

Rational IBM Rational Developer for System z
Version 7.6.1

Hostkonfiguration



Rational IBM Rational Developer for System z
Version 7.6.1

Hostkonfiguration



Hinweis

Vor Verwendung dieser Informationen sollten die allgemeinen Hinweise im Abschnitt „Dokumentationshinweise für IBM Rational Developer for System z“ auf Seite 357 gelesen werden.

Ausgabe Mai 2010

Diese Ausgabe bezieht sich auf IBM Rational Developer für System z Version 7.6.1 (Programmnummer 5724-T07) 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 System z Version 7.6.1 Host Configuration,
IBM Form SC23-7658-04,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2005, 2010
© Copyright IBM Deutschland GmbH 2010

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:
SW TSC Germany
Kst. 2877
Mai 2010

Inhaltsverzeichnis

Abbildungsverzeichnis ix

Tabellen xi

Zu diesem Handbuch xiii

Zielgruppe	xiii
Zusammenfassung der Änderungen	xiii
Beschreibung der Dokumentinhalte	xiv
Planung	xiv
Basisanpassung	xiv
Common Access Repository Manager (optional)	xiv
Application Deployment Manager (optional)	xv
SCLM Developer Toolkit (optional)	xv
Weitere Anpassungstasks (optional)	xv
Installationsprüfung	xvi
Bedienerbefehle	xvi
Konfigurationsprobleme lösen	xvi
Sicherheitsaspekte	xvi
Wissenswertes zu Developer for System z	xvi
Hinweise zu WLM	xvii
Optimierungsaspekte	xvii
Leistungsaspekte	xvii
CICSTS-Aspekte	xvii
Anpassung der TSO-Umgebung	xvii
Ausführung mehrerer Instanzen	xvii
Leitfaden für die Migration	xviii
SSL- und X.509-Authentifizierung konfigurieren	xviii
TCP/IP konfigurieren	xviii
INETD konfigurieren	xviii
APPC konfigurieren	xviii
Voraussetzungen	xviii

Teil 1. Anpassung für Developer for System z 1

Kapitel 1. Planung 3

Hinweise zur Migration	3
Hinweise zur Planung	3
Produktübersicht	3
Erforderliche Qualifikationen	4
Zeitbedarf	4
Hinweise zu den Installationsvorbereitungen	4
Konfigurationsoptionen	5
Vorausgesetzte Produkte	5
Erforderliche Ressourcen	5
Hinweise zur Konfigurationsvorbereitung	9
Auslastungsverwaltung	9
Ressourcennutzung und Systembegrenzungen	9
Erforderliche Konfiguration für vorausgesetzte Produkte	9
Hinweise zur Benutzer-ID	9
Hinweise zum Server	10
Konfigurationsmethode	12

Hinweise zu den Deploymentvorbereitungen	12
Prüfliste für den Client	13

Kapitel 2. Basisanpassung 15

Voraussetzungen und Prüfliste	15
Anpassungskonfiguration	15
PARMLIB-Änderungen	16
z/OS UNIX-Grenzwerte in BPXPRMxx festlegen	16
Gestartete Tasks zu COMMNDxx hinzufügen	18
LPA-Definitionen in LPALSTxx	18
APF-Berechtigungen in PROGxx	18
LINKLIST-Definitionen in PROGxx	19
Vorausgesetzte LINKLIST- und LPA-Definitionen	20
LINKLIST-Definitionen für andere Produkte	21
PROCLIB-Änderungen	21
JES Job Monitor	21
RSE-Dämon	23
Sperrdämon	24
Einschränkungen der JCL für die PARM-Variable	24
ELAXF*-Prozeduren für ferne Builderstellung	26
Sicherheitsdefinitionen	28
Konfigurationsdatei für JES Job Monitor (FEJJCNF)	29
RSE-Konfigurationsdatei rsed.envvars	33
Für RSE-Server verfügbaren PORTRANGE definieren	41
Zusätzliche Java-Startparameter mit _RSE_JA-VAOPTS definieren	42
Zusätzliche Java-Startparameter mit _RSE_CMD-SERV_OPTS definieren	47
Konfigurationsdatei TSO/ISPFISPF.conf des TSO/ISPF-Client-Gateways von ISPF	48
Optionale Komponenten	49
Installationsprüfung	49

Kapitel 3. Common Access Repository Manager (optional) 51

Voraussetzungen und Prüfliste	51
CARMA-Komponenten	52
Migrationshinweise für CARMA-VSAM	53
RSE-Schnittstelle zu CARMA	53
CARMA-Serverstart mit Batchübergabe	56
CRASRV.properties anpassen	56
CRASUBMT anpassen	56
Alternativer CARMA-Serverstart mit CRASTART (optional)	57
CRASRV.properties anpassen	58
crastart.conf anpassen	58
Alternativer CARMA-Serverstart mit TSO/ISPF-Client-Gateway (optional)	60
CRASRV.properties anpassen	60
ISPF.conf anpassen	61
Beispiel-RAM (Repository Access Manager) aktivieren (optional)	62
PDS-RAM aktivieren	63
SCLM-RAM aktivieren	63

Skeleton-RAM aktivieren	63
CA Endeavor SCM-RAM aktivieren (optional)	64
Voraussetzungen und Prüfliste	64
CA Endeavor SCM-RAM definieren	65
CA Endeavor SCM-RAM-Start mit Batchübergabe	65
CA Endeavor SCM-RAM-Start mit CRASTART	68
(Optional) CRANDVRA anpassen	69
(Optional) CA Endeavor SCM-RAM anpassen	70
(Optional) Unterstützung mehrerer RAMs	70
Beispiel	70
IRXJCL oder CRAXJCL (optional)	72
CRAXJCL erstellen	72

Kapitel 4. Application Deployment Manager (optional) 73

Voraussetzungen und Prüfliste	73
CRD-Repository	74
CICS-Verwaltungsdienstprogramm	75
RESTful oder Web-Service	75
CRD-Server mit der RESTful-Schnittstelle	75
Primäre CICS-Verbindungsregion	75
Nicht primäre CICS-Verbindungsregionen	76
CRD-Servertransaktions-IDs anpassen (optional)	76
CRD-Server mit der Web-Service-Schnittstelle	77
Handler für Pipelinenachrichten	77
CICS, primäre Verbindungsregion	78
CICS, nicht primäre Verbindungsregionen	79
Manifestrepository (optional)	79

Kapitel 5. SCLM Developer Toolkit (optional) 81

Voraussetzungen und Prüfliste	81
Voraussetzungen	82
Aktualisierung von ISPF.conf für SCLMDT	82
Aktualisierung von rsed.envvars für SCLMDT	83
Umsetzung langer/kurzer Namen (optional)	84
LSTRANS.FILE, die VSAM für die Umsetzung langer/kurzer Namen, erstellen	84
Aktualisierung von rsed.envvars für die Umsetzung langer/kurzer Namen	86
Ant installieren und anpassen (optional)	86
Aktualisierung von SCLM für SCLMDT	87
Alte Dateien aus WORKAREA entfernen	88

Kapitel 6. Weitere Anpassungstasks (optional). 89

Gespeicherte DB2-Prozedur (optional)	89
WLM-Änderungen (Workload Manager)	89
PROCLIB-Änderungen	90
DB2-Änderungen	90
EST-Unterstützung (Enterprise Service Tools) (optional)	91
Unterstützung bidirektionaler Sprachen für CICS (optional)	92
IRZ-Diagnosefehlernachrichten (optional)	93
RSE-SSL-Verschlüsselung (optional)	93
RSE-Trace (optional)	96
Hostbasierte Eigenschaftsgruppe (optional)	98
Hostbasierte Projekte (optional)	99
File Manager-Integration (optional)	100

Nicht editierbare Zeichen (optional)	101
REXEC (oder SSH) verwenden (optional)	102
Ferne (hostbasierte) Aktionen für z/OS UNIX-Unterprojekte	103
Alternative RSE-Verbindungsmethode	103
REXEC (oder SSH) konfigurieren	104
APPC-Transaktion für TSO Commands Service (optional)	104
Vorbereitungen	105
Implementierung	106
Hinweise zur Verwendung von APPC	107
WORKAREA-Bereinigung (optional)	108

Kapitel 7. Installationsprüfung 109

Gestartete Tasks prüfen	109
JMON, JES Job Monitor	109
LOCKD, Sperrdämon	109
RSED, RSE-Dämon	109
Services prüfen	111
Installationsprüfprogramm initialisieren	112
Portverfügbarkeit	113
TCP/IP konfigurieren	114
RSE-Dämonverbindung	115
JES Job Monitor-Verbindung	115
Sperrdämonverbindung	116
Verbindung mit TSO/ISPF-Client-Gateway von ISPF	116
Verbindung zu TSO Commands Service mit APPC (optional)	118
SCLMDT-Verbindung (optional)	118
REXEC-Verbindung (optional)	120
REXEC/SSH-Shell-Script (optional)	121

Teil 2. Informationen zu Developer for System z 123

Kapitel 8. Bedienerbefehle 125

Start (S)	125
JES Job Monitor	125
RSE-Dämon	126
Sperrdämon	126
Modify (F)	127
JES Job Monitor	127
RSE-Dämon	128
Sperrdämon	131
Stop (P)	131
Konsolnachrichten	132
JES Job Monitor	132
RSE-Dämon, RSE-Thread-Pool-Server und Sperrdämon	132
Hinweise zum Lesen eines Syntaxdiagramms	134
Symbole	134
Operanden	134
Syntaxbeispiel	135
Nicht alphanumerische Zeichen und Leerzeichen	135
Mehrere Operanden auswählen	135
Mehrere Zeilen	135
Syntaxfragmente	135

Kapitel 9. Konfigurationsprobleme lösen. 137

Protokoll- und Konfigurationsanalyse mit FEK-LOGS	137
Protokolldateien	138
Protokollierung von JES Job Monitor	139
Protokollierung des Sperrdämons.	139
Protokollierung des RSE-Dämons und des Thread-Pools	139
Protokollierung des RSE-Benutzers	140
Protokollierung der Fault Analyzer-Integration	141
Protokollierung der File Manager-Integration	142
Protokollierung von SCLM Developer Toolkit CARMA-Protokollierung	142
APPC-Transaktionsprotokollierung (TSO Commands Service)	143
Protokollierung des IVP-Tests fekfivpi	143
Protokollierung des IVP-Tests fekfivps	143
Speicherauszugsdateien	144
MVS-Speicherauszüge	144
Java-Speicherauszüge.	144
Positionen für z/OS UNIX-Speicherauszüge	146
Traceerstellung	146
Traceerstellung für JES Job Monitor	146
Traceerstellung für RSE	146
Traceerstellung für den Sperrdämon.	147
Traceerstellung für CARMA	147
Trace für Fehlerrückmeldungen	148
z/OS UNIX-Berechtigungsbits	149
Dateisystemattribut SETUID	149
Programmsteuerung autorisieren	149
APF-Autorisierung	150
Sticky Bit.	151
Reservierte TCP/IP-Ports	152
Größe des Adressraums	153
Anforderungen an die Start-JCL	153
In SYS1.PARMLIB(BPXPRMxx) festgelegte Begrenzungen	154
Im Sicherheitsprofil gespeicherte Begrenzungen	154
Von Systemexits erzwungene Begrenzungen	154
Begrenzungen für die 64-Bit-Adressierung.	154
APPC-Transaktion und TSO Commands Service	155
Weitere Informationen	156
Systemgrenzwerte	156
Bekannte Probleme mit den Voraussetzungen	157
Host-Connect-Emulator	158

Kapitel 10. Sicherheitsaspekte 159

Authentifizierungsmethoden	159
Benutzer-ID und Kennwort.	160
Benutzer-ID und Kennwort für einmaliges Anmelden	160
X.509-Zertifikat.	160
Authentifizierung durch JES Job Monitor	160
Verbindungssicherheit	161
Externe Kommunikation auf angegebene Ports beschränken.	161
Kommunikation mit SSL verschlüsseln	161
Eingangsport überprüfen	162
TCP/IP-Ports	162

Externe Kommunikation.	163
Interne Kommunikation	163
CARMA und TCP/IP-Ports.	164
PassTickets verwenden	164
Prüfprotokollierung	165
Steuerung der Prüffunktion	165
Prüfdaten	166
JES-Sicherheit	166
Aktionen für Beschränkungen der Jobziele	166
Aktionen für Beschränkungen der Jobausführung	168
Zugriff auf Spooldateien.	170
Mit SSL verschlüsselte Kommunikation.	170
Clientauthentifizierung unter Verwendung von X.509-Zertifikaten	171
Prüfung der Zertifizierungsstelle (CA)	172
Zertifikatswiderrufsliste (CRL) abfragen (optional).	173
Authentifizierung durch Ihre Sicherheitssoftware	173
Authentifizierung durch den RSE-Dämon	174
Eingangsport (POE) überprüfen	175
CICSTS-Sicherheit	176
CRD-Repository	176
CICS-Transaktionen	176
Mit SSL verschlüsselte Kommunikation.	176
SCLM-Sicherheit	176
Konfigurationsdateien von Developer for System z	177
JES Job Monitor - FEJJCENFG	177
RSE - rsed.envvars	177
RSE - ssl.properties	178
Sicherheitsdefinitionen	178
Voraussetzungen und Prüfliste	179
Sicherheitseinstellungen und -klassen aktivieren	180
OMVS-Segment für Benutzer von Developer for System z definieren	181
Dateiprofile definieren	181
Gestartete Tasks für Developer for System z definieren	184
JES-Befehlssicherheit definieren	185
RSE-Server als sicheren z/OS UNIX-Server definieren	187
Programmgesteuerte MVS-Bibliotheken für RSE definieren	187
Anwendungsschutz für RSE definieren.	188
PassTicket-Unterstützung für RSE definieren	188
Programmgesteuerte z/OS UNIX-Dateien für RSE definieren	190
Sicherheitseinstellungen prüfen	190

