AQE.AUTHDEBUG.AUTHPGM sind auch zur Fehlerbehebung bei Programmen mit APF-Berechtigung autorisiert. Ersetzen Sie den Platzhalter #apf durch gültige Benutzer-IDs oder RACF-Gruppennamen für die Benutzer, die die Fehlerbehebung bei berechtigten Programmen durchführen dürfen.

- RDEFINE FACILITY AQE.AUTHDEBUG.STDPGM UACC(NONE)
- PERMIT AQE.AUTHDEBUG.STDPGM CLASS(FACILITY) ACCESS(READ) ID(\*)
- RDEFINE FACILITY AQE.AUTHDEBUG.AUTHPGM UACC(NONE)
- PERMIT AQE.AUTHDEBUG.AUTHPGM CLASS(FACILITY) ACCESS(READ) ID(#apf)
- SETROPTS RACLIST(FACILITY) REFRESH

**Anmerkung:** Bei Versionen von IBM Rational Developer for System z vor Version 9.1.1 wurde ein anderes FACILITY-Klassenprofil (AQE.AUTHDEBUG.WRITEBUFFER) eingesetzt, das nicht mehr verwendet wird. Es kann gelöscht werden, wenn auf Ihrem Hostsystem nur noch IBM Rational Developer for System z ab Version 9.1.1 eingesetzt wird.

---

## Dateipprofile definieren

Für die meisten Dateien (Datasets) von Developer for z Systems 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 FEL.#CUST.

- ```
ADDGROUP (FEL) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('IBM Rational Developer for z Systems - HLQ STUB')
```
- ```
ADDSD 'FEL.*.*' UACC(READ)
DATA('IBM Rational Developer for z Systems')
```

```
PERMIT 'FEL.*.***' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
```

- SETROPTS GENERIC(DATASET) REFRESH

**Anmerkung:**

- Schützen Sie FEL.SFELAUTH gegen Aktualisierungen, weil diese Datei eine APF-Berechtigung hat.
- Bei den Beispielbefehlen in dieser Veröffentlichung und im Job FELRACF 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. Ersetzen 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 Komponenten von Developer for z Systems erfordern zusätzliche Sicherheitsdateiprofile. Ersetzen Sie die Platzhalter #sysprog und #ram-developer 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 FEL.#CUST.LSTRANS.FILE.

```
—
ADDSD 'FEL.#CUST.LSTRANS.*.***' UACC(UPDATE)
DATA('IBM Rational Developer for z Systems - SCLMDT')
```

```
—
PERMIT 'FEL.#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 (FEL.#CUST.CRA\*).

```
—
ADDSD 'FEL.#CUST.CRA*.*.***' UACC(READ)
DATA('IBM Rational Developer for z Systems - CARMA')
```

```
—
PERMIT 'FEL.#CUST.CRA*.*.***' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
```

```
—
PERMIT 'FEL.#CUST.CRA*.*.***' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
```

```
—
SETROPTS GENERIC(DATASET) REFRESH
```

---

## Sicherheitseinstellungen prüfen

Verwenden Sie die folgenden Beispielbefehle, um die Ergebnisse Ihrer Anpassungen in Bezug auf die Sicherheit anzuzeigen.

- Sicherheitseinstellungen und -klassen
  - SETROPTS LIST
- Gestartete Tasks
  - LISTGRP STCGROUP OMVS
  - LISTUSER STCDBM OMVS
  - RLIST STARTED DBGMR.\* ALL STDATA

- | • Debug Manager als sicherer z/OS UNIX-Server
- |   – RLIST FACILITY BPX.SERVER ALL
- | • Programmgesteuerte MVS-Bibliotheken für Debug Manager
- |   – RLIST PROGRAM \*\* ALL
- | • Integrierter Debuggerzugriff
- |   – RLIST FACILITY AQE.\*\* ALL
- | • Dateiprofile
- |   – LISTGRP FEL
- |   – LISTDSD PREFIX(FEL) 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 z Systems gearbeitet haben, speichern Sie die zugehörigen angepassten Dateien, bevor Sie ein Upgrade auf diese Version von IBM Rational Developer for z Systems durchführen.
- Wenn Sie planen, mehrere Instanzen von Developer for z Systems auszuführen, lesen Sie "Ausführung mehrerer Instanzen" in der *Hostkonfigurationsreferenz (IBM Form SC27-8578)*.
- 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 z Systems gearbeitet haben, sichern Sie die zugehörigen angepassten Dateien, bevor Sie diese Version von IBM Developer for z Systems installieren.

Anpassbare Developer for z Systems-Dateien sind an den folgenden Positionen zu finden:

- Alle älteren Versionen
  - 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/sc1mdt/\* kopiert, wobei '\*' CONFIG/, CONFIG/PROJECT/ und CONFIG/script/ entspricht.

Vorherige Developer for z Systems-Konfigurationen dokumentieren auch Änderungen an Konfigurationsdateien anderer Produkte.

- Ältere Versionen bis Version 9.0.1
    - SYS1.PARMLIB(IEASCVxx)
  - Alle älteren Versionen
    - SYS1.PARMLIB(BPXPRMxx)
    - SYS1.PARMLIB(COMMNDxx)
    - SYS1.PARMLIB(LPALSTxx)
    - SYS1.PARMLIB(PROGxx)
    - (WLM)
- Ordnet eine Anwendungsumgebung für eine gespeicherte DB2-Prozedur zu.

---

## Migration von Version 9.5 auf Version 9.5.1

In Version 9.5.1 wurden die zu RSE und JES Job Monitor gehörigen Funktionen von IBM Rational Developer for z Systems in ein anderes Produkt (IBM Explorer for z/OS) verschoben. IBM Explorer for z/OS ist nun eine Voraussetzung für IBM Rational Developer for z Systems.

IBM empfiehlt, alle Anpassungen erneut mithilfe der neuen Beispiele von z/OS Explorer und Developer for z Systems zu wiederholen, um sicherzustellen, dass Sie die neuen MVS-Dateinamen und z/OS UNIX-Pfade verwenden.

SMP/E-Dateien:

- z/OS Explorer ist eine SMP/E-Voraussetzung für Developer for z Systems. Dies setzt voraus, dass z/OS Explorer und Developer for z Systems in derselben CSI installiert sein müssen.
- Es gibt keine Änderungen bezüglich SMP/E für Developer for z Systems Host Utilities.

MVS-Dateien:

- z/OS Explorer verwendet das übergeordnete Qualifikationsmerkmal FEK und Dateinamen des Typs SF EK\*. An dieser Stelle ist keine Migrationsaktion erforderlich.
- Developer for z Systems verwendet ein neues übergeordnetes Qualifikationsmerkmal (FEL) und neue Dateinamen des Typs SF EL\*. Überprüfen Sie Ihre aktuellen Verweise auf FEK-Dateien und aktualisieren Sie sie bei Bedarf für die Verwendung von FEL.
  - Sicherheitsdefinitionen in den Klassen DATASET und PROGRAM
  - PROCLIB für gestartete Tasks, Prozeduren für ferne Build-Erstellung und zUnit
  - Konfiguration zum Starten von CARMA
  - JCL zur Verarbeitung von CARMA-VSAMS
  - JCL zum Verarbeiten des LE-Benutzerexits
- Developer for z Systems Host Utilities verwendet das übergeordnete Qualifikationsmerkmal AKG und Dateinamen des Typs SAKG\*. Es ist keine Migrationsaktion erforderlich.

z/OS UNIX-Pfade:

- IBM Explorer for z/OS verwendet neue z/OS UNIX-Pfade: /usr/lpp/ibm/zexpl, /etc/zexpl und /var/zexpl. Überprüfen Sie Ihre aktuellen Verweise auf die alten Pfade (/usr/lpp/rdz, /etc/rdz und /var/rdz) und aktualisieren Sie sie so, dass bei Bedarf die neuen Pfade verwendet werden.
  - PROCLIB für gestartete Tasks
  - JCL-Verweise auf /etc/rdz
  - Beachten Sie, dass /etc/rdz/rsed.envvars mit /etc/zexpl/rse.env ersetzt wird
  - Beachten Sie, dass Protokolle, Push-to-Client und ISPF WORKAREA nach /var/zexpl verschoben wurden
- Developer for z Systems verwendet neue z/OS UNIX-Pfade (/usr/lpp/ibm/rdz und /etc/zexpl). Es verwendet aber immer noch den Pfad /var/rdz. Überprüfen Sie Ihre aktuellen Verweise auf die alten Pfade (/usr/lpp/rdz, /etc/rdz) und aktualisieren Sie sie so, dass bei Bedarf die neuen Pfade verwendet werden.

- JCL zur Verwendung von zUnit-Beispielausgabeumsetzer
- Beachten Sie, dass /etc/rdz/rsed.envvars mit /etc/zexpl/rdz.env ersetzt wird
- Beachten Sie, dass SCLMDT in /var/rdz/sclmdt verbleibt
- Developer for z Systems Host Utilities verwendet den Pfad /usr/lpp/rdzutil.  
Es ist keine Migrationsaktion erforderlich.

## IBM Rational Developer for z Systems - FMID HHOP951

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden von FEK.SFEK\* und /usr/lpp/rdz/\* in FEL.SFEL\* und /usr/lpp/ibm/rdz/\* geändert. Das übergeordnete Qualifikationsmerkmal (High Level Qualifier, HLQ) 'FEK' wird nun von IBM Explorer for z/OS verwendet.
- Die Standardanpassungsposition für MVS- und z/OS UNIX-Komponenten wurde von /etc/rdz in /etc/zexpl geändert.
- Funktionen, die in IBM Explorer for z/OS verschoben wurden
  - Host Configuration Utility wurde in IBM Explorer for z/OS verschoben
    - Konfigurationsaktivitäten für RSE, JES Job Monitor und Legacy ISPF Gateway bleiben
    - Konfigurationsaktivitäten für CARMA, Integrated Debugger und SCLMDT werden entfernt
  - JES Job Monitor wurde in IBM Explorer for z/OS verschoben
    - CNTL(FEJTSO)
    - PROCLIB(JMON)
    - SAMPLIB(FEJJCNFG)
  - Push-to-Client wurde in IBM Explorer for z/OS verschoben
    - /etc/rdz/pushtoclient.properties wird in /etc/zexpl/pushtoclient.properties verschoben
    - /etc/zexpl/pushtoclient.properties ändert /var/rdz -> /var/zexpl
  - RSE wurde in IBM Explorer for z/OS verschoben
    - PROCLIB(RSED)
    - /etc/rdz/rsed.envvars wird nach /etc/zexpl/rse.env verschoben (umbenannt)
    - /etc/rdz/ISPF.conf wird nach /etc/zexpl/ISPF.conf verschoben
    - /etc/rdz/rsecomm.properties wird nach /etc/zexpl/rsecomm.properties verschoben
    - /etc/rdz/ssl.properties wird nach /etc/zexpl/ssl.properties verschoben
    - /etc/zexpl/rse.env enthält nur Anpassungsdefinitionen für z/OS Explorer-Funktionen
  - Systemtools wurden in IBM Explorer for z/OS verschoben
    - JCL(FEKRACF) enthält nur Sicherheitsdefinitionen für z/OS Explorer-Funktionen
    - JCL(FEKSETUP) enthält nur Anpassungsdefinitionen für z/OS Explorer-Funktionen
    - JCL(FEKSETUP) ändert usr/lpp/rdz -> /usr/lpp/ibm/zexpl, /etc/rdz -> /etc/zexpl, /var/rdz -> /var/zexpl
- Funktionen, die in IBM Rational Developer for z Systems verblieben sind
  - C/C++-Content-Assist

- /etc/rdz/include.conf wird in /etc/zexpl/include.conf verschoben
- CARMA
  - CNTL(CRASUBCA) ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - CNTL(CRASUBMT) ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - JCL(CRA\$VCAD) ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - JCL(CRA\$VCAS) ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - JCL(CRA\$VDEF) ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - JCL(CRA\$VMSG) ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - JCL(CRA\$VSTR) ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - JCL(CRA#CIRX) ändert FEK.\* -> FEL.\*
  - JCL(CRA#UADD) ändert FEK.\* -> FEL.\*
  - JCL(CRA#UQRY) ändert FEK.\* -> FEL.\*
  - JCL(CRA#VPDS) ändert FEK.\* -> FEL.\*
  - JCL(CRA#VSLM) ändert FEK.\* -> FEL.\*
  - /etc/rdz/CRASRV.properties wird in /etc/zexpl/CRASRV.properties verschoben
  - /etc/rdz/crastart.conf wird in /etc/zexpl/crastart.conf verschoben
  - /etc/rdz/crastart.endevor.conf wird in /etc/zexpl/crastart.endevor.conf verschoben
  - /etc/zexpl/crastart.conf ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
  - /etc/zexpl/crastart.endevor.conf ändert FEK.\* -> FEL.\*, SFEK\* -> SFEL\*
- Integrated Debugger
  - JCL(AQED3CEE) ändert FEK.\* -> FEL.\*  
Dadurch wird die Standardposition aktualisiert, an der der LE-Benutzerexit gespeichert ist; Benutzer müssen daher möglicherweise die Verwendung ihres LE-Benutzerexits entsprechend aktualisieren.
  - JCL(AQED3CXT) ändert FEK.\* -> FEL.\*
  - JCL(AQERACF) ändert FEK.SFEK\* -> FEL.SFEL\*
  - PROCLIB(DBGMGR) ändert FEK.SFEK\* -> FEL.SFEL\*
- Ferne Buildprozeduren
  - PROCLIB(ELAXFADT) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFASM) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFCOC) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFCOP) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFCOT) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFCP1) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFDCL) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFGO) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFLNK) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFPLP) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFPLT) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFPL1) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFPP1) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFTSO) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFUOP) ändert FEK -> FEL (LODPRFX & STEPLIB)
  - PROCLIB(ELAXFSP) ändert FEK.\* -> FEL.\*

- PROCLIB(ELAXFSQL) ändert FEK.\* -> FEL.\*
  - SQL(FEKTEP2) nach SQL(FELTEP2) verschoben
  - SQL(FEKTIAD) nach SQL(FELTIAD) verschoben
  - Systemtools
    - (Neu) JCL(FELRACF) enthält Sicherheitsdefinitionen für Developer for z Systems-Funktionen
    - (Neu) JCL(FELSETUP) enthält Anpassungsdefinitionen für Developer for z Systems-Funktionen
    - (Neu) /etc/zexpl/rdz.env enthält Anpassungsdefinitionen für Developer for z Systems-Funktionen, die bisher in /etc/rdz/rsed.envvars gespeichert waren
  - zUnit
    - PROCLIB(AZUZUNIT) ändert FEK -> FEL (LODPRFX & STEPLIB)
- Benutzer müssen möglicherweise ihren Prozeduraufruf so aktualisieren, dass der neue Developer for z Systems-Pfad (/usr/lpp/ibm/rdz/samples/zunit/\*) verwendet wird.

## Konfigurierbare Dateien

Tabelle 17 zeigt einen Überblick über Developer for z Systems-Dateien, die in Version 9.5.1 angepasst werden. Die Beispielbibliotheken von Developer for z Systems (FEL.SFELSAMP, FEL.SFELSAMPV und /usr/lpp/ibm/rdz/samples/) enthalten mehr anpassbare Member, als hier aufgeführt sind, wie zum Beispiel CARMA-Beispiel-quellcode und -Jobs für die Kompilierung dieses Codes.

Hinweis: Der Beispieljob 'FELSETUP' kopiert alle aufgelisteten Member in unterschiedliche Dateien und Verzeichnisse (standardmäßig in FEL.#CUST.\* und /etc/zexpl/.\*).

*Tabelle 17. Anpassungen in Version 9.5.1*

| Member/Datei | Standardposition                            | Zweck                                                                                                             | Migrationshinweise                   |
|--------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| FELSETUP     | FEL.SFELSAMP<br>[FEL.#CUST.JCL]             | JCL für die Erstellung von Dateien (Datasets) und Verzeichnissen und zum Füllen derselben mit anpassbaren Dateien | Neu, Konfiguration ist erforderlich. |
| DBGMGR       | FEL.SFELSAMP<br>[FEL.#CUST.JCL]             | JCL für Debug Manager                                                                                             | Verwendung neuer Datei-/Pfadnamen    |
| AQEJCL       | FEL.SFELSAMP<br>[FEL.#CUST.PROCLIB(DBGMGR)] | Name für DBGMGR-Member                                                                                            | Siehe DBGMBR-Member                  |
| ELAXF*       | FEL.SFELSAMP<br>[FEL.#CUST.PROCLIB]         | JCL für ferne Projektbuilds usw.                                                                                  | Verwendung neuer Datei-/Pfadnamen    |
| FELRACF      | FEL.SFELSAMP<br>[FEL.#CUST.JCL]             | JCL für Sicherheitsdefinitionen                                                                                   | Neu, Konfiguration ist erforderlich. |

Tabelle 17. Anpassungen in Version 9.5.1 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                                                             | Migrationshinweise                |
|--------------|-------------------------------------|---------------------------------------------------------------------------------------------------|-----------------------------------|
| AQERACF      | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL für Debug Manager                                                                             | Verwendung neuer Datei-/Pfadnamen |
| CRA\$VMSG    | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstellung der VSAM für CARMA-Nachrichten                                                 | Verwendung neuer Datei-/Pfadnamen |
| CRA\$VDEF    | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstellung der VSAM für CARMA-Konfiguration                                               | Verwendung neuer Datei-/Pfadnamen |
| CRA\$VSTR    | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstellung der VSAM für angepasste CARMA-Informationen                                    | Verwendung neuer Datei-/Pfadnamen |
| CRA\$VCAD    | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstellung der CARMA-Konfigurationsdatei (VSAM) für den CA Endeavor® SCM-RAM              | Verwendung neuer Datei-/Pfadnamen |
| CRA\$VCAS    | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstellung der VSAM-Datei für angepasste CARMA-Informationen für den CA Endeavor® SCM-RAM | Verwendung neuer Datei-/Pfadnamen |
| CRASUBMT     | FEL.SFELSAMP<br>[FEL.#CUST.CNTL]    | CLIST für CARMA-Batch-Start                                                                       | Verwendung neuer Datei-/Pfadnamen |
| CRASUBCA     | FEL.SFELSAMP<br>[FEL.#CUST.CNTL]    | CLIST für CARMA-Batchstart für den CA Endeavor® SCM-RAM                                           | Verwendung neuer Datei-/Pfadnamen |
| CRACFG       | FEL.SFELSAMP<br>[FEL.#CUST.PARMLIB] | CARMA-Interaktionskonfiguration für den CA Endeavor® SCM-RAM                                      | Keine                             |
| CRABCFG      | FEL.SFELSAMP<br>[FEL.#CUST.PARMLIB] | CARMA-Konfiguration für Batchaktionen für den CA Endeavor® SCM-RAM                                | Keine                             |

Tabelle 17. Anpassungen in Version 9.5.1 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                              | Migrationshinweise                       |
|--------------|-------------------------------------|--------------------------------------------------------------------|------------------------------------------|
| CRABATCA     | FEL.SFELSAMP<br>[FEL.#CUST.CNTL]    | CARMA-JCL für<br>Batchaktion für<br>den CA<br>Endevor® SCM-<br>RAM | Keine                                    |
| CRASCL       | FEL.SFELSAMP<br>[FEL.#CUST.PARMLIB] | Schablonen-SCL<br>für CA<br>Endevor® SCM                           | Keine                                    |
| CRASHOW      | FEL.SFELSAMP<br>[FEL.#CUST.PARMLIB] | CARMA-Konfi-<br>guration für den<br>CA Endevor®<br>SCM-RAM         | Keine                                    |
| CRATMAP      | FEL.SFELSAMP<br>[FEL.#CUST.PARMLIB] | CARMA-Konfi-<br>guration für den<br>CA Endevor®<br>SCM-RAM         | Keine                                    |
| CRANDVRA     | FEL.SFELPROC                        | CARMA-<br>Zuordnungs-<br>REXX für den<br>CA Endevor®<br>SCM-RAM    | Keine                                    |
| CRADYNDA     | FEL.SFELPROC                        | CARMA-<br>Zuordnungs-<br>REXX für den<br>CA Endevor®<br>SCM-RAM    | Keine                                    |
| CRAALLOC     | FEL.SFELPROC                        | CARMA-<br>Zuordnungs-<br>REXX                                      | Keine                                    |
| CRA#VSLM     | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstel-<br>lung der VSAM<br>für SCLM-RAM-<br>Nachrichten   | Verwendung<br>neuer Datei-/<br>Pfadnamen |
| CRA#ASLM     | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstel-<br>lung der SCLM-<br>RAM-Dateien                   | Verwendung<br>neuer Datei-/<br>Pfadnamen |
| CRA#VPDS     | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zur Erstel-<br>lung der VSAM<br>für PDS-RAM-<br>Nachrichten    | Verwendung<br>neuer Datei-/<br>Pfadnamen |
| CRA#UADD     | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zum Zu-<br>sammenführen<br>von RAM-Defi-<br>nitionen           | Verwendung<br>neuer Datei-/<br>Pfadnamen |
| CRA#UQRY     | FEL.SFELSAMP<br>[FEL.#CUST.JCL]     | JCL zum Extra-<br>hieren von<br>RAM-Definitio-<br>nen              | Verwendung<br>neuer Datei-/<br>Pfadnamen |
| CRAXJCL      | FEL.SFELSAMP<br>[FEL.#CUST.ASM]     | Beispielquellcode<br>für die Erset-<br>zung von<br>IRXJCL          | Keine                                    |

Tabelle 17. Anpassungen in Version 9.5.1 (Forts.)