Kapitel 11. Wissenswertes zu Developer for System z 193

Komponentenübersicht	193
RSE als Java-Anwendung	195
Taskeigner	196
Verbindungsflow	198
Sperrdämon	200
Sperrern aufheben	201
z/OS UNIX-Verzeichnisstruktur	202
Aktualisierungsberechtigungen für Benutzer ohne Systemadministratorrechte	203

Kapitel 12. Hinweise zu WLM.	205
Klassifikation für Verarbeitungsprozesse	205
Klassifikationsregeln	206
Ziele festlegen	207
Hinweise zur Zielauswahl	208
STC	209
OMVS.	210
JES	211
ASCH	212
CICS	213

Kapitel 13. Optimierungsaspekte . . . 215

Ressourcennutzung	215
Überblick.	216
Anzahl der Adressräume	217
Anzahl der Prozesse	220
Anzahl der Threads	223
Speicherbelegung	226
Begrenzung für die Größe des Java-Heapspeichers	226
Begrenzung für die Größe der Adressräume	227
Richtlinien für Größenschätzungen	227
Beispielanalyse der Speicherbelegung	228
Speicherbelegung im z/OS UNIX-Dateisystem	232
Definitionen von wichtigen Ressourcen.	235
/etc/rdz/rsed.envvars	235
SYS1.PARMLIB(BPXPRMxx)	236
Definitionen von verschiedenen Ressourcen	238
EXEC-Karte in der Server-JCL	238
FEK.#CUST.PARMLIB(FEJJCENFG)	239
SYS1.PARMLIB(IEASYSxx)	239
SYS1.PARMLIB(IVTPRMxx)	239
SYS1.PARMLIB(ASCHPMxx)	239
Überwachung	240
RSE überwachen	240
z/OS UNIX überwachen	241
Netz überwachen	243
z/OS UNIX-Dateisysteme überwachen	244
Beispielkonfiguration.	244
Thread-Pool-Anzahl	245
Mindestbegrenzungen festlegen	245
Begrenzungen definieren	246
Ressourcennutzung überwachen	247

Kapitel 14. Leistungsaspekte. . . . 249

Dateisystem zFS verwenden	249
Verwendung von STEPLIB vermeiden	249
Zugriff auf Systembibliotheken verbessern	249
LE-Laufzeitbibliotheken (Language Environment)	250
Anwendungsentwicklung	250
Durchsatz der Sicherheitsprüfung verbessern.	251
Auslastungsverwaltung	251
Feste Java-Heapgröße	251
Java-Option -Xquickstart	252
Gemeinsame Klassennutzung durch mehrere JVMs	252
Gemeinsame Klassennutzung aktivieren	253
Cachegrößenbegrenzung	253
Cachesicherheit.	253
SYS1.PARMLIB(BPXPRMxx)	254

Plattenspeicherplatz	254
Dienstprogramme für Cacheverwaltung	254

Kapitel 15. CICSTS-Aspekte 257

RESTful oder Web-Service	258
Primäre und nicht primäre Verbindungsregionen	258
CICS-Ressourceninstallation protokollieren	259
Application Deployment Manager, Sicherheit.	259
Sicherheit des CRD-Repositorys	259
Pipelinesicherheit	259
Transaktionssicherheit	259
Mit SSL verschlüsselte Kommunikation.	261
Ressourcensicherheit	261
Verwaltungsdienstprogramm	261
Migrationshinweise zum Verwaltungsdienstprogramm	266
Nachrichten des Verwaltungsdienstprogramms	266

Kapitel 16. TSO-Umgebung anpassen 269

TSO Commands Service.	269
Zugriffsmethoden	269
TSO/ISPF-Client-Gateway als Zugriffsmethode verwenden	270
Basisanpassung – ISPF.conf.	270
Erweitert – Vorhandene ISPF-Profile verwenden	270
Erweitert – Zuordnungs-Exec verwenden	271
Erweitert – Mehrere Zuordnungs-Execs verwenden.	271
Erweitert – Mehrere 'ISPF.conf'-Dateien mit mehreren Konfigurationen von Developer for System z	272
APPC als Zugriffsmethode verwenden	273
Basisanpassung – JCL für APPC-Transaktion	273
Erweitert – Vorhandene ISPF-Profile verwenden	273
Erweitert – Zuordnungs-Exec verwenden	274
Erweitert – Mehrere APPC-Transaktionen mit mehreren Konfigurationen von Developer for System z	274

Kapitel 17. Mehrere Instanzen ausführen 277

Identische Konfiguration in einem Sysplex	277
Identische Softwareversionen mit unterschiedlichen Konfigurationsdateien	278
Alle anderen Situationen	279

Kapitel 18. Leitfaden für die Migration 283

Hinweise zur Migration	283
Bereits konfigurierte Dateien sichern	283
Migrationshinweise für Version 7.6.1	285
Migration von Version 7.5 auf Version 7.6	286
IBM Rational Developer for System z, FMID HHOP760	286
Konfigurierbare Dateien.	288
Migration von Version 7.1 auf Version 7.5	292
IBM Rational Developer for System z, FMID HHOP750	292
Konfigurierbare Dateien.	293
Migration von Version 7.0 auf Version 7.1	295

IBM Rational Developer for System z, FMID HHOP710	295
IBM Common Access Repository Manager (CARMA), FMID HCMA710	296
Konfigurierbare Dateien	296

Anhang A. SSL- und X.509-Authentifizierung konfigurieren 299

Speicherpositionen für private Schlüssel und Zertifikate festlegen	300
Schlüsseldatei mit RACF erstellen	301
Signiertes Zertifikat verwenden (optional)	302
Vorhandene RSE-Konfiguration klonen	303
Koexistenz durch Aktualisieren von rsed.envvars aktivieren	303
Aktualisierung von ssl.properties durchführen, um SSL zu aktivieren	303
Neuen RSE-Dämon erstellen, um SSL zu aktivieren	304
Verbindung testen	305
Unterstützung der X.509-Clientauthentifizierung hinzufügen (optional)	307
Schlüsseldatenbank mit gskkyman erstellen (optional).	308
Keystore mit keytool erstellen (optional)	311

Anhang B. TCP/IP konfigurieren 313

Abhängigkeit vom Hostnamen	313
Wissenswertes zu Resolvern	314
Wissenswertes zur Suchreihenfolge für Konfigurationsdaten	314
Suchreihenfolgen in der z/OS UNIX-Umgebung	315
Basiskonfigurationsdateien des Resolvers	315
Umsetztabelle	316
Lokale Hosttabellen	316
Diese Konfigurationsinformationen in Developer for System z anwenden	317
Nicht ordnungsgemäß aufgelöste Hostadresse	320

Anhang C. INETD konfigurieren 321

inetd.conf	321
ETC.SERVICES	322
Suchreihenfolge in der z/OS UNIX-Umgebung	323
Suchreihenfolge in der nativen MVS-Umgebung	323
Portdefinitionen in PROFILE.TCPIP	324
/etc/inetd.pid	325
Start	325
/etc/rc	325
/etc/inittab	325
BPXBATCH	326
Shellsitzung	326
Sicherheit.	327

Anforderungen von Developer for System z	328
INETD	328
REXEC (oder SSH)	328

Anhang D. APPC konfigurieren 329

VSAM.	329
VTAM.	330
SYS1.PARMLIB(APPCCPMxx)	331
SYS1.PARMLIB(ASCHPMxx)	332
APPC-Änderungen aktivieren	333
Transaktion für TSO Commands Service definieren	333
Alternative Konfigurationsoptionen (optional)	334
Alternativer Transaktionsname	334
Mehrere LUs	334
LU-Sicherheit	334

Anhang E. Voraussetzungen 337

Hostvoraussetzungen für z/OS	337
z/OS	337
SMP/E	339
SDK für z/OS Java 2 Technology Edition	339
Zusätzliche Hostvoraussetzungen für z/OS	340
z/OS	340
COBOL-Compiler	342
PL/I-Compiler	342
Debug Tool for z/OS.	343
CICS Transaction Server.	344
IMS	344
DB2 for z/OS	345
Rational Team Concert for System z.	345
File Manager	346
Fault Analyzer	346
REXX	346
Ported Tools.	347
Ant.	347
Endevor	347

Literaturübersicht 349

Referenzierte Veröffentlichungen	349
Veröffentlichungen mit weiteren Informationen	351

Glossar 353

Dokumentationshinweise für IBM Rational Developer for System z 357

Copyrightlizenz	358
Marken	359

Index 361

Abbildungsverzeichnis

1.	JMON - Gestartete Task von JES Job Monitor	22
2.	RSED - Gestartete Task für den RSE-Dämon	23
3.	LOCKD - Gestartete Task für den Sperrdämon	24
4.	RSED - Alternativer Start des RSE-Dämons	25
5.	rsed.stdin.sh - Alternativer Start des RSE-Dämons.	25
6.	FEJJCNFG (Konfigurationsdatei für JES Job Monitor)	29
7.	rsed.envvars - RSE-Konfigurationsdatei	34
8.	(Fortsetzung)	35
9.	ISPF-Konfigurationsdatei ISPF.conf.	48
10.	CARMA-Konfigurationsdatei CRASRV.properties	54
11.	CRASRV.properties - CARMA-Start mit Batchübergabe	56
12.	CRASUBMT - CARMA-Start mit Batchübergabe.	57
13.	CRASRV.properties - *CRASTART für alternativen CARMA-Start	58
14.	crastart.conf - *CRASTART für alternativen CARMA-Start.	59
15.	CRASRV.properties - *ISPF für alternativen CARMA-Start	61
16.	ISPF.conf - *ISPF für alternativen CARMA-Start	62
17.	Abbildung x1. CRASRV.properties - CA Endeavor SCM-RAM-Start mit Batchübergabe	66
18.	Abbildung x2. CRASUBMT - CA Endeavor SCM-RAM-Start mit Batchübergabe	67
19.	Abbildung x3. CRASRV.properties - CA Endeavor SCM-RAM-Start mit CRASTART	68
20.	crastart.conf - CA Endeavor SCM-RAM-Start mit CRASTART	69
21.	Aktualisierung von ISPF.conf für SCLMDT	83
22.	Aktualisierung von rsed.envvars für SCLMDT	84
23.	FLM02LST - Konfigurations-JCL für Umsetzung langer/kurzer Namen	85
24.	ELAXMSAM - Task für gespeicherte DB2-Prozedur	90
25.	ELAXMJCL - Definition einer gespeicherten DB2-Prozedur.	91
26.	SSL-Konfigurationsdatei ssl.properties.	95
27.	Konfigurationsdatei für Protokollierung rse-comm.properties	97
28.	propertiescfg.properties - Konfigurationsdatei für hostbasierte Eigenschaftsgruppen	99
29.	projectcfg.properties - Konfigurationsdatei für hostbasierte Projekte	100
30.	File Manager-Konfigurationsdatei FMIEXT-properties	101
31.	uchars.settings - Konfigurationsdatei für nicht editierbaren Zeichen	102
32.	REXX für APPC-ISPF-Anzeigen	105
33.	Bedienerbefehl 'START JMON'	125
34.	Bedienerbefehl 'START RSED'	126
35.	Bedienerbefehl 'START LOCKD'	126
36.	Bedienerbefehl 'MODIFY JMON'	127
37.	Bedienerbefehl 'MODIFY RSED'	128
38.	Bedienerbefehl 'MODIFY LOCKD'	131
39.	Bedienerbefehl 'STOP'.	132
40.	TCP/IP-Ports	162
41.	Komponentenübersicht	193
42.	RSE als Java-Anwendung	195
43.	Taskeigner	196
44.	Verbindungsflow	198
45.	Sperrdämonflow	200
46.	z/OS UNIX-Verzeichnisstruktur	202
47.	WLM-Klassifikation	205
48.	Maximale Anzahl von Adressräumen	219
49.	Anzahl der Adressräume pro Client	219
50.	Maximale Anzahl von Prozessen	221
51.	Anzahl von Prozessen pro Client	222
52.	Maximale Anzahl von Threads in einem RSE-Thread-Pool	225
53.	Maximale Anzahl von Threads in einem RSE-Thread-Pool	225
54.	Ressourcennutzung mit 5 Anmeldungen	229
55.	Ressourcennutzung mit 5 Anmeldungen (Fortsetzung).	230
56.	Ressourcennutzung beim Bearbeiten eines Members der untergliederten Datei	231
57.	Speicherbelegung im z/OS UNIX-Dateisystem	233
58.	Ressourcennutzung der Beispielkonfiguration	247
59.	ADNJSPAU - CICSTS-Verwaltungsdienstprogramm.	263
60.	FEKAPPCC - Erstellen einer zweiten APPC-Transaktion	275
61.	RSEDSSL - RSE-Dämonbenutzerjob für SSL	304
62.	Dialog 'Hostzertifikat importieren'	305
63.	Vorgabendialog - SSL	306
64.	Start-JCL für INETD	326
65.	JCL zur Erstellung von APPC-VSAM	330
66.	SYS1.SAMPLIB(ATBAPPL)	331
67.	SYS1.PARMLIB(APPCPMxx)	331
68.	SYS1.PARMLIB(ASCHPMxx)	333