| Member/Datei      | Standardposition                           | Zweck                                                                                         | Migrationshinweise                                                                                          |
|-------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| CRA#CIRX          | FEL.SFELSAMP<br>[FEL.#CUST.JCL]            | JCL zur<br>Kompilierung<br>von CRAXJCL                                                        | Verwendung<br>neuer Datei-/<br>Pfadnamen                                                                    |
| AQECSD            | FEL.SFELSAMP<br>[FEL.#CUST.JCL]            | JCL zum Defi-<br>nieren von<br>Integrated<br>Debugger für<br>CICS-Regionen                    | Keine                                                                                                       |
| AQED3CEE          | FEL.SFELSAMP<br>[FEL.#CUST.JCL]            | JCL zum Erstel-<br>len angepasster<br>LE-<br>Laufzeitmodule                                   | Verwendung<br>neuer Datei-/<br>Pfadnamen                                                                    |
| AQED3CXT          | FEL.SFELSAMP<br>[FEL.#CUST.JCL]            | JCL zum Erstel-<br>len des LE-<br>Benutzerexits                                               | Verwendung<br>neuer Datei-/<br>Pfadnamen                                                                    |
| FEKTEP2           | FEL.SFELSAMP<br>[FEL.#CUST.SQL]            | Von ELAXF*<br>verwendete<br>SQL-<br>Befehlseingabe                                            | Keine                                                                                                       |
| FEKTIAD           | FEL.SFELSAMP<br>[FEL.#CUST.JCL]            | Von ELAXF*<br>verwendete<br>SQL-<br>Befehlseingabe                                            | Keine                                                                                                       |
| AZUZUNIT          | FEL.SFELSAMP<br>[FEL.#CUST.PROCLIB]        | JCL-Prozedur<br>für zUnit                                                                     | Verwendung<br>neuer Datei-/<br>Pfadnamen                                                                    |
| FEKRNPLI          | FEL.SFELSAMP<br>[FEL.#CUST.CNTL]           | REXX zum Auf-<br>rufen des PL/I-<br>Compilers<br>innerhalb des<br>Vorprozessor-<br>Frameworks | Keine                                                                                                       |
| IRZCSD            | FEL.SFELSAMP<br>[FEL.#CUST.JCL]            | JCL zum Defi-<br>nieren von<br>Enterprise Ser-<br>vice Tools für<br>CICS                      | Keine                                                                                                       |
| rdz.env           | /usr/lpp/ibm/rdz/samples/<br>[/etc/zexpl/] | RSE-<br>Umgebungsvariablen                                                                    | Ältere Kopien<br>müssen durch<br>diese ersetzt<br>und die Anpas-<br>sungen erneut<br>vorgenommen<br>werden. |
| CRASRV.properties | /usr/lpp/ibm/rdz/samples/<br>[/etc/zexpl/] | CARMA-<br>Konfigurationsdatei                                                                 | Keine                                                                                                       |
| crastart.conf     | /usr/lpp/ibm/rdz/samples/<br>[/etc/zexpl/] | CARMA-<br>Konfigurationsdatei<br>für die Verwen-<br>dung von<br>CRASTART                      | Verwendung<br>neuer Datei-/<br>Pfadnamen                                                                    |

Tabelle 17. Anpassungen in Version 9.5.1 (Forts.)

| Member/Datei          | Standardposition                           | Zweck                                                                                 | Migrationshinweise                |
|-----------------------|--------------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------|
| crastart.endevor.conf | /usr/lpp/ibm/rdz/samples/<br>[/etc/zexpl/] | CARMA-Konfigurationsdatei für die Verwendung von CRASTART für den CA Endevor® SCM-RAM | Verwendung neuer Datei-/Pfadnamen |
| include.conf          | /usr/lpp/ibm/rdz/samples/<br>[/etc/zexpl/] | Erzwungene Includes für C/C++-Content-Assist                                          | Keine                             |

Tabelle 18 zeigt einen Überblick über angepasste Dateien, die in Version 9.5.1 in z/OS Explorer verschoben wurden.

Hinweis: Der Beispieljob 'FEKSETUP' kopiert alle aufgelisteten Member in unterschiedliche Dateien und Verzeichnisse (standardmäßig in 'FEK.#CUST.\*' und '/etc/zexpl/\*').

Tabelle 18. Anpassungen in Version 9.5.1 von z/OS Explorer

| Member/Datei | Standardposition                              | Zweck                                                                                                             | Migrationshinweise                                                                                                              |
|--------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| FEKSETUP     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]               | JCL für die Erstellung von Dateien (Datasets) 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)<br>[FEK.#CUST.PROCLIB] | JCL für JES Job Monitor                                                                                           | Keine                                                                                                                           |
| FEJJJCL      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(JMON)]     | Name für JMON-Member                                                                                              | Siehe JMON-Member                                                                                                               |
| RSED         | FEK.SFEKSAMP (FEKRSED)<br>[FEK.#CUST.PROCLIB] | JCL für den RSE-Dämon                                                                                             | Keine                                                                                                                           |
| FEKRSED      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(RSED)]     | Name für RSED-Member                                                                                              | Siehe RSED-Member                                                                                                               |
| FEKRACF      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]               | JCL für Sicherheitsdefinitionen                                                                                   | Developer for z Systems-spezifische Daten wurden entfernt                                                                       |
| FEJJCNGF     | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB]           | Konfigurationsdatei für JES Job Monitor                                                                           | Keine                                                                                                                           |
| FEJTSO       | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]              | JCL für TSO-Übergabe                                                                                              | Keine                                                                                                                           |

Tabelle 18. Anpassungen in Version 9.5.1 von z/OS Explorer (Forts.)

| Member/Datei            | Standardposition                             | Zweck                                                        | Migrationshinweise                                                                                       |
|-------------------------|----------------------------------------------|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| FEKLOGS                 | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]              | JCL zum Erfassen von Protokolldateien                        | Verwendung neuer Datei-/Pfadnamen<br><br>Developer for z Systems-spezifische Daten wurden entfernt       |
| rse.env                 | /usr/lpp/ibm/zexpl/samples/<br>[/etc/zexpl/] | RSE-Umgebungsvariablen                                       | Ältere Kopien von rsed.envvars müssen durch diese ersetzt und die Anpassungen erneut vorgenommen werden. |
| ISPF.conf               | /usr/lpp/ibm/zexpl/samples/<br>[/etc/zexpl/] | ISPF Gateway-Konfigurationsdatei                             | Verwendung neuer Datei-/Pfadnamen                                                                        |
| ssl.properties          | /usr/lpp/ibm/zexpl/samples/<br>[/etc/zexpl/] | RSE-Konfigurationsdatei für verschlüsselte Kommunikation     | Keine                                                                                                    |
| rsecomm.properties      | /usr/lpp/ibm/zexpl/samples/<br>[/etc/zexpl/] | RSE-Trace-Konfigurationsdatei                                | Keine                                                                                                    |
| pushtoclient.properties | /usr/lpp/ibm/zexpl/samples/<br>[/etc/zexpl/] | Informationen mit Push in die Clientkonfiguration übertragen | Verwendung neuer Datei-/Pfadnamen                                                                        |

## IBM Rational Developer for z Systems Host Utilities - FMID HAKG951

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben daher AKG.\* und /usr/lpp/rdzutil/\*.

### Konfigurierbare Dateien

Tabelle 19 auf Seite 119 zeigt einen Überblick über Dateien, die in Version 9.5.1 angepasst werden. Die Beispielbibliotheken von Developer for z Systems 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 19. Anpassungen in Host Utilities Version 9.0

| Member oder Datei | Standardposition                    | Zweck                                                                                                         | Migrationshinweise |
|-------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------|--------------------|
| AKGSETUP          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL]     | JCL für die Erstellung von Dateigruppen (Datasets) und zum Füllen dieser Dateigruppen mit anpassbaren Dateien | Keine              |
| AKGCC             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeabdeckung                                                                                     | Keine              |
| AKGCR             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeüberprüfung                                                                                   | Keine              |
| AKGCRADD          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL]     | JCL zum Hinzufügen von Code von Drittherstellern zur Codeüberprüfung                                          | Keine              |

## Migration von Version 9.1 auf Version 9.5

Diese Hinweise beziehen sich auf eine Migration von einer Basisversion 9.1 auf Version 9.5. Sie enthält Änderungen, die bereits als Teil der Wartung von Version 9.1 dokumentiert sind. Die Änderungen, die Teil des Wartungsdatenstroms, und daher vermutlich bereits implementiert sind, sind mit dem Release gekennzeichnet, in dem sie eingeführt wurden.

Wir empfehlen, eine vorhandene Datei `rsed.envvars` (standardmäßig im Verzeichnis `/etc/rdz`) durch das neu bereitgestellte Beispiel (standardmäßig im Verzeichnis `/usr/lpp/rdz/samples`) zu ersetzen und die Anpassungen nochmal vorzunehmen.

## IBM Rational Developer for z Systems - FMID HHOP950

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben `FEK.*` und `/usr/lpp/rdz/*`.
- Ab Version 9.5
  - Application Deployment Manager wurde aus dem Produkt entfernt. Die folgenden Definitionen wurden entfernt:
    - `SFEKLOAD(ADN*)`
    - `FEK.#CUST.JCL(ADN*)`
    - `FEK.#CUST.COBOBOL(ADN*)`
    - `FEK.#CUST.ADNREPF0` [VSAM von `ADNVCRD` erstellt]
    - `FEK.#CUST.ADNMANF0` [VSAM von `ADNVMFST` erstellt]
    - `FEK.#CUST.ADN*` - Dateisicherheitsprofile [durchgeführt von `FEKRACF`]
    - CICS-CSD-Definitionen [durchgeführt von `ADNCSDAR`, `ADNCSDRS`, `ADNCSDTX`, `ADNCSDWS`, `ADNVMFST`, `AQECSD`]
    - `usr/lpp/rdz/cicsts`
  - CARMA: Die folgenden anpassbaren Member wurden geändert:
    - `CRACFG` definiert eine neue Option: `DYNAMIC-VB-DATASET-ALLOC`
    - `CRACFG` definiert eine neue Option: `DATASET-ALLOC-OVERRIDE`
  - Enterprise Service Tools: neue anpassbare Member

- IRZCSD: CSD für die CICS-Region aktualisieren
- Integrated Debugger: Das Debugging von CICS-Transaktionen erfordert nun IP CICS Sockets.
- PL/I: neue anpassbare Member
  - FEKRNPLI: REXX für PL/I-Vorprozessor
- Ferne Buildprozeduren: Die folgenden anpassbaren Member wurden geändert:
  - ELAXFDCL: SFEKAUTH zu STEPLIB hinzufügen für Integrated Debugger
  - ELAXFTSO: SFEKAUTH zu STEPLIB hinzufügen für Integrated Debugger
- RSE: Neue Bedienerbefehle wurden hinzugefügt:
  - F rsed,APPL=SEND 'message'[USER={ userid | (userid,userid,...)}]
  - F rsed,APPL=SEND 'message'[PID={ pid | (pid,pid,...)}]
  - F rsed,APPL=SEND CLEARALL
- RSE: Optionale Anweisungen wurden in 'rsed.envvars' hinzugefügt:
  - CGI\_CEATSO
  - CGI\_CEATSO\_KEEPLIVE
  - GSK\_PROTOCOL\_SSLV3
  - GSK\_V3\_CIPHER\_SPECS
- RSE: Optionale Anweisungen wurden in 'rsed.envvars' entfernt:
  - DSTORE\_SSL\_ALGORITHM
- RSE: Erforderliche Anweisungen wurden in rsed.envvars als nicht bearbeitbar markiert:
  - \_RSE\_HOST\_CODEPAGE
- RSE: Unterstützung wurde hinzugefügt für das Senden von Nachrichten an Clients mithilfe eines Bedienerbefehls, eines z/OS UNIX-Befehls oder des TSO-Befehls SEND
- RSE: Unterstützung für Interactive ISPF Gateway wurde hinzugefügt
- zUnit: Die folgenden anpassbaren Member wurden geändert:
  - AZUZUNIT: Unterstützung für Codeabdeckung wurde hinzugefügt
- Ab Version 9.1.1.1
  - Fehlerrückmeldung: FEK.SFEKPROC(FEKFERRF) wurde nach /usr/lpp/rdz/bin/fekferrf.rex verschoben
  - CARMA: Neue anpassbare Member wurden hinzugefügt:
    - CRASCL: CARMA-Beispielschablonen-SCL für CA Endevor® SCM
  - CARMA: Die folgenden anpassbaren Member wurden geändert:
    - CRACFG definiert eine neue Option: PACKAGE-EDITING-OPTION
    - CRACFG definiert eine neue Option: SCL-REQUIRED
    - CRACFG definiert eine neue Option: SCL-DATASET-TEMPLATE
  - Integrated Debugger: Die folgenden anpassbaren Member wurden geändert:
    - AQED3CXT - Neue Debugstufe für LE-Benutzerexit
    - AQED3CEE - Stellt sicher, dass der LE-Laufzeit der richtige Benutzerexit hinzugefügt wird
  - RSE: Optionales LPA-Modul wird erforderlich
    - FEKDSI muss nun über LPA/LINKLIST/STEPLIB verfügbar sein
  - RSE: Neue Bedienerbefehle wurden hinzugefügt:
    - F rsed,APPL=DEBUG {PROCESS | P},{[STORAGE | STOR]},[PID=pid]

- F rsed,APPL=DEBUG {DAEMON | D},{STORAGE | STOR}
- RSE: Produktregistrierung erforderlich bei Kauf als Bestandteil von Produktcode 5697-CDT, IBM Enterprise COBOL Suite for z/OS
- SYS1.PARMLIB: Aktualisiert ein neues PARMLIB-Member:
  - IFAPRDxx: Produktregistrierung
- Ab Version 9.1.1.0
  - CARMA: Die VSAM-Dateien CRADEF und CRASTRS für den CA Endevor® SCM-RAM wurden aktualisiert.