Tabellen

1. Erforderliche Ressourcen	6	28. Bedienerbefehle von JES3 Job Monitor	186
2. Optionale Ressourcen	6	29. WLM-Eingangspunktsysteme	206
3. Administratoren für erforderliche Tasks	7	30. WLM-Qualifikationsmerkmale für Arbeitsvor- gänge	207
4. Administratoren für optionale Tasks.	8	31. WLM-Verarbeitungsprozesse	208
5. Clientprüfliste - obligatorischer Teil	13	32. WLM-Verarbeitungsprozesse - STC	209
6. Clientprüfliste - optionaler Teil	14	33. WLM-Verarbeitungsprozesse - OMVS	210
7. ELAXF*-Beispielprozeduren	26	34. WLM-Verarbeitungsprozesse - JES	211
8. Prüfliste der übergeordneten Qualifikations- merkmale in ELAXF*	27	35. WLM-Verarbeitungsprozesse - ASCH	212
9. Matrix der Befehlsberechtigungen für LIMIT- _COMMANDS	32	36. WLM-Verarbeitungsprozesse - CICS	213
10. Variablen für crastart.conf.	59	37. Allgemeine Ressourcennutzung	216
11. Standard-Transaktions-IDs des CRD-Servers	76	38. Benutzerspezifische vorausgesetzte Ressour- cennutzung	216
12. Standard-Transaktions-IDs des CRD-Servers	77	39. Benutzerspezifische Ressourcennutzung	217
13. Prüfliste für den SCLM-Administrator	87	40. Anzahl der Adressräume.	217
14. Mechanismen für den SSL-Zertifikatsspeicher	94	41. Begrenzungen für Adressräume	220
15. Gültige Keystoretypen	96	42. Anzahl der Prozesse	220
16. Prüfliste für APPC-Transaktionen	105	43. Begrenzungen für Prozesse	223
17. Installationsprüfprogramme für Services	112	44. Anzahl der Threads	223
18. Fehlerstatus des Thread-Pools	129	45. Begrenzungen für Threads	226
19. RSE-Konsolnachrichten	132	46. Anweisungen für die Protokollausgabe	234
20. Variablen für JAVA_DUMP_TDUMP_PAT- TERN	145	47. Anpassungen bei Version 7.6	288
21. JES Job Monitor, Konsolbefehle	167	48. Anpassungen bei Version 7.5	293
22. Matrix der Befehlsberechtigungen für LIMIT- _COMMANDS	167	49. Anpassungen bei Version 7.1	297
23. Erweiterte JESSPOOL-Profile	167	50. Mechanismen für den SSL-Zertifikatsspeicher	300
24. Berechtigungsmatrix zum Durchsuchen für LIMIT_VIEW	170	51. Für den Resolver verfügbare lokale Definitio- nen	319
25. Mechanismen für den SSL-Zertifikatsspeicher	171	52. Referenzierte Veröffentlichungen	349
26. Variablen für die Sicherheitskonfiguration	179	53. Referenzierte Websites	351
27. Bedienerbefehle von JES2 Job Monitor	186	54. Veröffentlichungen mit weiteren Informatio- nen	351

Zu diesem Handbuch

Dieses Handbuch beschäftigt sich mit der Konfiguration der Funktionen von IBM Rational Developer for System z. Es enthält Konfigurationsanweisungen für IBM Rational Developer for System z Version 7.6.1 auf Ihrem z/OS-Hostsystem.

Im weiteren Verlauf dieses Handbuchs werden die folgenden Namen verwendet:

- *IBM Rational Developer for System z* wird als *Developer for System z* bezeichnet.
- *Common Access Repository Manager* wird mit *CARMA* abgekürzt.
- *Software Configuration and Library Manager Developer Toolkit* wird als *SCLM Developer Toolkit* bezeichnet und mit *SCLMDT* abgekürzt.
- *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.

Die Konfigurationsdaten für frühere Versionen, einschließlich IBM WebSphere Developer for System z, IBM WebSphere Developer für zSeries und IBM® WebSphere Studio Enterprise Developer, sind in den Veröffentlichungen 'Hostkonfiguration' und 'Program Directory' der entsprechenden Releases enthalten.

Zielgruppe

Dieses Handbuch wendet sich an Systemprogrammierer, die IBM Rational Developer for System z Version 7.6.1, FMID HHOP760 auf ihrem z/OS-Hostsystem installieren und konfigurieren möchten.

Im vorliegenden Handbuch sind detailliert die verschiedenen Schritte für eine vollständige Konfiguration des Produkts sowie einige vom Standard abweichende Szenarien beschrieben. 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 das Handbuch *Rational Developer for System z Version 7.6.1 Hostkonfiguration* (IBM Form SC12-4062-04) zusammengefasst (aktualisiert im Mai 2010).

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

Dieses Dokument enthält Informationen, die bisher im Handbuch *Rational Developer for System z Version 7.6 Hostkonfiguration* (IBM Form SC12-4062-03) enthalten waren.

Neue Informationen:

- Korrekturen und zusätzliche Informationen, die in den Releaseinformationen zu *Rational Developer for System z Version 7.6 Hostkonfiguration* angegeben waren, sind eingearbeitet.
- Migrationshinweise für Version 7.6.1. Lesen Sie hierzu „Migrationshinweise für Version 7.6.1“ auf Seite 285.

- Dokumentenübersicht hinzugefügt. Lesen Sie hierzu „Beschreibung der Dokumentinhalte“.
- Unterstützung der 64-Bit-Java™-Version. Lesen Sie hierzu Anhang E, „Voraussetzungen“, auf Seite 337.
- Produktkonfiguration über ISPF-Anzeigen. Lesen Sie hierzu „Hinweise zur Konfigurationsvorbereitung“ auf Seite 9.
- Neue Anweisungen in `rsed.envvars`. Lesen Sie hierzu „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 33.
- Neue PROCLIB-Member. Lesen Sie hierzu „ELAXF*-Prozeduren für ferne Builderstellung“ auf Seite 26.
- Neues Layout für CARMA-VSAM. Lesen Sie hierzu „Migrationshinweise für CARMA-VSAM“ auf Seite 53.
- Unterstützung mehrerer CARMA-RAMs. Lesen Sie hierzu „(Optional) Unterstützung mehrerer RAMs“ auf Seite 70.
- Neue Optionen in Application Deployment Manager. Lesen Sie hierzu „Verwaltungsdienstprogramm“ auf Seite 261.
- Neues Layout für Application Deployment Manager-VSAM. Lesen Sie hierzu „Migrationshinweise zum Verwaltungsdienstprogramm“ auf Seite 266.
- Neue Bedienerbefehle. Lesen Sie hierzu Kapitel 8, „Bedienerbefehle“, auf Seite 125.
- Neue Konsolnachrichten. Lesen Sie hierzu „Konsolnachrichten“ auf Seite 132.
- Workload Management-Informationen. Lesen Sie hierzu Kapitel 12, „Hinweise zu WLM“, auf Seite 205.

Beschreibung der Dokumentinhalte

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

Planung

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

Basisanpassung

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

- Anpassungskonfiguration
- PARMLIB-Änderungen
- PROCLIB-Änderungen
- Sicherheitsdefinitionen
- FEJJCNFG (Konfigurationsdatei für JES Job Monitor)
- `rsed.envvars`, RSE-Konfigurationsdatei
- Konfigurationsdatei "ISPF.conf" des TSO/ISPF-Client-Gateways von ISPF

Common Access Repository Manager (optional)

Common Access Repository Manager (CARMA) ist eine Produktivitätshilfe für Entwickler, die RAM (Repository Access Managers) erstellen. Ein RAM ist eine Anwendungsprogrammierschnittstelle (API) für z/OS-basierte SCMs (Software Configuration Managers).

Vom Benutzer geschriebene Anwendungen können einen CARMA-Server starten, der die RAM(s) lädt und eine Standardschnittstelle für den Zugriff auf den SCM bereitstellt.

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

Application Deployment Manager (optional)

Developer for System z verwendet bestimmte Funktionen von Application Deployment Manager als allgemeine Deploymentmethode für verschiedene Komponenten. Durch eine optionale Anpassung können weitere Funktionen von Application Deployment Manager aktiviert und der folgende Service zu Developer for System z hinzugefügt werden:

- Der IBM CICS-Explorer stellt eine Eclipse-basierte Infrastruktur für die Anzeige und Verwaltung von CICS-Ressourcen bereit und verbessert die Integration der verschiedenen CICS-Tools.
- Der CICS Resource Definition-Client (CRD-Client) und der CRD-Server stellen folgende Funktionen bereit:
 - CICS Resource Definition-Editor
 - Ermöglicht Anwendungsentwicklern, CICS-Ressourcen auf begrenzte, gesteuerte und sichere Weise zu definieren
 - Hindert die CICS-Entwicklung am Zugriff auf unautorisierte oder falsche VSAM-Dateien, indem dem CICS-Administrator die Kontrolle über das Attribut für den physischen Dateinamen in den Dateideinitionen gegeben wird.
 - Sonstige Unterstützung für die CICS-Entwicklung
 - Sonstige Unterstützung für die Entwicklung von CICS-Web-Services

SCLM Developer Toolkit (optional)

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

SCLM Developer Toolkit enthält ein auf Eclipse basierendes Plug-in mit Anbindung an SCLM, das den Zugriff auf alle SCLM-Prozesse für die herkömmliche Codeentwicklung ermöglicht. Das Plug-in unterstützt auch die vollständige Java- und J2EE-Entwicklung auf der Workstation. Dazu gehören die Synchronisation mit SCLM auf dem Mainframe-Computer sowie die Builderstellung, die Assemblierung und das Deployment des J2EE-Codes vom Mainframe-Computer.

Weitere Anpassungstasks (optional)

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

- DB2, gespeicherte Prozedur
- Enterprise Service Tools-Unterstützung (EST)
- CICS, Unterstützung bidirektionaler Sprachen
- IRZ-Diagnosefehlnachrichten
- RSE-SSL-Verschlüsselung
- RSE, Traceerstellung
- Hostbasierte Eigenschaftsgruppen

- Hostbasierte Projekte
- File Manager-Integration
- Nicht bearbeitbare Zeichen
- REXEC (oder SSH) verwenden
- APPC-Transaktion für TSO Commands Service
- WORKAREA-Bereinigung

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.

Bedienerbefehle

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

Konfigurationsprobleme lösen

Dieses Kapitel soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von Developer for System z auftreten könnten. Es enthält die folgenden Abschnitte:

- Protokoll- und Konfigurationsanalyse mit FEKLOGS
- Protokolldateien
- Speicherauszugsdateien
- Traceerstellung
- z/OS UNIX-Berechtigungsbits
- Reservierte TCP/IP-Ports
- Adressraum, Größe
- APPC-Transaktion und TSO Commands Service
- Sonstige Informationen

Sicherheitsaspekte

Developer for System z ermöglicht Benutzern einer Workstation den Zugriff auf Mainframe-Computer, wenn diese selbst kein Mainframe-Computer ist. Wichtige Aspekte bei der Produktkonfiguration sind deshalb das Prüfen von Verbindungsanforderungen, das Bereitstellen von sicherer Kommunikation zwischen dem Host und der Workstation sowie das Autorisieren und Protokollieren der Aktivitäten.

Wissenswertes zu Developer for System z

Developer for System z umfasst mehrere interagierende Komponenten, damit der Client auf die Host-Services und -daten zugreifen kann. Wenn Sie das Design dieser Komponenten verstehen, können Sie die richtigen Konfigurationsentscheidungen treffen.

Hinweise zu WLM

Im Gegensatz zu herkömmlichen z/OS-Anwendungen ist Developer for System z keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for System z umfasst mehrere interagierende Komponenten, damit der Client auf die Host-Services und -daten zugreifen kann. Einige dieser Services sind in verschiedenen Adressräumen aktiv und werden somit verschiedenen WLM-Klassifikationen zugeordnet.

Optimierungsaspekte

RSE (Remote Systems Explorer) ist ein zentraler Bestandteil von Developer for System z. RSE besteht aus einem Dämonadressbereich, der Thread-Pooling und Adressräume steuert, um die Verbindungen und die Arbeitslast der Clients zu verwalten. Der Dämon wird als Sammelpunkt für Verbindungs- und Verwaltungszwecke eingesetzt, während die Thread-Pools die Clientarbeitslast verarbeiten.

Dadurch wird RSE das Hauptziel für die Optimierung der Installation von Developer for System z. Wenn Sie allerdings Hunderte von Benutzern verwalten, die jeweils mindestens 16 Threads, eine bestimmte Speichermenge und mindestens einen Adressraum verwenden, müssen Developer for System z und z/OS richtig konfiguriert sein.