  - CARMA: Die folgenden anpassbaren Member wurden geändert:
    - CRACFG definiert eine neue Option: ALTERNATIVE-ALLOC.
    - CRABCFG definiert eine neue Aktion: TRANSFER-ELEMENT.
    - In CRASUBCA wurde APIHJC hinzugefügt.
    - In der Konfigurationsdatei "crastart.endevor.conf" wurde APIHJC hinzugefügt.
  - Integrated Debugger: Unterstützung der Koexistenz mit IBM Debug Tool for z/OS
  - Integrated Debugger: Die folgenden anpassbaren Member wurden geändert:
    - AQECSD (CICS-CSD-Aktualisierung) definiert neue Ressourcen.
    - AQERACF definiert neue AQE.\*\*-Profile.
    - FEKRACF definiert neue AQE.\*\*-Profile.
    - Integrated Debugger: Erforderliche CICS-CSD-Definitionen wurden hinzugefügt.
      - Warteschlangen AQEM und AQED für temporäre Daten
      - Programm AQEW3Z3, AQEW3Z6, AQEL3Z3, AQEL3Z6, AQEL6Z3, AQEL6Z6
    - Integrated Debugger: Einige CICS-CSD-Definitionen wurden entfernt.
      - Warteschlangen CIGZ und CIBM für temporäre Daten
      - Programm AQEEV006
  - Integrated Debugger: Die folgenden anpassbaren Member wurden geändert:
    - In gestarteter Task AQESTC [DBGMGR] befindet sich nun SFEKLPA in STEPLIB.
  - Integrated Debugger: Erforderlicher SVC wurde hinzugefügt.
    - IEASVCxx-PARMLIB-Änderungen für AQESVC03 (kann dynamisch geladen werden)
  - Integrated Debugger: Optionaler SVC wurde entfernt.
    - IEASVCxx-PARMLIB-Änderungen für AQESVC01
  - Integrated Debugger: Erforderliche RACF-Profile wurden hinzugefügt.
    - AQE.AUTHDEBUG.STDPGM
    - AQE.AUTHDEBUG.AUTHPGM
  - Integrated Debugger: Optionale RACF-Profile wurden entfernt.
    - AQE.AUTHDEBUG.WRITEBUFFER
  - Protokollerfassung: Clients können Hostprotokolle erfassen.
  - RSE: Unterstützung für Kennphrasen wurde hinzugefügt.
  - RSE: Optionale Anweisungen wurden in 'rsed.envvars' hinzugefügt:
    - search.server.limit.scanned\_objects
    - search,server.limit.errcount

- RSE: Optionale Anweisungen wurden in 'rsed.envvars' entfernt:
  - enable.saf.check
  - RSE\_DSICALL
  - search.server.limit.datasets

## Konfigurierbare Dateien

Tabelle 20 enthält eine Übersicht über die Dateien, die in Version 9.5 angepasst werden. Die Beispielbibliotheken von Developer for z Systems (FEK.SFEKSAMP, FEK.SFEKSAMPV und /usr/lpp/rdz/samples/) enthalten mehr anpassbare Member, als hier aufgeführt sind, wie zum Beispiel CARMA-Beispiel Quellcode und Jobs für die Kompilierung dieses Codes.

**Anmerkung:** Der Beispieljob FEKSETUP kopiert alle aufgelisteten Member in verschiedene Dateien und Verzeichnisse (standardmäßig in FEK.#CUST.\* und /etc/rdz/\*).

Tabelle 20. Anpassungen in Version 9.5

| Member/Datei | Standardposition                             | Zweck                                                                                                             | Migrationshinweise                                                                                                              |
|--------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| FEKSETUP     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]              | JCL für die Erstellung von Dateien (Datasets) 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)<br>[FEK.#CUST.PROCLIB] | JCL für JES Job Monitor                                                                                           | Keine                                                                                                                           |
| FEJJJCL      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(JMON)]    | Name für JMON-Member                                                                                              | Siehe JMON-Member                                                                                                               |
| RSED         | FEK.SFEKSAMP(FEKRSED)<br>[FEK.#CUST.PROCLIB] | JCL für den RSE-Dämon                                                                                             | Keine                                                                                                                           |
| FEKRSED      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(RSED)]    | Name für RSED-Member                                                                                              | Siehe RSED-Member                                                                                                               |
| DBGMGR       | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]              | JCL für Debug Manager                                                                                             | SFEKLPA zu STEPLIB hinzugefügt                                                                                                  |
| AQEJCL       | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(DBGMGR)]  | Name für DBGMGR-Member                                                                                            | Siehe DBGMBR-Member                                                                                                             |
| ELAXF*       | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB]          | JCL für ferne Projektbuilds usw.                                                                                  | SFEKAUTH zu STEPLIB hinzugefügt                                                                                                 |
| FEKRACF      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]              | JCL für Sicherheitsdefinitionen                                                                                   | Neue AQE.**-Profile                                                                                                             |
| AQERACF      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]              | JCL für Debug Manager                                                                                             | Neue AQE.**-Profile                                                                                                             |

Tabelle 20. Anpassungen in Version 9.5 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                                                                                         | Migrationshinweise                         |
|--------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| FEKPBITS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Än-<br>dern von<br>Zugriffsberechtigungen<br>für<br>Protokolldateien                                                  | Keine                                      |
| FEJJCNFG     | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | Konfigurationsdatei<br>für JES Job<br>Monitor                                                                                 | Keine                                      |
| FEJTSO       | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | JCL für TSO-<br>Übergabe                                                                                                      | Keine                                      |
| CRA\$VMSG    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Er-<br>stellung der<br>VSAM für<br>CARMA-<br>Nachrichten                                                              | Keine                                      |
| CRA\$VDEF    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Er-<br>stellung der<br>VSAM für<br>CARMA-Kon-<br>figuration                                                           | Keine                                      |
| CRA\$VSTR    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Er-<br>stellung der<br>VSAM für<br>angepasste<br>CARMA-In-<br>formationen                                             | Keine                                      |
| CRA\$VCAD    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Er-<br>stellung der<br>CARMA-<br>Konfigurationsdatei<br>(VSAM) für<br>den CA<br>Endevor®<br>SCM-RAM                   | VSAM-Eingabe<br>wurde geändert.            |
| CRA\$VCAS    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Er-<br>stellung der<br>VSAM-Datei<br>für angepass-<br>te CARMA-<br>Informationen<br>für den CA<br>Endevor®<br>SCM-RAM | VSAM-Eingabe<br>wurde geändert             |
| CRASUBMT     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CLIST für<br>CARMA-<br>Batch-Start                                                                                            | Keine                                      |
| CRASUBCA     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CLIST für<br>CARMA-<br>Batchstart für<br>den CA<br>Endevor®<br>SCM-RAM                                                        | DD APIHJC und<br>neue DDs hinzu-<br>gefügt |

Tabelle 20. Anpassungen in Version 9.5 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                             | Migrationshinweise                       |
|--------------|-------------------------------------|-------------------------------------------------------------------|------------------------------------------|
| CRACFG       | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Interaktionskonfigurationen für den CA Endevor® SCM-RAM     | Optionale Anweisungen wurden hinzugefügt |
| CRABCFG      | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Konfiguration für Batchaktionen für den CA Endevor® SCM-RAM | Optionale Anweisungen wurden hinzugefügt |
| CRABATCA     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CARMA-JCL für Batchaktion für den CA Endevor® SCM-RAM             | Keine                                    |
| CRASCL       | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | Schablonen-SCL für CA Endevor® SCM                                | Neu, Konfiguration ist optional.         |
| CRASHOW      | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Konfiguration für den CA Endevor® SCM-RAM                   | Keine                                    |
| CRATMAP      | FEK.SFEKSAMP<br>[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 zuordnen                         |
| CRADYNDA     | FEK.SFEKPROC                        | CARMA-Zuordnungs-REXX für den CA Endevor® SCM-RAM                 | Neu, Konfiguration ist optional.         |
| CRAALLOC     | FEK.SFEKPROC                        | CARMA-Zuordnungs-REXX                                             | Keine                                    |
| CRA#VSLM     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für SCLM-RAM-Nachrichten              | Keine                                    |
| CRA#ASLM     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der SCLM-RAM-Dateien                           | Keine                                    |

Tabelle 20. Anpassungen in Version 9.5 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                        | Migrationshinweise                                                        |
|--------------|-------------------------------------|--------------------------------------------------------------|---------------------------------------------------------------------------|
| CRA#VPDS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für PDS-RAM-Nachrichten          | Keine                                                                     |
| CRA#UADD     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Zusammenführen von RAM-Definitionen                  | Keine                                                                     |
| CRA#UQRY     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Extrahieren von RAM-Definitionen                     | Keine                                                                     |
| CRAXJCL      | FEK.SFEKSAMP<br>[FEK.#CUST.ASM]     | Beispielquellcode für die Ersetzung von IRXJCL               | Keine                                                                     |
| CRA#CIRX     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Kompilierung von CRAXJCL                             | Keine                                                                     |
| AQECSD       | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Definieren von Integrated Debugger für CICS-Regionen | Neue Ressourcen definieren                                                |
| AQED3CEE     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Erstellen angepasster LE-Laufzeitmodule              | Aktualisiert um sicherzustellen, dass der richtige LE-Exit verwendet wird |
| AQED3CXT     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Erstellen des LE-Benutzerexits                       | Neue Debugstufe für LE-Exit                                               |
| FEKTEP2      | FEK.SFEKSAMP<br>[FEK.#CUST.SQL]     | Von ELAXF* verwendete SQL-Befehlseingabe                     | Keine                                                                     |
| FEKTIAD      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | Von ELAXF* verwendete SQL-Befehlseingabe                     | Keine                                                                     |
| AZUZUNIT     | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB] | JCL-Prozedur für zUnit                                       | SFEKAUTH zu STEPLIB hinzugefügt                                           |

Tabelle 20. Anpassungen in Version 9.5 (Forts.)

| Member/Datei          | Standardposition                     | Zweck                                                                                 | Migrationshinweise                                                                      |
|-----------------------|--------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| FEKRNPLI              | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]     | REXX zum Aufrufen des PL/I-Compilers innerhalb des Vorprozessor-Frameworks            | Keine                                                                                   |
| IRZCSD                | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zum Definieren von Enterprise Service Tools für CICS                              | Neu, Konfiguration ist optional.                                                        |
| FEKLOGS               | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zum Erfassen von Protokolldateien                                                 | Keine                                                                                   |
| rsed.envvars          | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Umgebungsvariablen                                                                | Ältere Kopien müssen durch diese ersetzt und die Anpassungen erneut vorgenommen werden. |
| ISPF.conf             | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | TSO/ISPF-Client-Gateway, Konfigurationsdatei                                          | Keine                                                                                   |
| CRASRV.properties     | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei                                                             | Keine                                                                                   |
| crastart.conf         | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei für die Verwendung von CRASTART                             | Keine                                                                                   |
| crastart.endevor.conf | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei für die Verwendung von CRASTART für den CA Endevor® SCM-RAM | DD APIHJC und neue DDs hinzugefügt                                                      |
| include.conf          | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Erzwungene Includes für C/C++-Content-Assist                                          | Keine                                                                                   |
| ssl.properties        | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Konfigurationsdatei für verschlüsselte Kommunikation                              | Keine                                                                                   |
| rsecomm.properties    | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Trace-Konfigurationsdatei                                                         | Keine                                                                                   |

Tabelle 20. Anpassungen in Version 9.5 (Forts.)

| Member/Datei            | Standardposition                     | Zweck                                                              | Migrationshinweise |
|-------------------------|--------------------------------------|--------------------------------------------------------------------|--------------------|
| pushtoclient.properties | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Informationen mit Push in die Clientkonfigurationsdatei übertragen | Keine              |

## IBM Rational Developer for z Systems Host Utilities - FMID HAKG950

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben daher AKG.\* und /usr/lpp/rdzutil/\*.
- Ab Version 9.1.1.1
  - Codeabdeckung: Produktregistrierung erforderlich bei Kauf als Bestandteil von Produktcode 5697-CDT, IBM Enterprise COBOL Suite for z/OS
  - Codeüberprüfung: Produktregistrierung erforderlich bei Kauf als Bestandteil von Produktcode 5697-CDT, IBM Enterprise COBOL Suite for z/OS
- Ab Version 9.1.1.0
  - Codeüberprüfung: Neue Optionen für die AKGCR-Prozedur
    - SYSLIB: Ersetzt PROPERTY, da für PROPERTY Daten vom Client importiert werden mussten
    - LIST: Liste der Dateien/Member und des zugehörigen Dateityps (ersetzt PDS, MEMBERS und EXTMAP)
    - JUNIT: Neues Ausgabeformat

### Konfigurierbare Dateien

Tabelle 21 enthält eine Übersicht über die Dateien, die in Version 9.5 angepasst werden. Die Beispielbibliotheken von Developer for z Systems 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 21. Anpassungen in Host Utilities Version 9.0

| Member oder Datei | Standardposition                    | Zweck                                                                                                         | Migrationshinweise                    |
|-------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------|---------------------------------------|
| AKGSETUP          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL]     | JCL für die Erstellung von Dateigruppen (Datasets) und zum Füllen dieser Dateigruppen mit anpassbaren Dateien | Keine                                 |
| AKGCC             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeabdeckung                                                                                     | Keine                                 |
| AKGCR             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeüberprüfung                                                                                   | Neue DDs 'SYSLIB', 'LIST' und 'JUNIT' |

Tabelle 21. Anpassungen in Host Utilities Version 9.0 (Forts.)

| Member oder Datei | Standardposition                | Zweck                                                                         | Migrationshinweise |
|-------------------|---------------------------------|-------------------------------------------------------------------------------|--------------------|
| AKGCRADD          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL] | JCL zum Hinzufügen<br>von Code von<br>Drittherstellern zur<br>Codeüberprüfung | Keine              |

## Migrationshinweise für Version 9.1

Die folgenden Migrationshinweise gelten ausschließlich für IBM Rational Developer for z Systems Version 9.1. Diese Hinweise gelten für eine Migration von IBM Rational Developer for z Systems Version 9.1.0 auf 9.1.1 und sind Zusätze zu den vorhandenen Migrationshinweisen für Version 9.1.0.

Alle aufgeführten Änderungen sind ab Version 9.1.1 gültig.

### IBM Rational Developer for z Systems - FMID HHOP910

- CARMA: Die VSAM-Dateien CRADEF und CRASTRS für den CA Endevor<sup>®</sup> SCM-RAM wurden aktualisiert.
- CARMA: Die folgenden anpassbaren Member wurden geändert:
  - CRACFG definiert eine neue Option: ALTERNATIVE-ALLOC.
  - CRABCFG definiert eine neue Aktion: TRANSFER-ELEMENT.
  - In CRASUBCA wurde APIHJC hinzugefügt.
  - In der Konfigurationsdatei "crastart.endevor.conf" wurde APIHJC hinzugefügt.
- Integrated Debugger: Unterstützung der Koexistenz mit IBM Debug Tool for z/OS
- Integrated Debugger: Die folgenden anpassbaren Member wurden geändert:
  - Die gestartete DBGMGR-Task verwendet keine SVC-Nummer mehr.
  - AQECSD (CICS-CSD-Aktualisierung) definiert neue Ressourcen.
  - AQERACF definiert neue AQE.\*\*-Profile.
  - FEKRACF definiert neue AQE.\*\*-Profile.
  - Integrated Debugger: Erforderliche CICS-CSD-Definitionen wurden hinzugefügt.
    - Warteschlangen AQEM und AQED für temporäre Daten
    - Programm AQEW3Z3, AQEW3Z6, AQEL3Z3, AQEL3Z6, AQEL6Z3, AQEL6Z6
  - Integrated Debugger: Einige CICS-CSD-Definitionen wurden entfernt.
    - Warteschlangen CIGZ und CIBM für temporäre Daten
    - Programm AQEEV006
- Integrated Debugger: Erforderlicher SVC wurde hinzugefügt.
  - IEASVCxx-PARMLIB-Änderungen für AQESVC03
- Integrated Debugger: Optionaler SVC wurde entfernt.
  - IEASVCxx-PARMLIB-Änderungen für AQESVC01
- Integrated Debugger: Erforderliche RACF-Profilen wurden hinzugefügt.
  - AQE.AUTHDEBUG.STDPGM
  - AQE.AUTHDEBUG.AUTHPGM
- Integrated Debugger: Optionale RACF-Profilen wurden entfernt.

- AQE.AUTHDEBUG.WRITEBUFFER
- RSE: Unterstützung für Kennphrasen wurde hinzugefügt.
- RSE: Optionale Anweisungen wurden in 'rsed.envvars' hinzugefügt:
  - search.server.limit.scanned\_objects
  - search,server.limit.errcount
- RSE: Optionale Anweisungen wurden in 'rsed.envvars' entfernt:
  - enable.saf.check
  - RSE\_DSICALL
  - search.server.limit.datasets

## IBM Rational Developer for z Systems Host Utilities - FMID HAKG910

- Codeüberprüfung: Neue Optionen für die AKGCR-Prozedur
  - SYSLIB: Ersetzt PROPERTY, da für PROPERTY Daten vom Client importiert werden mussten
  - LIST: Liste der Dateien/Member und des zugehörigen Dateityps (ersetzt PDS, MEMBERS und EXTMAP)
  - JUNIT: Neues Ausgabeformat

---

### Migration von Version 9.0 auf Version 9.1

Diese Hinweise beziehen sich auf eine Migration von einer Basisversion 9.0 auf Version 9.1. Sie enthält Änderungen, die bereits als Teil der Wartung von Version 9.0 dokumentiert sind. Die Änderungen, die Teil des Wartungsdatenstroms, und daher vermutlich bereits implementiert sind, sind mit dem Release gekennzeichnet, in dem sie eingeführt wurden.

Wir empfehlen, eine vorhandene Datei rsed.envvars (standardmäßig im Verzeichnis /etc/rdz) durch das neu bereitgestellte Beispiel (standardmäßig im Verzeichnis /usr/lpp/rdz/samples) zu ersetzen und die Anpassungen nochmal vorzunehmen.

## IBM Rational Developer for z Systems - FMID HHOP910

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben FEK.\* und /usr/lpp/rdz/\*.
- CARMA: Die folgenden anpassbaren Member wurden geändert:
  - CRASUBMT
  - CRASUBCA
- Anpassung: Der JCL-Job FEKSETUP verarbeitet jetzt die neuen Member:
  - AQED3CEE: In FEK.#CUST.JCL(AQED3CEE) kopiert
  - AQED3CXT: In FEK.#CUST.JCL(AQED3CXT) kopiert
  - FEKPBITS: In FEK.#CUST.JCL(FEKPBITS) kopiert
- RSE: Neue optionale Anweisungen wurden in 'rsed.envvars' hinzugefügt:
  - (\_RSE\_JAVAOPTS) -Denable.dDVIPA
  - (\_RSE\_JAVAOPTS) -Dlog.file.mode
  - (\_RSE\_JAVAOPTS) -Dlog.secure.mode
  - (\_RSE\_JAVAOPTS) -Denable.saf.check
  - (\_RSE\_JAVAOPTS) -Dkeep.stats.copy.local
  - (\_RSE\_JAVAOPTS) -Ddebug.miner.localhost

- (\_RSE\_JAVAOPTS) -DDSTORE\_USE\_THREADED\_MINERS
- RSE: Neue Bedienerbefehle wurden hinzugefügt:
  - F rsed,APPL=LOG {USER | AUDIT | NOSERVER | OWNER}
- RSE: Neue Konsolennachrichten wurden hinzugefügt:
  - FEK220I = Host logs are written onto {0}
  - FEK221E = {0} was interrupted because of {1}
  - FEK301E = {0} (uid:{1}) does not own the directory of {2} (file\_owner uid:{3})
  - FEK302E = The requester, {0}, of the LOGS command does not have authority to access the profile of {1}
  - FEK303E = The symbolic link, {0}, cannot be used as a log directory
  - FEK304W = Invalid {0}, {1}, was specified. The default mode, {3}, is used instead.
  - FEK305E = The ID, {0}, does not have appropriate privileges to access {1}.
- Sicherheit: Unterstützung für sichere Protokolldateien wurden hinzugefügt:
  - FEKPBITS: Script zum Ändern von Berechtigungen für vorhandene Protokollinfrastruktur
  - Neues Verhalten: Serverprotokolle werden jetzt im Verzeichnis '\$daemon.log/server' abgelegt.
  - Neues Verhalten: Protokolle werden mit eingeschränkten Zugriffsberechtigungen erstellt.
- Sicherheit: Unterstützung für neue Sicherheitsprofile wurde hinzugefügt:
  - FEK.CMD.LOGS.\*\*
  - FEK.REJECT.\*.UPDATES.system.group
- seit Version 9.0.1.1
  - RSE: Neue optionale Anweisungen wurden in 'rsed.envvars' hinzugefügt:
    - (\_RSE\_JAVAOPTS) -Ddebug.miner.autoreconnect
- seit Version 9.0.1
  - 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 Endeavor<sup>®</sup> SCM-RAM in "crastart.endeavor.conf" und CRASUBCA hinzugefügt:
  - CRAPARM, von CRANDVRA zugeordnet
  - CRACFG
- CARMA: Es wurden zusätzliche DD-Anweisungen für den Nicht-“CA Endeavor<sup>®</sup> 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
  - AQEJCL: 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
- RSE: Neue Konsolennachrichten wurden hinzugefügt:
  - FEK910I = FEK107E = Almost Disk-Full in {0}, existing {1} removed
- zUnit: Es wurden neue optionale Startargumente hinzugefügt:
  - CLOCALE / -l

**Anmerkung:** Um die Migration von einer vorhandenen Developer for z Systems-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.