Leistungsaspekte

z/OS ist ein sehr anpassungsfähiges Betriebssystem, bei dem (manchmal kleine) Systemänderungen eine enorme Auswirkung auf die Gesamtleistung haben können. Dieses Kapitel hebt einige der Änderungen hervor, die zu einer Verbesserung der Leistung von Developer for System z führen können.

CICSTS-Aspekte

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

Anpassung der TSO-Umgebung

Dieses Kapitel soll Sie beim Imitieren einer TSO-Anmeldeprozedur durch das Hinzufügen von DD-Anweisungen und Dateien zur TSO-Umgebung in Developer for System z unterstützen.

Ausführung mehrerer Instanzen

In bestimmten Situationen, z. B. beim Testen eines Upgrades, kann die Ausführung mehrerer aktiver Instanzen von Developer for System z auf demselben System erwünscht sein. Manche Ressourcen können jedoch nicht gemeinsam genutzt werden, z. B. TCP/IP-Ports, sodass die Standardeinstellungen nicht immer anwendbar sind. Anhand der Informationen in diesem Kapitel können Sie die Koexistenz verschiedener Instanzen von Developer for System z planen, um sie dann gestützt auf dieses Konfigurationshandbuch anzupassen.

Leitfaden für die Migration

Dieser Abschnitt hebt die Installations- und Konfigurationsänderungen im Vergleich zu früheren Produktreleases hervor. Darüber hinaus finden Sie hier allgemeine Richtlinien für die Migration auf dieses Release. Weitere Informationen hierzu finden Sie in den entsprechenden Abschnitten dieses Handbuchs.

SSL- und X.509-Authentifizierung konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von SSL (Secure Sockets Layer) oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten. Dieser Anhang stellt auch eine Beispielkonfiguration zur Verfügung, um Benutzer zu unterstützen, die sich mit einem X.509-Zertifikat selbst authentifizieren.

TCP/IP konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von TCP/IP oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten.

INETD konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von INETD oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten. INETD wird von Developer for System z zur REXEC/SSH-Funktionalität verwendet.

APPC konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von APPC (Advanced Program-to-Program Communication) oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten.

Voraussetzungen

In diesem Anhang werden die Hostvoraussetzungen und die zusätzlich erforderliche Software für diese Version von Developer for System z aufgelistet.

Teil 1. Anpassung für Developer for System z

Kapitel 1. Planung

Nutzen Sie für die Planung der Installation und des Deployment von Developer for System z die Informationen in diesem Kapitel und in Anhang E, „Voraussetzungen“, auf Seite 337. Die folgenden Themen werden beschrieben:

- „Hinweise zur Migration“
- „Hinweise zur Planung“
- „Hinweise zu den Installationsvorbereitungen“ auf Seite 4
- „Hinweise zur Konfigurationsvorbereitung“ auf Seite 9
- „Hinweise zu den Deploymentvorbereitungen“ auf Seite 12
- „Prüfliste für den Client“ auf Seite 13

Hinweise zur Migration

Kapitel 18, „Leitfaden für die Migration“, auf Seite 283 beschreibt die Installations- und Konfigurationsänderungen im Vergleich zu früheren Produktreleases. Nutzen Sie diese Informationen für die Planung Ihrer Migration auf das aktuelle Release von Developer for System z.

Anmerkungen:

1. Wenn Sie mit einer früheren Version von IBM Rational Developer for System z, IBM WebSphere Developer for System z, IBM WebSphere Developer für zSeries oder IBM WebSphere Studio Enterprise Developer arbeiten, sollten Sie die zugehörigen Anpassungsdateien speichern, BEVOR Sie IBM Rational Developer for System z Version 7.6.1 installieren. In Kapitel 18, „Leitfaden für die Migration“, auf Seite 283 finden Sie einen Überblick über die Dateien, die eine Anpassung erforderten.
2. Falls Sie planen, mehrere Instanzen von Developer for System z auszuführen, lesen Sie Kapitel 17, „Mehrere Instanzen ausführen“, auf Seite 277.

Hinweise zur Planung

Produktübersicht

Developer for System z besteht aus einem Client, der auf dem Personal Computer des Benutzers installiert ist, und einem Server, der auf mindestens einem Host installiert ist. Der Host ist in dieser Dokumentation ein z/OS-System. Andere Betriebssysteme, wie AIX und Linux® auf System z werden aber auch unterstützt.

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

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

Developer for System z kann auch mit Subsystemen und anderer Anwendungssoftware auf dem Host interagieren, wie CICS, Debug Tool und Software Configuration Managers (SCMs), wenn Developer for System z entsprechend konfiguriert ist und wenn diese (zusätzlich erforderlichen) Produkte verfügbar sind.

In Kapitel 11, „Wissenswertes zu Developer for System z“, auf Seite 193 finden Sie Informationen zum grundsätzlichen Verständnis des Designs von Developer for System z.

Weitere Informationen zur Funktionalität, die in Developer for System z enthalten ist, finden Sie auf der Website für Developer for System z: <http://www-01.ibm.com/software/awdtools/rdz/>. Sie können sich auch an Ihren IBM Ansprechpartner wenden.

Erforderliche Qualifikationen

Für eine Hostinstallation von Developer for System z ist minimales Know-how für SMP/E for z/OS erforderlich.

Für die Konfiguration von Developer for System z ist mehr als die typischen Berechtigungen für Systemprogrammierer und deren übliches Fachwissen erforderlich. Es ist zu erwarten, dass Unterstützung von anderen notwendig sein wird. Die jeweiligen Administratoren für die erforderlichen und optionalen Anpassungstasks finden Sie in Tabelle 3 auf Seite 7 und in Tabelle 4 auf Seite 8 aufgelistet.

Zeitbedarf

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

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

Die Erfahrung hat gezeigt, dass der Installations- und Konfigurationsprozess für den Host von Developer for System z insgesamt einen Tag bis vier Tage dauert. 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 entsprechend länger.

Hinweise zu den Installationsvorbereitungen

Ausführliche Anweisungen zur SMP/E-Installation des Produkts enthält die Veröffentlichung *Program Directory for IBM Rational Developer for System z* (IBM Form GI11-8298).

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

Falls Sie planen, mehrere Instanzen von Developer for System z auszuführen, lesen Sie Kapitel 17, „Mehrere Instanzen ausführen“, auf Seite 277.

Konfigurationsoptionen

Developer for System z bietet Optionen für den Zugriff auf TSO Commands Service an. Sie müssen eine der nachfolgend aufgelisteten Methoden auswählen und konfigurieren:

- TSO/ISPF-Client-Gateway-Service von ISPF mit einem erforderlichen ISPF-Mindestservicelevel. Dies ist die in den bereitgestellten Beispielen verwendete Standardmethode.
- Eine APPC-Transaktion (wie in den Vorgängerreleases von Version 7.1)

Anmerkung: Das TSO/ISPF-Client-Gateway von ISPF wird auch von SCLM Developer Toolkit verwendet und kann darüber hinaus von einer alternativen Startmethode für CARMA (Common Access Repository Manager) genutzt werden.

Vorausgesetzte Produkte

Anhang E, „Voraussetzungen“, auf Seite 337 enthält eine Liste vorausgesetzter Software, die installiert und betriebsbereit sein muss, damit Developer for System z funktioniert. Außerdem gibt es eine Liste zusätzlich erforderlicher Software zur Unterstützung bestimmter Features von Developer for System z. Zur Laufzeit muss diese zusätzlich erforderliche Software installiert und betriebsbereit sein, damit das entsprechende Feature ordnungsgemäß funktionieren kann.

Eine aktuelle Liste der Produkte, die für Ihre Version von Developer for System z vorausgesetzt werden bzw. zusätzlich erforderlich sind, enthält die Veröffentlichung *Rational Developer for System z Prerequisites* (IBM Form SC23-7659) in der Onlinebibliothek für Developer for System z unter <http://www-01.ibm.com/software/awdtools/rdz/library/>. 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 Basiskonfiguration aufgeführt:

- z/OS ab Version 1.8
- ISPF-APAR OA29489 (TSO/ISPF-Client-Gateway)
- Java ab Version 5.0

Anmerkung: Bei Verwendung einer 64-Bit-Version von Java muss die vorläufige Programmkorrektur für Developer for System z "APAR PM07305" angewendet werden. Die vorläufige Programmkorrektur ist über die Seite für empfohlene Services für Developer for System z" verfügbar: <http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335>.

Erforderliche Ressourcen

Developer for System z erfordert die Reservierung der in Tabelle 1 auf Seite 6 aufgelisteten Systemressourcen. Die in Tabelle 2 auf Seite 6 aufgelisteten Ressourcen sind für Zusatzservices erforderlich. Planen Sie vorausschauend, damit diese Ressourcen rechtzeitig verfügbar sind. Die Bereitstellung kann je nach Geschäftspolitik an Ihrem Standort einige Zeit in Anspruch nehmen.

Anmerkung: Developer for System z umfasst mehrere Tasks, die miteinander und mit dem Client kommunizieren. Diese Tasks verwenden verschiedene Zeitgeber, um Kommunikationsunterbrechungen mit ihren Partnern festzustellen. Dies bedeutet, dass auf Systemen mit hoher CPU-Belastung oder mit falschen WLM-Einstellungen (Auslastungsverwaltung) für Developer for System z Zeitlimitprobleme auftreten können (weil während des Zeitlimitfensters nicht genug CPU-Zeit verfügbar war).

Tabelle 1. Erforderliche Ressourcen

Ressource	Standardwert	Informationen
Datei mit APF-Berechtigung	FEK.SFEKAUTH	„APF-Berechtigungen in PROGxx“ auf Seite 18
Gestartete Task	JMON, RSED und LOCKD	„Hinweise zum Server“ auf Seite 10
Port für die hostinterne Verwendung	6715	„Konfigurationsdatei für JES Job Monitor (FEJJC�FG)“ auf Seite 29
Port für die hostinterne Verwendung	4036	„RSE-Konfigurationsdatei rsed.envvars“ auf Seite 33
Port für die Kommunikation zwischen Client und Host	4035	„PROCLIB-Änderungen“ auf Seite 21
Portbereich für die Kommunikation zwischen Client und Host	Jeder verfügbare Port kann verwendet werden.	„Für RSE-Server verfügbaren PORTRANGE definieren“ auf Seite 41
Definition der Anwendungssicherheit	Uneingeschränkter Lesezugriff auf FEKAPPL	„Anwendungsschutz für RSE definieren“ auf Seite 188
PassTicket-Sicherheitsdefinitionen	Kein Standard	„PassTicket-Unterstützung für RSE definieren“ auf Seite 188

Tabelle 2. Optionale Ressourcen

Ressource	Standardwert	Informationen
LINKLIST-Datei	FEK.SFEKAUTH und FEK.SFEKLOAD	Kapitel 5, „SCLM Developer Toolkit (optional)“, auf Seite 81
LPA-Datei	FEK.SFEKLPA	Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 51
Portbereich für die hostinterne Verwendung	5227-5326 (100 Ports)	Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 51
Ports für die hostinterne Verwendung	Jeder verfügbare Port kann verwendet werden.	„APPC-Transaktion für TSO Commands Service (optional)“ auf Seite 104
Port für die Kommunikation zwischen Client und Host	Kein Standard	Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73
Aktualisierung für die CICS-Systemdefinition	Mehrere Werte	Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73

Tabelle 2. Optionale Ressourcen (Forts.)

Ressource	Standardwert	Informationen
Aktualisierung für CICS-JCL	FEK.SFEKLOAD	<ul style="list-style-type: none"> • Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73 • „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 92

Für die Konfiguration von Developer for System z ist mehr als die typischen Berechtigungen für Systemprogrammierer und deren übliches Fachwissen erforderlich. Es ist zu erwarten, dass ein gewisses Maß an Unterstützung von anderen notwendig sein wird. Die jeweiligen Administratoren für die erforderlichen und optionalen Anpassungstasks finden Sie in Tabelle 3 und in Tabelle 4 auf Seite 8 aufgelistet.