## Konfigurierbare Dateien

Tabelle 22 auf Seite 132 zeigt einen Überblick über Dateien, die in Version 9.1.0 angepasst werden. Die Beispielbibliotheken von Developer for z Systems (FEK.SFEKSAMP, FEK.SFEKSAMV und /usr/lpp/rdz/samples/) enthalten mehr anpassbare Member, als hier aufgeführt sind, wie zum Beispiel CARMA-Beispiel Quellcode und Jobs für die Kompilierung dieses Codes.

**Anmerkung:** Der Beispieljob FEKSETUP kopiert alle aufgelisteten Member in verschiedene Dateien und Verzeichnisse (standardmäßig in FEK.#CUST.\* und /etc/rdz/\*).

Tabelle 22. Anpassungen in Version 9.1.0

| Member/Datei | Standardposition                              | Zweck                                                                                                             | Migrationshinweise                                                                                                              |
|--------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| FEKSETUP     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]               | JCL für die Erstellung von Dateien (Datasets) 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)<br>[FEK.#CUST.PROCLIB] | JCL für JES Job Monitor                                                                                           | Keine                                                                                                                           |
| FEJJJCL      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(JMON)]     | Name für JMON-Member                                                                                              | Siehe JMON-Member                                                                                                               |
| RSED         | FEK.SFEKSAMP (FEKRSED)<br>[FEK.#CUST.PROCLIB] | JCL für den RSE-Dämon                                                                                             | Keine                                                                                                                           |
| FEKRSED      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(RSED)]     | Name für RSED-Member                                                                                              | Siehe RSED-Member                                                                                                               |
| DBGMGR       | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]               | JCL für Debug Manager                                                                                             | Neu, Anpassung ist optional.                                                                                                    |
| AQEJCL       | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(DBGMGR)]   | Name für DBGMGR-Member                                                                                            | Siehe DBGMBR-Member                                                                                                             |
| ELAXF*       | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB]           | JCL für ferne Projektbuilds usw.                                                                                  | ELAXFGO enthält SFEKAUTH in STEPLIB.                                                                                            |
| FEKRACF      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]               | JCL für Sicherheitsdefinitionen                                                                                   | Neu, gestartete Task DBGMGR                                                                                                     |
| AQERACF      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]               | JCL für Debug Manager                                                                                             | Neu, Anpassung ist optional.                                                                                                    |
| FEKPBITS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]               | JCL zum Ändern von Zugriffsberechtigungen für Protokolldateien                                                    | Neu, Anpassung ist optional.                                                                                                    |
| FEJJCNGF     | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB]           | Konfigurationsdatei für JES Job Monitor                                                                           | Keine                                                                                                                           |
| FEJTSO       | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]              | JCL für TSO-Übergabe                                                                                              | Keine                                                                                                                           |

Tabelle 22. Anpassungen in Version 9.1.0 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                                                             | Migrationshinweise                                                              |
|--------------|-------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| CRA\$VMSG    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für CARMA-Nachrichten                                                 | Keine                                                                           |
| CRA\$VDEF    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für CARMA-Konfiguration                                               | Keine                                                                           |
| CRA\$VSTR    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für angepasste CARMA-Informationen                                    | Keine                                                                           |
| CRA\$VCAD    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der CARMA-Konfigurationsdatei (VSAM) für den CA Endeavor® SCM-RAM              | VSAM-Eingabe wurde geändert.                                                    |
| CRA\$VCAS    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM-Datei für angepasste CARMA-Informationen für den CA Endeavor® SCM-RAM | VSAM-Eingabe wurde geändert.                                                    |
| CRASUBMT     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CLIST für CARMA-Batchstart                                                                        | Hinzugefügte Unterstützung für 8 Startargumente, DD CRAPARM und Zuordnungs-Exec |
| CRASUBCA     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CLIST für CARMA-Batchstart für den CA Endeavor® SCM-RAM                                           | Hinzugefügte Unterstützung für 8 Startargumente, DD CRAPARM und neue DDs        |
| CRACFG       | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Interaktionskonfiguration für den CA Endeavor® SCM-RAM                                      | Neu, Konfiguration ist optional.                                                |

Tabelle 22. Anpassungen in Version 9.1.0 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                              | Migrationshinweise                                                  |
|--------------|-------------------------------------|--------------------------------------------------------------------|---------------------------------------------------------------------|
| CRABCFG      | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Konfiguration für Batchaktionen für den CA Endeavor® SCM-RAM | Keine                                                               |
| CRABATCA     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CARMA-JCL für Batchaktion für den CA Endeavor® SCM-RAM             | Keine                                                               |
| CRASHOW      | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Konfiguration für den CA Endeavor® SCM-RAM                   | Keine                                                               |
| CRATMAP      | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Konfiguration für den CA Endeavor® SCM-RAM                   | Keine                                                               |
| CRANDVRA     | FEK.SFEKPROC                        | CARMA-Zuordnungs-REXX für den CA Endeavor® SCM-RAM                 | Hinzugefügte Unterstützung für Benutzerexit und Zuordnung neuer DDs |
| CRAALLOC     | FEK.SFEKPROC                        | CARMA-Zuordnungs-REXX                                              | Neu, Konfiguration ist optional.                                    |
| CRA#VSLM     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für SCLM-RAM-Nachrichten               | Keine                                                               |
| CRA#ASLM     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der SCLM-RAM-Dateien                            | Keine                                                               |
| CRA#VPDS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für PDS-RAM-Nachrichten                | Keine                                                               |
| CRA#UADD     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Zusammenführen von RAM-Definitionen                        | Keine                                                               |

Tabelle 22. Anpassungen in Version 9.1.0 (Forts.)

| Member/Datei | Standardposition                  | Zweck                                                                            | Migrationshinweise               |
|--------------|-----------------------------------|----------------------------------------------------------------------------------|----------------------------------|
| CRA#UQRY     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Extrahieren von RAM-Definitionen                                         | Keine                            |
| CRAXJCL      | FEK.SFEKSAMP<br>[FEK.#CUST.ASM]   | Beispiel Quellcode für die Ersetzung von IRXJCL                                  | Keine                            |
| CRA#CIRX     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zur Kompilierung von CRAXJCL                                                 | Keine                            |
| AQECSD       | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Definieren von Integrated Debugger für CICS-Regionen                     | Neu, Konfiguration ist optional. |
| AQED3CEE     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Erstellen angepasster LE-Laufzeitmodule                                  | Neu, Konfiguration ist optional. |
| AQED3CXT     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Erstellen des LE-Benutzerexits                                           | Neu, Konfiguration ist optional. |
| ADNCSDRS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Definieren des RESTful-CRD-Servers für die primäre CICS-Region           | Keine                            |
| ADNCSDTX     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Definieren von alternativen Transaktions-IDs für die primäre CICS-Region | Keine                            |
| ADNTXNC      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Erstellen von alternativen Transaktions-IDs                              | Keine                            |
| ADNMSGHC     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zur Kompilierung von ADNMSGHS                                                | Keine                            |
| ADNMSGHS     | FEK.SFEKSAMP<br>[FEK.#CUST.COBOL] | Beispiel Quellcode für den Pipelinenachrichtenhandler                            | Keine                            |
| ADNVCRD      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zur Erstellung des CRD-Repositorys                                           | Keine                            |

Tabelle 22. Anpassungen in Version 9.1.0 (Forts.)

| Member/Datei | Standardposition                     | Zweck                                                                      | Migrationshinweise                                                                       |
|--------------|--------------------------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| ADNCSDWS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zum Definieren des Web-Service-CRD-Servers für die primäre CICS-Region | Keine                                                                                    |
| ADNCSDAR     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zum Definieren des CRD-Servers für nicht primäre CICS-Regionen         | Keine                                                                                    |
| ADNJSPAU     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zur Aktualisierung der CRD-Standardwerte                               | Keine                                                                                    |
| ADNMFST      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zur Erstellung und zum Definieren des Manifestrepositorys              | Keine                                                                                    |
| FEKTEP2      | FEK.SFEKSAMP<br>[FEK.#CUST.SQL]      | Von ELAXF* verwendete SQL-Befehlseingabe                                   | Keine                                                                                    |
| FEKTIAD      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | Von ELAXF* verwendete SQL-Befehlseingabe                                   | Keine                                                                                    |
| AZUZUNIT     | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB]  | JCL-Prozedur für zUnit                                                     | Keine                                                                                    |
| FEKRNPLI     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]     | REXX zum Aufrufen des PL/I-Compilers innerhalb des Vorprozessor-Frameworks | Keine                                                                                    |
| FEKLOGS      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zum Erfassen von Protokolldateien                                      | Keine                                                                                    |
| rsed.envvars | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Umgebungsvariablen                                                     | Ältere Kopien ablesen durch diese ersetzt und die Anpassungen erneut vorgenommen werden. |
| ISPF.conf    | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Konfigurationsdatei für TSO/ISPF Client Gateway                            | Keine                                                                                    |

Tabelle 22. Anpassungen in Version 9.1.0 (Forts.)

| Member/Datei            | Standardposition                     | Zweck                                                                                 | Migrationshinweise                                                              |
|-------------------------|--------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| CRASRV.properties       | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei                                                             | Hinzugefügte Unterstützung für Benutzerexit                                     |
| crastart.conf           | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei für die Verwendung von CRASTART                             | Hinzugefügte Unterstützung für 8 Startargumente, DD CRAPARM und Zuordnungs-Exec |
| crastart.endevor.conf   | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei für die Verwendung von CRASTART für den CA Endevor® SCM-RAM | Hinzugefügte Unterstützung für 8 Startargumente, DD CRAPARM und neue DDs        |
| include.conf            | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Erzwungene Includes für C/C++-Content-Assist                                          | Keine                                                                           |
| ssl.properties          | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Konfigurationsdatei für verschlüsselte Kommunikation                              | Keine                                                                           |
| rsecomm.properties      | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Tracekonfigurationsdatei                                                          | Hinzugefügte Unterstützung für Schlüsselwort USER                               |
| pushtoclient.properties | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Informationen mit Push in die Clientkonfigurationsdatei übertragen                    | Keine                                                                           |

## IBM Rational Developer for z Systems Host Utilities - FMID HAKG910

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben daher AKG.\* und /usr/lpp/rdzutil/\*.
- Codeüberprüfung: Es wurden zusätzliche DD-Anweisungen in AKGCR hinzugefügt.
  - BEXPORT
  - BIMPORT

## Konfigurierbare Dateien

Tabelle 23 zeigt einen Überblick über Dateien, die in Version 9.1 angepasst werden. Die Beispielbibliotheken von Developer for z Systems 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 23. Anpassungen in Host Utilities Version 9.0*

| Member oder Datei | Standardposition                    | Zweck                                                                                                         | Migrationshinweise               |
|-------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------|----------------------------------|
| AKGSETUP          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL]     | JCL für die Erstellung von Dateigruppen (Datasets) und zum Füllen dieser Dateigruppen mit anpassbaren Dateien | Keine                            |
| AKGCC             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeabdeckung                                                                                     | Keine                            |
| AKGCR             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeüberprüfung                                                                                   | Neue DDs 'BIMPORT' und 'BEXPORT' |
| AKGCRADD          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL]     | JCL zum Hinzufügen von Code von Drittherstellern zur Codeüberprüfung                                          | Keine                            |

---

## 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 z Systems - 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 Endeavor® SCM-RAM müssen aktualisiert werden, damit die neue Unterstützung der anpassbaren CA Endeavor® 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 Endeavor® SCM-Batch-Aktionen (ab Version 8.5.1)
- CARMA: Die folgenden anpassbaren Members 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 Endeavor® 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:
  - CRABJOB: Wird in FEK.#CUST.CNTL(CRABJOB) 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 FEJJCNFG hinzugefügt:
  - LOOPBACK\_ONLY (ab Version 9.0)
- JES Job Monitor - Optionale Anweisungen wurden aus FEJJCNFG 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\_SFELoad (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 24 auf Seite 141 enthält eine Übersicht über die Dateien, die in Version 9.0 angepasst werden. Die Beispielbibliotheken von Developer for z Systems (FEK.S-FEKSAMP, 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:

- 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 und Verzeichnisse (standardmäßig in FEK.#CUST.\* und /etc/rdz/\*).

Tabelle 24. Anpassungen in Version 9.0

| Member/Datei | Standardposition                             | Zweck                                                                                                             | Migrationshinweise                                                                                                              |
|--------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| FEKSETUP     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]              | JCL für die Erstellung von Dateien (Datasets) 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)<br>[FEK.#CUST.PROCLIB] | JCL für JES Job Monitor                                                                                           | Keine                                                                                                                           |
| FEJJJCL      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(JMON)]    | Name für JMON-Member                                                                                              | Siehe JMON-Member                                                                                                               |
| RSED         | FEK.SFEKSAMP(FEKRSED)<br>[FEK.#CUST.PROCLIB] | JCL für den RSE-Dämon                                                                                             | Keine                                                                                                                           |
| FEKRSED      | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB(RSED)]    | Name für RSED-Member                                                                                              | Siehe RSED-Member                                                                                                               |
| ELAXF*       | FEK.SFEKSAMP<br>[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<br>[FEK.#CUST.JCL]              | JCL für Sicherheitsdefinitionen                                                                                   | Keine                                                                                                                           |
| FEJJCNFG     | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB]          | Konfigurationsdatei für JES Job Monitor                                                                           | Neue optionale Anweisungen sind hinzugekommen. Vorhandene optionale Anweisungen wurden entfernt.                                |
| FEJTSO       | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]             | JCL für TSO-Übergabe                                                                                              | Keine                                                                                                                           |
| CRA\$VMSG    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]              | JCL zur Erstellung der VSAM für CARMA-Nachrichten                                                                 | Keine                                                                                                                           |

Tabelle 24. Anpassungen in Version 9.0 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                                                            | Migrationshinweise                                                                              |
|--------------|-------------------------------------|--------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| CRA\$VDEF    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für CARMA-Konfiguration                                              | Unterstützung für das Ausschließen von RAM wurde hinzugefügt.                                   |
| CRA\$VSTR    | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für angepasste CARMA-Informationen                                   | Keine                                                                                           |
| CRA\$VCAD    | FEK.SFEKSAMP<br>[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<br>[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<br>[FEK.#CUST.CNTL]    | CLIST für CARMA-Batchstart                                                                       | Keine                                                                                           |
| CRASUBCA     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CLIST für CARMA-Batchstart für den CA Endevor® SCM-RAM                                           | Keine                                                                                           |
| CRABCFG      | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Konfiguration für Batchaktionen für den CA Endevor® SCM-RAM                                | Neue Anweisungen wurden hinzugefügt.                                                            |
| CRABATCA     | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]    | CARMA-JCL für Batchaktion für den CA Endevor® SCM-RAM                                            | Unterstützung für variable Jobkarte wurde hinzugefügt.                                          |

Tabelle 24. Anpassungen in Version 9.0 (Forts.)

| Member/Datei | Standardposition                    | Zweck                                                                  | Migrationshinweise                      |
|--------------|-------------------------------------|------------------------------------------------------------------------|-----------------------------------------|
| CRASHOW      | FEK.SFEKSAMP<br>[FEK.#CUST.PARMLIB] | CARMA-Konfiguration für den CA Endevor® SCM-RAM                        | Keine                                   |
| CRATMAP      | FEK.SFEKSAMP<br>[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. |
| CRA#VSLM     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für SCLM-RAM-Nachrichten                   | Keine                                   |
| CRA#ASLM     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der SCLM-RAM-Dateien                                | Keine                                   |
| CRA#VPDS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Erstellung der VSAM für PDS-RAM-Nachrichten                    | Keine                                   |
| CRA#UADD     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Zusammenführen von RAM-Definitionen                            | Keine                                   |
| CRA#UQRY     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Extrahieren von RAM-Definitionen                               | Keine                                   |
| CRAXJCL      | FEK.SFEKSAMP<br>[FEK.#CUST.ASM]     | Beispielquellen für die Ersetzung von IRXJCL                           | Keine                                   |
| CRA#CIRX     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zur Kompilierung von CRAXJCL                                       | Keine                                   |
| ADNCSDRS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]     | JCL zum Definieren des RESTful-CRD-Servers für die primäre CICS-Region | Keine                                   |

Tabelle 24. Anpassungen in Version 9.0 (Forts.)

| Member/Datei | Standardposition                  | Zweck                                                                            | Migrationshinweise           |
|--------------|-----------------------------------|----------------------------------------------------------------------------------|------------------------------|
| ADNCSDTX     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Definieren von alternativen Transaktions-IDs für die primäre CICS-Region | Keine                        |
| ADNTXNC      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Erstellen von alternativen Transaktions-IDs                              | Keine                        |
| ADNMSGHC     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zur Kompilierung von ADNMSGHS                                                | Keine                        |
| ADNMSGHS     | FEK.SFEKSAMP<br>[FEK.#CUST.COBOL] | Beispielquelle für den Pipelinenachrichtenhandler                                | Keine                        |
| ADNVCRD      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zur Erstellung des CRD-Repositorys                                           | Keine                        |
| ADNCSDWS     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Definieren des Web-Service-CRD-Servers für die primäre CICS-Region       | Keine                        |
| ADNCSDAR     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zum Definieren des CRD-Servers für nicht primäre CICS-Regionen               | Keine                        |
| ADNJSPAU     | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zur Aktualisierung der CRD-Standardwerte                                     | Keine                        |
| ADNMFST      | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]   | JCL zur Erstellung und zum Definieren des Manifestrepositorys                    | Keine                        |
| FEKTEP2      | FEK.SFEKSAMP<br>[FEK.#CUST.SQL]   | Von ELAXF* verwendete SQL-Befehls Eingabe                                        | Neu, Anpassung ist optional. |

Tabelle 24. Anpassungen in Version 9.0 (Forts.)

| Member/Datei          | Standardposition                     | Zweck                                                                                          | Migrationshinweise                                                                                                |
|-----------------------|--------------------------------------|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| FEKTIAD               | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | Von ELAXF*<br>verwendete<br>SQL-<br>Befehlseingabe                                             | Neu, Anpassung ist optional.                                                                                      |
| AZUZUNIT              | FEK.SFEKSAMP<br>[FEK.#CUST.PROCLIB]  | JCL-Prozedur für<br>zUnit                                                                      | Keine                                                                                                             |
| FEKRNPLI              | FEK.SFEKSAMP<br>[FEK.#CUST.CNTL]     | REXX zum Aufrufen des<br>PL/I-Compilers innerhalb des<br>Vorprozessor-Frameworks               | Keine                                                                                                             |
| FEKLOGS               | FEK.SFEKSAMP<br>[FEK.#CUST.JCL]      | JCL zum Erfassen von<br>Protokolldateien                                                       | Es wurden zusätzliche Überprüfungen hinzugefügt. Anpassungen an älteren Dateien müssen erneut vorgenommen werden. |
| rsed.envvars          | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Umgebungsvariablen                                                                         | Ältere Kopien müssen durch diese ersetzt und die Anpassungen erneut vorgenommen werden.                           |
| ISPF.conf             | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Konfigurationsdatei für TSO/<br>ISPF Client<br>Gateway                                         | Keine                                                                                                             |
| CRASRV.properties     | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei                                                                      | Unterstützung für Standardwerte hinzugefügt                                                                       |
| crastart.conf         | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei für die Verwendung von<br>CRASTART                                   | Keine                                                                                                             |
| crastart.endevor.conf | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | CARMA-Konfigurationsdatei für die Verwendung von<br>CRASTART für den CA<br>Endevor®<br>SCM-RAM | Keine                                                                                                             |

Tabelle 24. Anpassungen in Version 9.0 (Forts.)

| Member/Datei            | Standardposition                     | Zweck                                                              | Migrationshinweise |
|-------------------------|--------------------------------------|--------------------------------------------------------------------|--------------------|
| include.conf            | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Erzwungene Includes für C/C++-Content-Assist                       | Keine              |
| ssl.properties          | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Konfigurationsdatei für verschlüsselte Kommunikation           | Keine              |
| rsecomm.properties      | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | RSE-Tracekonfigurationsdatei                                       | Keine              |
| pushtoclient.properties | /usr/lpp/rdz/samples/<br>[/etc/rdz/] | Informationen mit Push in die Clientkonfigurationsdatei übertragen | Keine              |

## IBM Rational Developer for z Systems Host Utilities - FMID HAKG900

Es gibt keine Migrationshinweise, da in Version 8.5 keine entsprechende Funktion vorhanden ist.

### Konfigurierbare Dateien

Tabelle 25 enthält eine Übersicht über die Dateien, die in Version 9.0 angepasst werden. Die Beispielbibliotheken von Developer for z Systems 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 25. Anpassungen in Host Utilities Version 9.0

| Member oder Datei | Standardposition                    | Zweck                                                                                                         | Migrationshinweise               |
|-------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------|----------------------------------|
| AKGSETUP          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL]     | JCL für die Erstellung von Dateigruppen (Datasets) und zum Füllen dieser Dateigruppen mit anpassbaren Dateien | Keine                            |
| AKGCC             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeabdeckung                                                                                     | Keine                            |
| AKGCR             | AKG.SAKGSAMP<br>[AKG.#CUST.PROCLIB] | JCL für die Codeüberprüfung                                                                                   | Neue DDs 'BIMPORT' und 'BEXPORT' |

*Tabelle 25. Anpassungen in Host Utilities Version 9.0 (Forts.)*

| Member oder Datei | Standardposition                | Zweck                                                                         | Migrationshinweise |
|-------------------|---------------------------------|-------------------------------------------------------------------------------|--------------------|
| AKGCRADD          | AKG.SAKGSAMP<br>[AKG.#CUST.JCL] | JCL zum Hinzufügen<br>von Code von<br>Drittherstellern zur<br>Codeüberprüfung | Keine              |



---

## Kapitel 11. Hostkonfigurationsreferenz

In diesem Abschnitt werden die Informationen im Handbuch *IBM Rational Developer for z Systems Hostkonfigurationsreferenz (IBM Form SC27-857)* zusammengefasst. Weitere Details finden Sie in dieser Veröffentlichung.

---

### Wissenswertes zu Developer for z Systems

Das Developer for z Systems-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 z Systems interagiert mit anderen Hostkomponenten, was sich auf  
| die Sicherheit auswirkt.

---

### TCP/IP-Aspekte

Developer for z Systems 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 z Systems keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for z Systems 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.