Tabelle 3. Administratoren für erforderliche Tasks

Administrator	Task	Informationen
Systemadministrator	Für alle Anpassungstasks sind typische Systemprogrammiereraktionen erforderlich.	Nicht zutreffend
Sicherheitsadministrator	<ul style="list-style-type: none"> • OMVS-Segment für Benutzer von Developer for System z definieren • Dateiprofile definieren • Gestartete Tasks definieren • Befehlssicherheit für Bediener definieren • z/OS UNIX-Serverprofile definieren • Anwendungssicherheit definieren • PassTicket-Unterstützung definieren • Programmgesteuerte Dateien definieren • Programmgesteuerte z/OS UNIX-Dateien definieren 	Kapitel 10, „Sicherheitsaspekte“, auf Seite 159
TCP/IP-Administrator	Neue TCP/IP-Ports definieren	„TCP/IP-Ports“ auf Seite 162
WLM-Administrator	Ziele für gestartete Tasks für die Server und deren untergeordneten Prozesse zuordnen	Kapitel 12, „Hinweise zu WLM“, auf Seite 205

Tabelle 4. Administratoren für optionale Tasks

Administrator	Task	Informationen
Systemadministrator	Für alle Anpassungstasks sind typische Systemprogrammiereraktionen erforderlich.	Nicht zutreffend
Sicherheitsadministrator	<ul style="list-style-type: none"> • Dateiprofile definieren • Programmgesteuerte Dateien definieren • Berechtigung für die Übergabe von xxx*-Jobs definieren • CICS-Transaktions-sicherheit definieren • Zertifikat für SSL hinzufügen • Unterstützung des X.509-Clientzertifikats definieren 	<ul style="list-style-type: none"> • Kapitel 10, „Sicherheitsaspekte“, auf Seite 159 • Kapitel 15, „CICSTS-Aspekte“, auf Seite 257 • Anhang A, „SSL- und X.509-Authentifizierung konfigurieren“, auf Seite 299
TCP/IP-Administrator	Neue TCP/IP-Ports definieren	„TCP/IP-Ports“ auf Seite 162
SCLM-Administrator	<ul style="list-style-type: none"> • SCLM-Sprachumsetzer für JAVA/J2EE-Unterstützung definieren • SCLM-Typen für JAVA/J2EE-Unterstützung definieren 	Kapitel 5, „SCLM Developer Toolkit (optional)“, auf Seite 81
CICS-TS-Administrator	<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 	<ul style="list-style-type: none"> • Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73 • „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 92
DB2	Gespeicherte DB2-Prozedur definieren	„Gespeicherte DB2-Prozedur (optional)“ auf Seite 89
WLM-Administrator	<ul style="list-style-type: none"> • Einer gespeicherten DB2-Prozedur Ziele zuweisen • Einer APPC-Transaktion TSO-Ziele zuweisen 	<ul style="list-style-type: none"> • „Gespeicherte DB2-Prozedur (optional)“ auf Seite 89 • Kapitel 12, „Hinweise zu WLM“, auf Seite 205
APPC-Administrator	APPC-Transaktion definieren	„APPC-Transaktion für TSO Commands Service (optional)“ auf Seite 104

Hinweise zur Konfigurationsvorbereitung

Auslastungsverwaltung

Im Gegensatz zu herkömmlichen z/OS-Anwendungen ist Developer for System z keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for System z umfasst mehrere interagierende Komponenten, damit der Client auf die Host-Services und -daten zugreifen kann. Kapitel 12, „Hinweise zu WLM“, auf Seite 205 enthält Informationen, wie Sie Ihre WLM-Konfiguration entsprechend planen können.

Ressourcennutzung und Systembegrenzungen

Wenn Developer for System z aktiv ist, wird eine variable Anzahl von Systemressourcen wie Adressräume und z/OS UNIX-Prozesse und -Threads verwendet. Die Verfügbarkeit dieser Ressourcen ist durch verschiedene Systemdefinitionen begrenzt. Kapitel 13, „Optimierungsaspekte“, auf Seite 215 enthält Informationen zum Abschätzen der Verwendung von wichtigen Ressourcen. Auf diese Weise können Sie Ihre Systemkonfiguration entsprechend planen.

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:

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

Hinweise zur Benutzer-ID

Die Benutzer-ID eines Benutzers von Developer for System z 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
FILEPROCMA= NONE
PROCUSERMA= NONE
THREDSMAX= NONE
MMAPAREAMA= NONE
```

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

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

GROUP group

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

- Zugriffsrechte READ und EXECUTE für das Installations- und Konfigurationsverzeichnis sowie die Installations- und Konfigurationsdateien von Developer for System z (standardmäßig /usr/lpp/rdz/*, /etc/rdz/* und /var/rdz/*)
- Zugriffsrechte READ, WRITE und EXECUTE für das Verzeichnis WORKAREA von Developer for System z (standardmäßig /var/rdz/WORKAREA)
- Zugriffsrecht READ für die Installationsdateien von Developer for System z (standardmäßig FEK.SFEK*)

Hinweise zum Server

Developer for System z umfasst drei permanent aktive Server, die gestartete Tasks oder Benutzerjobs sein können. Diese Server stellen selbst die erforderlichen Services bereit oder starten dafür andere Server (z. B. z/OS UNIX-Threads oder -Benutzerjobs).

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

JMON (JES Job Monitor) stellt alle Services mit Bezug zum JES bereit.

- Die von JES Job Monitor verwendeten Sicherheitsmechanismen sind nur dann wirksam, wenn die zugrunde liegenden Dateien geschützt sind. Dies impliziert, dass die Bibliotheken und Konfigurationsdateien nur von vertrauenswürdigen Systemadministratoren aktualisiert werden können.

Remote Systems Explorer (RSE) ist die Komponente von Developer for System z, die Kernservices wie die Verbindung vom Client zum Host bereitstellt.

- Ab Version 7.5 ist der RSE-Dämon eine gestartete Task und nicht länger ein von INETD verwalteter Prozess.
- Ab Version 7.5 funktioniert der RSE-Server nach einem Einzelservermodell. In früheren Versionen gab es dagegen für jede Client-Host-Verbindung einen privaten RSE-Server.
- Verschiedene, von RSE unterstützte Ebenen der Kommunikationssicherheit:
 - Die externe Kommunikation (Client-Host) kann auf bestimmte Ports beschränkt werden. Dieses Feature ist standardmäßig inaktiviert.
 - Die externe Kommunikation (Client-Host) kann mit SSL verschlüsselt werden. Dieses Feature ist standardmäßig inaktiviert.
 - Durch die Prüfung des Eingangsports kann erreicht werden, dass nur anerkannten TCP/IP-Adressen der Zugriff gewährt wird. Dieses Feature ist standardmäßig inaktiviert.
- RSE unterstützt mehrere Clientauthentifizierungsmethoden:
 - Benutzer-ID und Kennwort
 - Benutzer-ID und Kennwort für einmaliges Anmelden
 - X.509-Zertifikat
- Die von RSE verwendeten Sicherheitsmechanismen sind nur wirksam, wenn das zugrunde liegende Dateisystem geschützt ist. Dies impliziert, dass die Bibliotheken und Konfigurationsdateien nur von vertrauenswürdigen Systemadministratoren aktualisiert werden können.

Zu bestimmten Hostservices (und somit zu den zugehörigen Ports) muss der Client, wie im Abschnitt „TCP/IP-Ports“ auf Seite 162 beschrieben, eine Verbindung herstellen können. Diese Services und Ports müssen deshalb für die Firewall, die den Host schützt, definiert sein. An allen anderen von Developer for System z verwendeten Ports gibt es nur Hostdatenverkehr. Nachfolgend sind die Ports aufgelistet, die für eine Basiskonfiguration von Developer for System z notwendig sind.

- RSE-Dämon für die Einrichtung der Client-Host-Kommunikation (Standardport 4035) unter Verwendung des TCP-Protokolls
- RSE-Server für die Client-Host-Kommunikation unter Verwendung des TCP-Protokolls. Standardmäßig kann jeder verfügbare Port verwendet werden. Eine Einschränkung auf einen bestimmten Portbereich ist jedoch möglich.

Anmerkung: Ältere Clientversionen (bis Version 7.0) kommunizieren direkt mit JES Job Monitor (unter Verwendung des TCP-Protokolls) am Standardport 6715.

Konfigurationsmethode

Ab Version 7.6.1 stellt Developer for System z mit einer ISPF-Anzeigenanwendung eine alternative Methode zur Konfiguration der Hostseite des Produkts bereit. Damit stehen Ihnen die folgenden Methoden zur Auswahl:

- Verwendung der ISPF-Anzeigenanwendung. Auf diese Weise werden Sie durch die erforderlichen und die ausgewählten optionalen Anpassungsschritte geführt. Weitere Informationen enthält das White Paper *Host Configuration Utility*, das in der Internetbibliothek für Developer for System z unter <http://www-306.ibm.com/software/awdtools/rdz/library/> verfügbar ist.
- Verwendung des *Handbuchs für den Schnelleinstieg in die Hostkonfiguration*. Auf diese Weise werden Sie durch die erforderlichen Anpassungsschritte geführt. Der Inhalt dieses Handbuchs ist auf eine Basiskonfiguration beschränkt.
- Verwendung des *Handbuchs "Hostkonfiguration"*. Auf diese Weise werden Sie durch die erforderlichen und alle optionalen Anpassungsschritte geführt. In diesem Handbuch sind alle konfigurierbare Optionen beschrieben, einschließlich einiger vom Standard abweichender Szenarien.

Hinweise zu den Deploymentvorbereitungen

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

Für das 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 das Deployment vorausgesetzte und zusätzlich erforderliche Software.

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

Anmerkungen:

1. FEK und /usr/lpp/rdz sind das während der Produktinstallation verwendete übergeordnete Qualifikationsmerkmal (High Level Qualifier, HLQ) und der Pfad. FEK.#CUST, /etc/rdz und /var/rdz sind die während der Anpassung des Produkts verwendeten Standardpositionen. (Weitere Informationen hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.)
2. Sie sollten Developer for System z in einem privaten Dateisystem (HFS oder zFS) installieren, um das Deployment der z/OS UNIX-Produktkomponenten zu vereinfachen.
3. Wenn Sie kein privates Dateisystem verwenden können, sollten Sie für den Transport der z/OS UNIX-Verzeichnisse von einem System zu einem anderen ein Archivierungstool wie den z/OS UNIX-Befehl tar nutzen. Auf diese Weise bleiben die Attribute (z. B. für die Programmsteuerung) für die Dateien und Verzeichnisse von Developer for System z erhalten.

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

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

Prüfliste für den Client

Benutzer der Clientkomponente von Developer for System z müssen das Ergebnis bestimmter Hostanpassungen, 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 14 sind die erforderlichen Ergebnisse optionaler Anpassungsschritte aufgelistet.

Tabelle 5. Clientprüfliste - obligatorischer Teil

Anpassung	Wert
Portnummer des JES Job Monitor-Servers (Standard: 6715) Lesen Sie die Beschreibung für SERV_PORT im Abschnitt „Konfigurationsdatei für JES Job Monitor (FEJJCNFG)“ auf Seite 29.	
TCP/IP-Portnummer des RSE-Dämons (Standard: 4035) Lesen Sie hierzu den Abschnitt „RSE-Dämon“ auf Seite 23.	

Tabelle 6. Clientprüfliste - optionaler Teil

Anpassung	Wert
Position der ELAXF*-Prozeduren, falls sie nicht in einer Prozedurenbibliothek des Systems enthalten sind Lesen Sie die Anmerkung zu JCLLIB im Abschnitt „ELAXF*-Prozeduren für ferne Builderstellung“ auf Seite 26.	
Namen der ELAXF*-Prozeduren oder der zugehörigen Prozedurschritte, sofern sie geändert wurden Lesen Sie die Anmerkung zur Änderung der Namen im Abschnitt „ELAXF*-Prozeduren für ferne Builderstellung“ auf Seite 26.	
Name der gespeicherten DB2-Prozedur (Standard: ELAXMSAM) Informationen zu gespeicherten DB2-Prozeduren finden Sie in Kapitel 17, „Mehrere Instanzen ausführen“, auf Seite 277.	
Position der gespeicherten DB2-Prozedur, sofern sie nicht in einer Prozedurenbibliothek des Systems enthalten ist: Lesen Sie hierzu den Abschnitt „Gespeicherte DB2-Prozedur (optional)“ auf Seite 89.	
TN3270-Portnummer für den Host-Connect-Emulator (zusätzlich erforderliche Software) (Standard: 23) Lesen Sie hierzu Kapitel 10, „Sicherheitsaspekte“, auf Seite 159.	
REXEC- oder SSH-Portnummer (zusätzlich erforderliche Software) (Standard: 512 bzw. 22) Lesen Sie hierzu den Abschnitt „REXEC (oder SSH) verwenden (optional)“ auf Seite 102.	
Position der Datei server.zseries bei Verwendung der Verbindungsmethode mit REXEC/SSH (Standard: /etc/rdz) Lesen Sie hierzu den Abschnitt „REXEC (oder SSH) verwenden (optional)“ auf Seite 102.	
Position der CRA#ASLM-JCL für das Anlegen von CARMA-SCLM-RAM-Dateien (Standard: FEK.#CUST.JCL) Lesen Sie die Anmerkung zu CRA#ASLM im Abschnitt „SCLM-RAM aktivieren“ auf Seite 63.	

Kapitel 2. Basisanpassung

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

Voraussetzungen und Prüfliste

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

- Datei mit APF-Berechtigung
- Verschiedene Aktualisierungen von PARMLIB
- Verschiedene Aktualisierungen der Sicherheitssoftware
- Unterschiedliche TCP/IP-Ports für interne und Client-Host-Kommunikation

Bevor Sie Developer for System z an Ihrem Standort verwenden können, müssen Sie die folgenden Tasks ausführen, um die Installation zu prüfen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Erstellen Sie anpassbare Kopien der Beispiele sowie die Arbeitsumgebung für Developer for System z. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“.
2. Aktualisieren Sie die z/OS UNIX-Systembegrenzungen und starten Sie gestartete Tasks. Definieren Sie außerdem über APF autorisierte Dateien und LINKLIST-Dateien sowie optional LPA-Dateien. Weitere Details enthält der Abschnitt „PARMLIB-Änderungen“ auf Seite 16.
3. Erstellen Sie gestartete Taskprozeduren und Kompilierungs-/Verknüpfungsprozeduren. Weitere Details enthält der Abschnitt „PROCLIB-Änderungen“ auf Seite 21.
4. Aktualisieren Sie die Sicherheitsdefinitionen. Weitere Details enthält der Abschnitt „Sicherheitsdefinitionen“ auf Seite 28. Sie sollten auch wissen und verstehen, wie mit PassTickets innerhalb des Servers die Threadsicherheit gewährleistet wird. Details hierzu enthält der Abschnitt „PassTickets verwenden“ auf Seite 164.
5. Passen Sie die Konfigurationsdateien für Developer for System z an. Weitere Details enthalten die Abschnitte
 - „Konfigurationsdatei für JES Job Monitor (FEJJCNFG)“ auf Seite 29
 - „RSE-Konfigurationsdatei rsed.envvars“ auf Seite 33
 - „Konfigurationsdatei TSO/ISPFISPF.conf des TSO/ISPF-Client-Gateways von ISPF“ auf Seite 48

Anpassungskonfiguration

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

Einige Funktionen von Developer for System z erfordern, dass bestimmte Verzeichnisse in z/OS UNIX vorhanden sind. Diese Verzeichnisse müssen während der

Anpassung des Produkts erstellt werden. Zur Vereinfachung der Installation steht der Beispieljob FEKSETUP bereit, mit dem Sie die Kopien und die erforderlichen Verzeichnisse erstellen können.

Passen Sie das Beispielmembere FEKSETUP in der Datei FEK.SFEKSAMP an und übergeben Sie es, um anpassbare Kopien von Konfigurationsdateien und der Konfigurations-JCL sowie die erforderlichen z/OS UNIX-Verzeichnisse zu erstellen. Die notwendigen Anpassungsschritte sind innerhalb des Members beschrieben.

Dieser Job führt die folgenden Tasks aus:

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

Anmerkungen:

1. Für die Konfigurationsschritte in dieser Veröffentlichung werden die vom Job FEKSETUP erstellten Memberpositionen bzw. Dateipositionen verwendet, sofern nichts anderes angegeben ist. Die Originalbeispiele, die nicht aktualisiert werden sollten, finden Sie in FEK.SFEKSAMP und /usr/lpp/rdz/samples/.
2. Falls Sie alle z/OS UNIX-Dateien von Developer for System z in demselben Dateisystem (HFS oder zFS) behalten, die Konfigurationsdateien aber trotzdem in das Verzeichnis /etc/rdz stellen möchten, können Sie das Problem mit symbolischen Verbindungen lösen. Die folgenden z/OS UNIX-Beispielbefehle erstellen im vorhandenen Dateisystem ein neues Verzeichnis (/usr/lpp/rdz/cust) und definieren eine symbolische Verbindung (/etc/rdz) zu diesem Verzeichnis:

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

PARMLIB-Änderungen

Weitere Informationen zu den nachfolgend aufgelisteten PARMLIB-Definitionen enthält die Veröffentlichung *MVS Initialization and Tuning Reference* (IBM Form SA22-7592). Weitere Informationen zu den Beispielkonsolbefehlen enthält die Veröffentlichung *MVS System Commands* (IBM Form SA22-7627).

z/OS UNIX-Grenzwerte in BPXPRMxx festlegen

RSE (Remote Systems Explorer), mit dem Kernservices wie der Verbindungsaufbau vom Client zum Host bereitgestellt werden, ist ein auf z/OS UNIX basierender Prozess. Aus diesem Grund ist es wichtig, richtige Werte für die z/OS UNIX-Systemgrenzwerte in BPXPRMxx festzulegen. Diese basieren auf der Anzahl der gleichzeitig aktiven Benutzer in Developer for System z und auf ihrer durchschnittlichen Arbeitslast.

Weitere Informationen zu verschiedenen definierten BPXPRMxx-Grenzwerten und ihrer Auswirkung auf Developer for System z finden Sie in Kapitel 13, „Optimierungsaspekte“, auf Seite 215.

MAXASSIZE gibt die maximale Regionsgröße des Adressraums/Adressierungsprozesses an. Setzen Sie MAXASSIZE in SYS1.PARMLIB(BPXPRMxx) auf 2G. Dies ist der zulässige Maximalwert. Dieser Grenzwert gilt systemweit. Er ist daher für alle z/OS UNIX-Adressräume aktiv. Wenn Sie dies nicht wünschen, können Sie den Grenzwert in Ihrer Sicherheitssoftware auch nur für Developer for System z festlegen. Dies wird im Abschnitt „Gestartete Tasks für Developer for System z definieren“ auf Seite 184 beschrieben.

MAXTHREADS gibt die maximale Anzahl der aktiven Threads für einen einzelnen Prozess an. Setzen Sie MAXTHREADS in SYS1.PARMLIB(BPXPRMxx) auf mindestens 1500. Dieser Grenzwert gilt systemweit. Er ist daher für alle z/OS UNIX-Adressräume aktiv. Wenn Sie dies nicht wünschen, können Sie den Grenzwert in Ihrer Sicherheitssoftware auch nur für Developer for System z festlegen. Dies wird im Abschnitt „Gestartete Tasks für Developer for System z definieren“ auf Seite 184 beschrieben.

MAXTHREADTASKS gibt die maximale Anzahl der aktiven MVS-Tasks für einen einzelnen Prozess an. Setzen Sie MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) auf mindestens 1500. Dieser Grenzwert gilt systemweit. Er ist daher für alle z/OS UNIX-Adressräume aktiv. Wenn Sie dies nicht wünschen, können Sie den Grenzwert in Ihrer Sicherheitssoftware auch nur für Developer for System z festlegen. Dies wird im Abschnitt „Gestartete Tasks für Developer for System z definieren“ auf Seite 184 beschrieben.

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

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

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

Anmerkungen:

1. Weitere Informationen zu anderen Positionen, an denen die Größe des Adressraums gesetzt oder eingeschränkt werden kann, finden Sie im Abschnitt „Größe des Adressraums“ auf Seite 153.
2. Der oben verwendete 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.
3. Stellen Sie sicher, dass andere BPXPRMxx-Werte - wie der für MAXPROCSYS und MAXUIDS - für die erwartete Anzahl gleichzeitig aktiver Benutzer von Developer for System z ausreichen. Weitere Details hierzu enthält Kapitel 13, „Optimierungsaspekte“, auf Seite 215.

Gestartete Tasks zu COMMNDxx hinzufügen

Fügen Sie SYS1.PARMLIB(COMMANDxx) Startbefehle für die Server RSED, LOCKD und JMON von Developer for System z hinzu, damit sie beim nächsten Start über IPL automatisch gestartet werden.

Sobald die Server definiert und konfiguriert sind, können sie mit den folgenden Konsolbefehlen dynamisch (bis zum nächsten IPL) gestartet werden:

- S RSED
- S LOCKD
- S JMON

Anmerkung: Der Sperrdämon muss gestartet werden, bevor Developer for System z-Benutzer sich beim RSE-Dämon anmelden. Auf diese Weise kann der Sperrdämon die Dateisperrenanforderungen dieser Benutzer überwachen. Daher sollte der Sperrdämon beim Systemstart gestartet werden.

LPA-Definitionen in LPALSTxx

Der optionale Service von Common Access Repository Manager (CARMA) unterstützt alternative Methoden für den Serverstart, bei denen kein JES-Initiator erforderlich ist. Die flexibelste dieser Alternativen setzt voraus, dass sich das Modul CRASTART der Ladebibliothek FEK.SFEKLPA im Link-Pack-Bereich (LPA) befindet.

LPA-Dateien sind in SYS1.PARMLIB(LPALSTxx) definiert.

LPA-Definitionen können mit folgenden Konsolbefehlen dynamisch (bis zum nächsten IPL) gesetzt werden:

- SETPROG LPA,ADD,DSN=FEK.SFEKLPA

APF-Berechtigungen in PROGxx

Das Modul FEJJMON in der Ladebibliothek FEK.SFEKAUTH und die LE-Laufzeitbibliotheken (Language Environment) (CEE.SCEERUN*) müssen für APF berechtigt werden, damit JES Job Monitor auf JES-Spooldateien zugreifen kann.

Das Modul BWBTSOW in der Ladebibliothek FEK.SFEKAUTH und die REXX-Laufzeitbibliothek (REXX.*.SEAGLPA) müssen für APF berechtigt werden, damit der optionale Service von SCLM Developer Toolkit funktioniert.

Damit das TSO/ISPF-Client-Gateway von ISPF erstellt werden kann, muss das Modul ISPZTS0 in SYS1.LINKLIB für APF berechtigt werden. Das TSO/ISPF-Client-Gateway wird von TSO Commands Service, SCLM Developer Toolkit und optional von CARMA von Developer for System z verwendet.

Wenn Sie sich an Ihrem Standort nach den IBM Empfehlungen gerichtet haben, sind die APF-Berechtigungen in SYS1.PARMLIB(PROGxx) definiert.

APF-Berechtigungen können mit den folgenden Konsolbefehlen dynamisch (bis zum nächsten IPL) gesetzt werden, wobei volser für den Datenträger steht, auf dem sich die Datei befindet, sofern sie nicht von den SMS verwaltet wird:

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

Anmerkungen:

1. Wenn Sie die Alternativbibliothek für das REXX-Produktpaket verwenden, ist der Standardname der REXX-Laufzeitbibliothek REXX.*.SEAGALT anstelle von REXX.*.SEAGLPA im vorherigen Beispiel.
2. LPA-Bibliotheken, wie z. B. REXX.*.SEAGLPA, erhalten automatisch eine APF-Berechtigung, wenn sie sich in der LPA befinden und von daher keine expliziten Definitionen benötigen.
3. Für einige der zusätzlich erforderlichen Produkte ist ebenfalls eine APF-Berechtigung erforderlich. Dies gilt beispielsweise für IBM Debug Tool. Weitere Informationen hierzu enthält das Anpassungshandbuch zum jeweiligen Produkt.

LINKLIST-Definitionen in PROGxx

LINKLIST-Definitionen für Developer for System z können in 3 Kategorien gruppiert werden:

- Ladebibliotheken in Developer for System z, die für Funktionen in Developer for System z erforderlich sind. Diese Definitionen werden in diesem Abschnitt beschrieben.
- Vorausgesetzte Ladebibliotheken, die für Funktionen in Developer for System z erforderlich sind. Diese Definitionen werden im Abschnitt „Vorausgesetzte LINKLIST- und LPA-Definitionen“ auf Seite 20 beschrieben.
- Ladebibliotheken in Developer for System z, die für andere Produkte erforderlich sind. Diese Definitionen werden im Abschnitt „LINKLIST-Definitionen für andere Produkte“ auf Seite 21 beschrieben.

Alle BWB*-Module in den Ladebibliotheken FEK.SFEKAUTH und FEK.SFEKLOAD müssen mithilfe von STEPLIB oder LINKLIST verfügbar gemacht werden, damit der (optionale) Service von SCLM Developer Toolkit funktioniert.

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

- 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.

Wenn Sie sich an Ihrem Standort nach den IBM Empfehlungen gerichtet haben, sind die LINKLIST-Dateien in SYS1.PARMLIB(PROGxx) definiert.

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:

- LNKST ADD NAME(listname) DSNAME(FEK.SFEKAUTH) VOLUME(volser)
- LNKST ADD NAME(listname) DSNAME(FEK.SFEKLOAD)

LINKLIST-Definitionen können mit den folgenden Konsolbefehlen dynamisch (bis zum nächsten einleitenden Programmladen) erstellt werden. Dabei ist listname der Name der zu aktivierenden LINKLIST-Gruppe und volser steht für den Datenträger, auf dem sich die Datei befindet, sofern sie nicht im Masterkatalog katalogisiert ist:

1. LNKST DEFINE,NAME=LLTMP,COPYFROM=CURRENT

2. LNKST ADD NAME=LLTMP,DSN=FEK.SFEKAUTH,VOL=volser
3. LNKST ADD NAME=LLTMP,DSN=FEK.SFEKLOAD
4. LNKST ACTIVATE,NAME=LLTMP
5. LNKST UNDEFINE,NAME=listname
6. LNKST UPDATE,JOB=*

Vorausgesetzte LINKLIST- und LPA-Definitionen

Remote Systems Explorer (RSE) ist ein z/OS UNIX-Prozess, der auf die MVS-Ladebibliotheken zugreifen muss. Die folgenden (vorausgesetzten) Bibliotheken müssen über STEPLIB oder LINKLIST/LPALIB verfügbar sein:

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

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

- REXX-Laufzeitbibliothek (für SCLM Developer Toolkit)
 - REXX.*.SEAGLPA
- Systemladebibliothek (für SSL-Verschlüsselung)
 - SYS1.SIEALNKE
- TCP/IP-Ladebibliothek (wenn Sie für TSO Commands Service APPC verwenden)
 - TCPIP.SEZALOAD

Anmerkungen:

1. Wenn Sie die Alternativbibliothek für das REXX-Produktpaket verwenden, ist der Standardname der REXX-Laufzeitbibliothek REXX.*.SEAGALT anstelle von REXX.*.SEAGLPA im vorherigen Beispiel.
2. Bibliotheken, die in den Link-Pack-Bereich (LPA) gestellt werden müssen, wie z. B. REXX.*.SEAGLPA, erfordern unter Umständen zusätzliche Programmsteuerberechtigungen und/oder APF-Berechtigungen, wenn für den Zugriff auf diese Bibliotheken LINKLIST oder STEPLIB verwendet wird.
3. Für einige der zusätzlich erforderlichen Produkte sind ebenfalls STEPLIB- oder LINKLIST/LPALIB-Definitionen erforderlich. Dies gilt beispielsweise für IBM Debug Tool. Weitere Informationen hierzu enthält das Anpassungshandbuch zum jeweiligen Produkt.
4. Wenn CEE.SCEELKED in LINKLIST oder STEPLIB verwendet wird, muss TCPIP.SEZALOAD vor CEE.SCEELKED eingefügt werden. Andernfalls wird für die TCP/IP-REXX-Socketaufrufe ein Systemabbruch 0C1 ausgegeben.

Wenn Sie sich an Ihrem Standort nach den IBM Empfehlungen gerichtet haben, sind die LINKLIST-Dateien in SYS1.PARMLIB(PROGxx) definiert. LPA-Dateien sind in SYS1.PARMLIB(LPALSTxx) definiert.

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

- 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.

LINKLIST-Definitionen für andere Produkte

Im Client von Developer for System z gibt es eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools' (EST). Alle IRZ*- und IIRZ*-Module in der Ladebibliothek FEK.SFEKLOAD müssen mithilfe von STEPLIB oder LINKLIST verfügbar gemacht werden, damit der generierte Code Diagnosefehlernachrichten ausgeben kann.

Wenn Sie sich an Ihrem Standort nach den IBM Empfehlungen gerichtet haben, sind die LINKLIST-Dateien 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. Beachten Sie dabei allerdings Folgendes:

- 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.

PROCLIB-Änderungen

Die gestartete Task und die Prozeduren für ferne Builds, die nachfolgend aufgelistet sind, müssen sich in einer für Ihr JES definierten Prozedurenbibliothek des Systems befinden. In den folgenden Anweisungen wird die Standardprozedurenbibliothek der IBM, SYS1.PROCLIB, verwendet.

JES Job Monitor

Passen Sie das Beispielmember FEK.#CUST.PROCLIB(JMON) der gestarteten Task wie innerhalb des Members beschrieben an und kopieren Sie es in SYS1.PROCLIB. Sie müssen wie im nachstehenden Beispiel Folgendes angeben:

- das übergeordnete Qualifikationsmerkmal der (autorisierten) Ladebibliothek (standardmäßig FEK)
- die Konfigurationsdatei von JES Job Monitor (standardmäßig FEK.#CUST.PARMLIB(FEJJCNFG))