---

### Hinweise zu Push-to-Client

| Developer for z Systems erweitert den Push-to-Client von z/OS Explorer (der host-  
| basierten Clientsteuerung) mit der Unterstützung für Projektdefinitionen.

---

### CICSTS-Aspekte

Dieses Kapitel enthält nützliche Informationen für CICS Transaction Server-Administratoren.

---

### AT-TLS konfigurieren

Dieser Abschnitt soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von AT-TLS (Application Transparent Transport Layer Security) oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten können.



# Literaturübersicht

## Referenzierte Veröffentlichungen

In diesem Dokument werden die folgenden Veröffentlichungen referenziert:

Tabelle 26. Referenzierte Veröffentlichungen

| Titel der Veröffentlichung                                                              | Formnummer | Bezug                   | Referenzwebsite                                                                                                                   |
|-----------------------------------------------------------------------------------------|------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Program Directory for IBM Rational Developer for z Systems                              | GI11-8298  | Developer for z Systems | <a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a> |
| Program Directory for IBM Rational Developer for z Systems Host Utilities               | GI13-2864  | Developer for z Systems | <a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a> |
|                                                                                         |            |                         |                                                                                                                                   |
| IBM Rational Developer for z Systems Hostkonfiguration                                  | SC43-2896  | Developer for z Systems | <a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a> |
| IBM Rational Developer for z Systems Hostkonfigurationsreferenz                         | SC43-2898  | Developer for z Systems | <a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a> |
| IBM Rational Developer for z Systems Common Access Repository Manager Developer's Guide | SC23-7660  | Developer for z Systems | <a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a> |
| SCLM Developer Toolkit Administrator's Guide                                            | SC23-9801  | Developer for z Systems | <a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a> |
| IBM Explorer for z/OS Host Configuration Guide                                          | SC27-8437  | z/OS Explorer           |                                                                                                                                   |
| IBM Explorer for z/OS Host Configuration Reference                                      | SC27-8438  | z/OS Explorer           |                                                                                                                                   |
| Communications Server IP CICS Sockets Guide                                             | SC31-8807  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| Communications Server IP Configuration Guide                                            | SC31-8775  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| Communications Server IP Configuration Reference                                        | SC31-8776  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| MVS Initialization and Tuning Guide                                                     | SA22-7591  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| MVS Initialization and Tuning Reference                                                 | SA22-7592  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| MVS JCL Reference                                                                       | SA22-7597  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| MVS Planning Workload Management                                                        | SA22-7602  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| MVS System Commands                                                                     | SA22-7627  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| Security Server RACF Command Language Reference                                         | SA22-7687  | z/OS 1.13               | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |

Tabelle 26. Referenzierte Veröffentlichungen (Forts.)

| Titel der Veröffentlichung                          | Formnummer | Bezug     | Referenzwebsite                                                                                                                   |
|-----------------------------------------------------|------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------|
| Security Server RACF Security Administrator's Guide | SA22-7683  | z/OS 1.13 | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| UNIX System Services Command Reference              | SA22-7802  | z/OS 1.13 | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| UNIX System Services Planning                       | GA22-7800  | z/OS 1.13 | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| UNIX System Services User's Guide                   | SA22-7801  | z/OS 1.13 | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |
| Using REXX and z/OS UNIX System Services            | SA22-7806  | z/OS 1.13 | <a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a> |

In diesem Dokument werden die folgenden Websites referenziert:

Tabelle 27. Referenzierte Websites

| Beschreibung                                       | Referenzwebsite                                                                                                                                                                                 |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Developer for z Systems IBM Knowledge Center       | <a href="http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html">http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html</a>                                       |
| Developer for z Systems-Bibliothek                 | <a href="http://www-01.ibm.com/support/docview.wss?uid=swg27038517">http://www-01.ibm.com/support/docview.wss?uid=swg27038517</a>                                                               |
| Developer for z Systems-Homepage                   | <a href="http://www-03.ibm.com/software/products/en/developerforsystemz/">http://www-03.ibm.com/software/products/en/developerforsystemz/</a>                                                   |
| Empfohlener Service für Developer for z Systems    | <a href="http://www-01.ibm.com/support/docview.wss?rs=2294&amp;context=SS2QJ2&amp;uid=swg27006335">http://www-01.ibm.com/support/docview.wss?rs=2294&amp;context=SS2QJ2&amp;uid=swg27006335</a> |
| Verbesserungsvorschlag für Developer for z Systems | <a href="https://www.ibm.com/developerworks/support/rational/rfe/">https://www.ibm.com/developerworks/support/rational/rfe/</a>                                                                 |
| Download von Apache Ant                            | <a href="http://ant.apache.org/">http://ant.apache.org/</a>                                                                                                                                     |

## Veröffentlichungen mit weiteren Informationen

Die folgenden Veröffentlichungen können Antworten auf Fragen enthalten, die vielleicht bei der Konfiguration der erforderlichen Hostsystemkomponenten auftreten.

Tabelle 28. Veröffentlichungen mit weiteren Informationen

| Titel der Veröffentlichung                                                     | Formnummer | Bezug   | Referenzwebsite                                                         |
|--------------------------------------------------------------------------------|------------|---------|-------------------------------------------------------------------------|
| ABCs of z/OS System Programming Volume 9 (z/OS UNIX)                           | SG24-6989  | Redbook | <a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a> |
| System Programmer's Guide to: Workload Manager                                 | SG24-6472  | Redbook | <a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a> |
| TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing       | SG24-7532  | Redbook | <a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a> |
| TCPIP Implementation Volume 3: High Availability, Scalability, and Performance | SG24-7534  | Redbook | <a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a> |

*Tabelle 28. Veröffentlichungen mit weiteren Informationen (Forts.)*

| <b>Titel der Veröffentlichung</b>                                          | <b>Formnummer</b> | <b>Bezug</b> | <b>Referenzwebsite</b>                                                  |
|----------------------------------------------------------------------------|-------------------|--------------|-------------------------------------------------------------------------|
| TCP/IP Implementation<br>Volume 4: Security and<br>Policy-Based Networking | SG24-7535         | Redbook      | <a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a> |
| Tivoli Directory Server for<br>z/OS                                        | SG24-7849         | Redbook      | <a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a> |



---

## Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. IBM stellt dieses Material möglicherweise auch in anderen Sprachen zur Verfügung. Für den Zugriff auf das Material in einer anderen Sprache kann eine Kopie des Produkts oder der Produktversion in der jeweiligen Sprache erforderlich sein.

Möglicherweise bietet IBM die in 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 Corporation  
North Castle Drive, MD-NC119  
92066 Paris La Defense  
US*

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.

IBM kann alle von Ihnen bereitgestellten Informationen beliebig verwenden oder verteilen, ohne dass eine Verpflichtung gegenüber Ihnen entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
92066 Paris La Defense  
US*

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.

Die aufgeführten Leistungsdaten und Clientbeispiele sind nur zur Veranschaulichung gedacht. Tatsächliche Leistungsergebnisse können je nach Konfiguration und Betriebsbedingungen variieren.

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 Eigenschaften machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Die oben genannten Erklärungen bezüglich der Produktstrategien und Absichtserklärungen von IBM stellen die gegenwärtige Absicht von IBM dar, unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

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 Unternehmen sind rein zufällig.

#### **COPYRIGHTLIZENZ:**

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmiertechniken 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.

---

## Informationen zu Programmierschnittstellen

---

### 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](http://www.ibm.com/legal/copytrade.shtml) im Abschnitt "Copyright and trademark information".

---

## Nutzungsbedingungen für die Produktdokumentation

Die Berechtigungen zur Nutzung dieser Veröffentlichungen werden Ihnen auf der Basis der folgenden Bedingungen gewährt.

### Anwendbarkeit

Diese Bedingungen sind eine Ergänzung der Nutzungsbedingungen auf der IBM Website.

### Persönliche Nutzung

Sie dürfen diese Veröffentlichungen für Ihre persönliche, nicht kommerzielle Nutzung unter der Voraussetzung vervielfältigen, dass alle Eigentumsvermerke erhalten bleiben. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM weder weitergeben oder anzeigen noch abgeleitete Werke davon erstellen.

### Kommerzielle Nutzung

Sie dürfen diese Veröffentlichungen nur innerhalb Ihres Unternehmens und unter der Voraussetzung, dass alle Eigentumsvermerke erhalten bleiben, vervielfältigen, weitergeben und anzeigen. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM außerhalb Ihres Unternehmens weder vervielfältigen, weitergeben oder anzeigen noch abgeleitete Werke davon erstellen.

### Rechte

Abgesehen von den hier gewährten Berechtigungen erhalten Sie keine weiteren Berechtigungen, Lizenzen oder Rechte (veröffentlicht oder stillschweigend) in Bezug auf die Veröffentlichungen oder darin enthaltene Informationen, Daten, Software oder geistiges Eigentum.

IBM behält sich das Recht vor, die in diesem Dokument gewährten Berechtigungen nach eigenem Ermessen zurückzuziehen, wenn sich die Nutzung der Veröffentlichungen für IBM als nachteilig erweist oder wenn die obigen Nutzungsbestimmungen nicht genau befolgt werden.

Sie dürfen diese Informationen nur in Übereinstimmung mit allen anwendbaren Gesetzen und Vorschriften, einschließlich aller US-amerikanischen Exportgesetze und Verordnungen, herunterladen und exportieren.

IBM übernimmt keine Gewährleistung für den Inhalt dieser Veröffentlichungen. Diese Veröffentlichungen werden auf der Grundlage des gegenwärtigen Zustands

(auf "as-is"-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung für die Handelsüblichkeit, die Verwendungsfähigkeit für einen bestimmten Zweck oder die Freiheit von Rechten Dritter zur Verfügung gestellt.

---

## Copyrightlizenz

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben wurden. 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.

---

## Marken

IBM, das IBM Logo und [ibm.com](http://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](http://www.ibm.com/legal/copytrade.shtml).

Adobe und PostScript sind Marken von Adobe Systems Incorporated.

Cell Broadband Engine - Sony Computer Entertainment Inc.

Rational ist eine Marke der International Business Machines Corporation und der Rational Software Corporation in den USA und/oder anderen Ländern.

Intel, Intel Centrino, Intel SpeedStep, Intel Xeon, Celeron, Itanium und Pentium sind Marken der Intel Corporation in den USA und/oder anderen Ländern.

IT Infrastructure Library ist eine Marke der Central Computer and Telecommunications Agency.

ITIL ist eine Marke des Cabinet Office (The Minister for the Cabinet Office).

Linear Tape-Open, LTO und Ultrium sind Marken von HP, IBM Corp. und Quantum.

Linux ist eine Marke von Linus Torvalds.

Microsoft, Windows und das Windows-Logo sind Marken oder eingetragene Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

---

# Antwort

IBM Rational Developer for z Systems  
Version 9.5.1  
Konfiguration

IBM Form SC43-2896-00

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.

Als Brief an die Postanschrift auf der Rückseite dieses Formulars

\_\_\_\_\_  
Name

\_\_\_\_\_  
Adresse

\_\_\_\_\_  
Firma oder Organisation

\_\_\_\_\_  
Rufnummer

\_\_\_\_\_  
E-Mail-Adresse

IBM Corporation  
Building 501  
P.O Box 12195  
Research Triangle Park, NC  
USA





Gedruckt in Deutschland

SC43-2896-00