```

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

```

Abbildung 1. JMON - Gestartete Task von JES Job Monitor

Anmerkungen:

1. Weitere Informationen zu Startparametern enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.
2. Die Beispiel-JCL wird ursprünglich mit dem Namen FEK.SFEKSAMP(FEJJJCL) geliefert und dann, wie im Abschnitt „Anpassungskonfiguration“ auf Seite 15 beschrieben, in FEK.#CUST.PROCLIB(JMON) umbenannt.
3. Die Traceerstellung kann auch mit Konsolbefehlen gesteuert werden. Eine diesbezügliche Beschreibung enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.
4. Diese Task muss SYSSTC oder einem entsprechenden Ziel in Workload Manager (WLM) zugeordnet werden.
5. Für die LE-Umgebungsvariable _CEE_ENVFILE_S ist z/OS 1.8 oder höher erforderlich. Die Variable kann bei älteren z/OS-Versionen durch _CEE_ENVFILE ersetzt werden. Die TZ-Variable in der JES Job Monitor-Konfigurationsdatei (FEJJCNFG) kann aufgrund eines Fehlers in der C-Laufzeit unter Umständen jedoch falsch interpretiert werden.

RSE-Dämon

Passen Sie das Beispielmember FEK.#CUST.PROCLIB(RSED) der gestarteten Task wie innerhalb des Members beschrieben an und kopieren Sie es in SYS1.PROCLIB. Sie müssen wie im nachstehenden Beispiel Folgendes angeben:

- den RSE-Dämonport (standardmäßig 4035)
- das Ausgangsverzeichnis, in dem Developer for System z installiert ist (standardmäßig /usr/lpp/rdz)
- die Position der Konfigurationsdateien (standardmäßig /etc/rdz)

```
/*  
/* RSE-DÄMON  
/*  
//RSED      PROC IVP='',                                * 'IVP' für einen IVP-Test  
//          PORT=4035,  
//          HOME='/usr/lpp/rdz',  
//          CNFG='/etc/rdz'  
/*  
//RSE       EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,  
//          PARM='PGM &HOME/bin/rsed.sh &IVP &PORT &CNFG'  
//STDERR    DD SYSOUT=*  
//STDOUT    DD SYSOUT=*  
//          PEND  
/*
```

Abbildung 2. RSED - Gestartete Task für den RSE-Dämon

Anmerkung:

- Weitere Informationen zu Startparametern enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.
- Die Beispiel-JCL wird ursprünglich mit dem Namen FEK.SFEKSAMP(FEKRSED) geliefert und dann, wie im Abschnitt „Anpassungskonfiguration“ auf Seite 15 beschrieben, in FEK.#CUST.PROCLIB(RSED) umbenannt.
- Begrenzen Sie den Jobnamen auf maximal 7 Zeichen. Bei der Verwendung von 8 Zeichen scheitern die Bedienerbefehle **modify** und **Sstop** mit der Nachricht "IEE342I MODIFY REJECTED-TASK BUSY". Dieses Verhalten ist durch das z/OS UNIX-Design für untergeordnete Prozesse festgelegt.
- Diese Task und die von ihr erstellten untergeordneten Prozesse müssen SYSSTC oder einem entsprechenden Ziel in Workload Manager (WLM) zugeordnet werden. Die untergeordneten Prozesse haben denselben Namen wie die übergeordnete Task (RSED), dem eine einstellige Zufallszahl angehängt ist, beispielsweise RSED8.

Sperrdämon

Passen Sie das Beispielmember FEK.#CUST.PROCLIB(LOCKD) der gestarteten Task wie innerhalb des Members beschrieben an und kopieren Sie es in SYS1.PROCLIB. Sie müssen wie im nachstehenden Beispiel Folgendes angeben:

- Das Ausgangsverzeichnis, in dem Developer for System z installiert ist (standardmäßig /usr/lpp/rdz)
- Die Position der Konfigurationsdateien (standardmäßig /etc/rdz)
- Die ursprüngliche Protokolldetailebene (standardmäßig 1)

```
/*  
/* RSE LOCK DAEMON  
/*  
/*LOCKD   PROC HOME='/usr/lpp/rdz',  
/*         CNFG='/etc/rdz',  
/*         LOG=1  
/*  
/*LOCKD   EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,  
/*         PARM=PGM &HOME./bin/lockd.sh &CNFG &LOG'  
/*STDOUT   DD SYSOUT=*  
/*STDERR   DD SYSOUT=*  
/*         PEND  
/*
```

Abbildung 3. LOCKD - Gestartete Task für den Sperrdämon

Anmerkungen:

1. Weitere Informationen zu Startparametern enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.
2. Die Beispiel-JCL wird ursprünglich mit dem Namen FEK.SFEKSAMP(FEKLCKD) geliefert und dann, wie im Abschnitt „Anpassungskonfiguration“ auf Seite 15 beschrieben, in FEK.#CUST.PROCLIB(LOCKD) umbenannt.
3. Diese Task muss SYSSTC oder einem entsprechenden Ziel in Workload Manager (WLM) zugeordnet werden.

Einschränkungen der JCL für die PARM-Variable

Die Länge der PARM-Variablen liegt bei maximal 100 Zeichen. Dies kann zu Problemen führen, wenn Sie benutzerdefinierte Verzeichnisnamen verwenden. Sie können dieses Problem mit einer der folgenden Methoden umgehen:

- Verwenden Sie symbolische Links.

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

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

- Verwenden Sie STDIN.

Wenn das PARM-Feld leer ist, startet **BPXBATSL** eine z/OS UNIX-Shell und führt das von STDIN bereitgestellte Shell-Skript aus. Beachten Sie, dass STDIN eine (als ORDONLY angelegte) z/OS UNIX-Datei sein muss und dass bei Verwendung von STDIN die Verwendung von PROC-Variablen für den Port usw. inaktiviert wird. Beachten Sie auch, dass die Shell die Shellanmeldedescriptors /etc/profile und \$HOME/.profile ausführt.

Wenn Sie diese Methode anwenden möchten, müssen Sie zunächst die Start-JCL aktualisieren, damit sie in etwa wie im folgenden Beispiel aussieht:

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

Abbildung 4. RSED - Alternativer Start des RSE-Dämons

Als Zweites müssen Sie das Shell-Script erstellen (in diesem Beispiel /etc/rdz/rsed.stdin.sh), das den RSE-Dämon startet. Der Inhalt dieses Scripts sieht etwa wie das folgende Beispiel aus:

```
/long/directory/name/usr/lpp/rdz/bin/rsed.sh 4035 /etc/rdz
```

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

Anmerkung: In der Start-JCL für den Dämon sollten Sie rsed.envvars zu STDENV zuordnen, denn diese Datei definiert einige z/OS UNIX-Anweisungen, die bei Verwendung dieser Startmethoden beim Einsparen von Systemressourcen helfen.

ELAXF*-Prozeduren für ferne Builderstellung

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

Die Beispielprozeduren und ihre Funktionen sind in Tabelle 7 aufgelistet.

Tabelle 7. ELAXF-Beispielprozeduren*

Member	Zweck
ELAXFADT	Beispielprozedur für die Assemblierung und das Debugging von High-Level-Assembler-Programmen
ELAXFASM	Beispielprozedur für die Assemblierung von High-Level-Assembler-Programmen
ELAXFBMS	Beispielprozedur für die Erstellung eines CICS-BMS-Objekts und des entsprechenden Copy-, Dsect- oder Include-Members
ELAXFCOC	Beispielprozedur für COBOL-Kompilierung, integrierte CICS-Umsetzung und integrierte DB2-Umsetzung
ELAXFCOP	Beispielprozedur für die DB2-Vorverarbeitung von "EXEC SQL"-Anweisungen, die in COBOL-Programmen eingebettet sind
ELAXFCOT	Beispielprozedur für die CICS-Umsetzung von "EXEC CICS"-Anweisungen, die in COBOL-Programme eingebettet sind
ELAXFCPC	Beispielprozedur für C-Kompilierungen
ELAXFCPP	Beispielprozedur für C++-Kompilierungen
ELAXFCP1	Beispielprozedur für COBOL-Kompilierungen mit SCM-Vorprozessoranweisungen (-INC und ++INCLUDE)
ELAXFDCL	Beispielprozedur für die Ausführung eines Programms im TSO-Modus
ELAXFGO	Beispielprozedur für den GO-Schritt
ELAXFLNK	Beispielprozedur für die Verknüpfung von C/C++, COBOL-, PLI- und High-Level-Assembler-Programmen
ELAXFMFS	Beispielprozedur für die Erstellung von IMS-MFS-Anzeigen
ELAXFPLP	Beispielprozedur für die DB2-Vorverarbeitung von "EXEC SQL"-Anweisungen, die in PLI-Programme eingebettet sind
ELAXFPLT	Beispielprozedur für die CICS-Umsetzung von "EXEC-CICS"-Anweisungen, die in PLI-Programme eingebettet sind
ELAXFPL1	Beispielprozedur für PL/I-Kompilierung, integrierte CICS-Umsetzung und integrierte DB2-Umsetzung
ELAXFPP1	Beispielprozedur für PL/I-Kompilierungen mit SCM-Vorprozessoranweisungen (-INC und ++INCLUDE)
ELAXFTSO	Beispielprozedur für die Ausführung bzw. 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 Clients von Developer for System z überein. Falls Sie den Namen einer Prozedur oder eines Prozedurschritts ändern möchten, sollten Sie auch die entsprechende Eigenschaftendatei auf allen Clients aktualisieren. Das Ändern der Namen von Prozeduren oder Prozedurschritten ist nicht zu empfehlen.

Passen Sie die Member der Build-Beispielprozeduren FEK.#CUST.PROCLIB(ELAXF*) wie in den Membern beschrieben an und kopieren Sie sie in SYS1.PROCLIB. Für andere Produktbibliotheken müssen Sie die korrekten übergeordneten Qualifikationsmerkmale angeben (siehe Tabelle 8).

*Tabelle 8. Prüfliste der übergeordneten Qualifikationsmerkmale in ELAXF**

Produkt	Standard-HLQ	Wert
Developer for System z	FEK	
CICS	CICSTS32.CICS	
DB2	DSN910	
IMS	IMS	
COBOL	IGY.V4R1M0	
PL/I	IBMZ.V3R8M0	
C/C++	CBC	
LE	CEE	
LINKLIB des Systems	SYS1	
MACLIB des Systems	SYS1	

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

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

Sicherheitsdefinitionen

Passen Sie das Beispielmembere FEKRACF in der Datei FEK.#CUST.JCL an und übergeben Sie es, um die Sicherheitsdefinitionen für Developer for System z zu erstellen. Der Benutzer, der diesen Job übergibt, muss die Zugriffsrechte eines Sicherheitsadministrators haben, z. B. RACF SPECIAL.

Anmerkung:

- Für Sites, die CA ACF2™ for z/OS verwenden, rufen Sie den folgenden Link auf, um Details zu den Befehlen für die Sicherheitsfunktion zu erhalten, die für eine ordnungsgemäße Konfiguration von Developer for System z erforderlich sind: <https://support.ca.com/inj/portal/kbtech?ipLogNrow=0&docid=492389&searchID=TEC492389>.
- Für Sites, die CA Top Secret® for z/OS verwenden, rufen Sie die Seite Ihres Produkts auf der CA-Unterstützungssite (<https://support.ca.com>) auf und überprüfen Sie das zugehörige Dokument mit Informationen zu Developer for System z. Dieses Dokument enthält Details zu den Befehlen für die Sicherheitsfunktion, die für die ordnungsgemäße Konfiguration von Developer for System z erforderlich sind.

Die folgende Liste obligatorischer Sicherheitsdefinitionen für Developer for System z ist in Kapitel 10, „Sicherheitsaspekte“, auf Seite 159 ausführlich erläutert. In diesem Kapitel sind auch allgemeine Sicherheitsaspekte von Developer for System z beschrieben, einschließlich der nicht vom Beispielljob FEKRACF abgedeckten Sicherheitsaspekte vorausgesetzter Produkte.

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

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

<p>Achtung: Die Clientverbindungsanforderung schlägt fehl, wenn die Anwendungssicherheit und PassTickets nicht ordnungsgemäß konfiguriert sind.</p>
--

Konfigurationsdatei für JES Job Monitor (FEJJC�FG)

JMON (JES Job Monitor) stellt alle Services mit Bezug zum JES bereit. Das Verhalten des JES Job Monitor kann mit den Definitionen in FEJJC�FG gesteuert werden.

FEJJC�FG befindet sich in FEK.#CUST.PARMLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP (FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

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

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

```
SERV_PORT=6715
TZ=EST5EDT
#_BPXK_SETIBMOPT_TRANSPORT=TCPIP
#APPLID=FEKAPPL
#AUTHMETHOD=SAF
#CODEPAGE=UTF-8
#CONCHAR=$
#CONSOLE_NAME=JMON
#GEN_CONSOLE_NAME=OFF
#HOST_CODEPAGE=IBM-1047
#LIMIT_COMMANDS=NOLIMIT
#LIMIT_VIEW=USERID
#LISTEN_QUEUE_LENGTH=5
#MAX_DATASETS=32
#MAX_THREADS=200
#TIMEOUT=3600
#TIMEOUT_INTERVAL=1200
#SUBMITMETHOD=TSO
#TSO_TEMPLATE=FEK.#CUST.CNTL(FEJTS0)
```

Abbildung 6. FEJJC�FG (Konfigurationsdatei für JES Job Monitor)

SERV_PORT

Die Portnummer für den Hostserver mit JES Job Monitor. Der Standardport ist 6715. Eine Änderung wird angeraten, allerdings müssen die Client- und die Serverkomponente von Developer for System z mit derselben Portnummer konfiguriert werden. Wenn Sie die Server-Port-Nummer ändern, müssen alle Clients in der Ansicht 'Ferne Systeme' den JES Job Monitor-Port für dieses System ändern.

Anmerkung:

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

- Wenn Sie einen Client ab Version 7.1 verwenden, ist die Kommunikation über diesen Port auf Ihre z/OS-Hostmaschine beschränkt.

TZ Zeitzonenselektor. Die Standardeinstellung ist EST5EDT. Die Standardzeitzone ist UTC + 5 Stunden (Eastern Standard Time mit Sommerzeit). Setzen Sie diesen Wert auf Ihre Zeitzone. Weitere Informationen hierzu finden Sie in der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802).

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

_BPXK_SETIBMOPT_TRANSPORT

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

Anmerkung:

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

APPLID

Gibt die Anwendungs-ID an, die für Ihre Sicherheitssoftware zur Erkennung von JES Job Monitor verwendet wird. Die Standardeinstellung ist FEKAPPL. Entfernen Sie das Kommentarzeichen und ersetzen Sie sie durch die gewünschte Anwendungs-ID.

Anmerkung: Dieser Wert muss der Anwendungs-ID entsprechen, die in der `rsed.envvars`-Konfigurationsdatei für RSE gesetzt wurde. Wenn unterschiedliche Werte verwendet werden, ist keine Verbindung zwischen Client und JES Job Monitor möglich.

AUTHMETHOD

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

CODEPAGE

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

CONCHAR

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

CONSOLE_NAME

Gibt den Namen der EMCS-Konsole an, über die Befehle für Jobs (Hold, Release, Cancel und Purge) abgesetzt werden. Der Standardname lautet

JMON. Entfernen Sie das Kommentarzeichen und ersetzen Sie ihn unter Beachtung der untenstehenden Richtlinien durch den gewünschten Konsolennamen.

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

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

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

GEN_CONSOLE_NAME

Aktiviert bzw. inaktiviert das automatische Generieren von alternativen Konsolennamen. Die standardmäßige Einstellung ist `OFF`. Entfernen Sie das Kommentarzeichen und wählen Sie `ON` aus, um alternative Konsolennamen zu ermöglichen.

Diese Anweisung wird nur verwendet, wenn `CONSOLE_NAME` mit `&SYSUID` identisch ist und die Benutzer-ID nicht als Konsolennamen verfügbar ist. Wenn `GEN_CONSOLE_NAME=ON` ist, wird ein alternativer Konsolennamen generiert, indem der Benutzer-ID eine einzelne Ziffer hinzugefügt wird. Dafür werden die Ziffern von 0 bis 9 versucht. Wenn keine verfügbare Konsole gefunden wird, scheitert der vom Client abgesetzte Befehl.

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

Anmerkung: Die einzigen gültigen Einstellungen sind `ON` und `OFF`.

HOST_CODEPAGE

Die Host-Codepage. Die Standardeinstellung ist `IBM-1047`. Entfernen Sie das Kommentarzeichen und ändern Sie den Wert so, dass er mit Ihrer Host-Codepage übereinstimmt.

Ab Version 7.6.1 wird der hier angegebene Wert für `HOST_CODEPAGE` von Developer for System z-Clients ignoriert und die Codepage verwendet, die lokal in den Eigenschaften des "MVS Files"-Subsystems angegeben ist.

Anmerkung: Auch bei neueren Clients verwendet JES Job Monitor die Host-Codepage, die während der ersten Konfiguration der Clientkommunikation in `HOST_CODEPAGE` angegeben wird.

LIMIT_COMMANDS

Definiert, für welche Jobs der Benutzer ausgewählte JES-Befehle absetzen kann (`Show JCL`, `Hold`, `Release`, `Cancel` und `Purge`). Die Standardeinstellung (`LIMIT_COMMANDS=USERID`) schränkt die Befehle auf Jobs ein, deren Eigentümer der Benutzer ist. Entfernen Sie das Kommentarzeichen vor dieser Anweisung und geben Sie `LIMITED` oder `NOLIMIT` an, wenn der Benutzer

berechtigt sein soll, Befehle für alle Spooldateien abzusetzen (sofern dies von Ihrem Sicherheitsprodukt unterstützt wird).

Tabelle 9. Matrix der Befehlsberechtigungen für LIMIT_COMMANDS

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

Anmerkung: Die einzigen gültigen Einstellungen sind USERID, LIMITED und NOLIMIT.

LIMIT_VIEW

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

Anmerkung: Die einzigen gültigen Einstellungen sind USERID und NOLIMIT.

LISTEN_QUEUE_LENGTH

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

MAX_DATASETS

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

MAX_THREADS

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

TIMEOUT

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

TIMEOUT_INTERVAL

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

SUBMITMETHOD=TSO

Jobübergabe mithilfe von TSO. Bei Verwendung der Standardeinstellung (SUBMITMETHOD=JES) werden Jobs direkt an JES übergeben. Entfernen Sie das Kommentarzeichen vor dieser Anweisung und geben Sie TSO an, um den Job mit dem TSO-Befehl **SUBMIT** zu übergeben. Bei dieser Methode können TSO-Exits aufgerufen werden. Da sich diese Methode jedoch nachteilig auf die Leistung auswirkt, wird sie nicht empfohlen.

Anmerkung:

- Die einzigen gültigen Einstellungen sind TSO und JES.
- Wenn SUBMITMETHOD=TSO angegeben ist, muss TSO_TEMPLATE ebenfalls definiert sein.

TSO_TEMPLATE

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

Anmerkung:

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

RSE-Konfigurationsdatei rsed.envvars

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

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

Die Datei rsed.envvars befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

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

Anmerkung: Damit die Änderungen wirksam werden, müssen die gestarteten Tasks RSED und LOCKD erneut gestartet werden.

```

#=====
# (1) erforderliche Definitionen
JAVA_HOME=/usr/lpp/java/J5.0
RSE_HOME=/usr/lpp/rdz
_RSE_LOCKD_PORT=4036
_RSE_HOST_CODEPAGE=IBM-1047
TZ=EST5EDT
LANG=C
PATH=/bin:/usr/sbin
_CEE_DMPTARG=/tmp
STEPLIB=NONE
#STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBC.SCLBDLL
_RSE_SAF_CLASS=/usr/include/java_classes/IRRRacf.jar
_RSE_JAVAOPTS=""
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Xms1m -Xmx256m"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Ddaemon.log=/var/rdz/logs"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Duser.log=/var/rdz/logs"
_RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY="
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.clients=60"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.threads=1000"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dminimum.threadpool.process=1"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dmaximum.threadpool.process=100"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dipv6=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dkeep.last.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.standard.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.port.of.entry=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.certificate.mapping=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.automount=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Denable.audit.log=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.cycle=30"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Daudit.retention.period=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Ddeny.nonzero.port=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dsingle.logon=false"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dprocess.cleanup.interval=0"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DAPPLID=FEKAPPL"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDENY_PASSWORD_SAVE=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -Dhide_zos_unix=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=3600000"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"
# _RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DTSO_SERVER=APPC"
#=====
# (2) erforderliche Definitionen für TSO/ISPF-Client-Gateway
_CMDSERV_BASE_HOME=/usr/lpp/ispf
_CMDSERV_CONF_HOME=/etc/rdz
_CMDSERV_WORK_HOME=/var/rdz
#STEPLIB=$STEPLIB:ISP.SISPLPA:SYS1.LINKLIB
_RSE_CMDSERV_OPTS=""
# _RSE_CMDSERV_OPTS="$ _RSE_CMDSERV_OPTS&ISPPROF=&SYSUID..ISPPROF"
#=====
# (3) erforderliche Definitionen für SCLM Developer Toolkit
_SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:FEK.SFEKAUTH:FEK.SFEKLOAD
# _SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1
#=====
# (4) optionale Definitionen
# _RSE_PORTRANGE=8108-8118
# _BPXK_SETIBMOPT_TRANSPORT=TCPIP
# _FEKFSCMD_TP_NAME=_FEKFRSRV
# _FEKFSCMD_PARTNER_LU=_lu_name
#GSK_CRL_SECURITY_LEVEL=HIGH
#GSK_LDAP_SERVER=ldap_server_url
#GSK_LDAP_PORT=ldap_server_port
#GSK_LDAP_USER=ldap_userid
#GSK_LDAP_PASSWORD=ldap_server_password

```

Abbildung 7. rsed.envvars - RSE-Konfigurationsdatei

```

#=====
# (5) nur auf Anweisung des IBM Support Center ändern
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_BPX_SHAREAS=YES
_BPX_SPAWN_SCRIPT=YES
JAVA_PROPAGATE=NO
RSE_LIB=$RSE_HOME/lib
PATH=.:$JAVA_HOME/bin:$RSE_HOME/bin:$CMDSEV_BASE_HOME/bin:$PATH
LIBPATH=$JAVA_HOME/bin:$JAVA_HOME/bin/classic:$RSE_LIB:$RSE_LIB/icuc
LIBPATH=.:usr/lib:$LIBPATH
CLASSPATH=$RSE_LIB:$RSE_LIB/dstore_core.jar:$RSE_LIB/clientserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_extra_server.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/zosserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_miners.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/universalminers.jar:$RSE_LIB/mvsminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/carma.jar:$RSE_LIB/luceneminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsluceneminer.jar:$RSE_LIB/cdzminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvscdzminer.jar:$RSE_LIB/jesminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/FAMiner.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsutil.jar:$RSE_LIB/jesutils.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/lucene-core-2.3.2.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/cdtparser.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/wdzBidi.jar:$RSE_LIB/fmiExtensions.jar
CLASSPATH=$CLASSPATH:$RSE_SAF_CLASS
CLASSPATH=.:$CLASSPATH
_RSE_CMDSEV_OPTS="&SESSION=SPAWN$RSE_CMDSEV_OPTS"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DISPF_OPTS='$RSE_CMDSEV_OPTS'"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DA_PLUGIN_PATH=$RSE_LIB"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Xbootclasspath/p:$RSE_LIB/bidiTools.jar"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dfile.encoding=$RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dconsole.encoding=$RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_SPIRIT_ON=true"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DSPIRIT_EXPIRY_TIME=6"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DSPIRIT_INTERVAL_TIME=6"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dcom.ibm.cacheLocalHost=true"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Duser.home=$HOME"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dclient.username=$RSE_USER_ID"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dlow.heap.usage.ratio=15"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dmaximum.heap.usage.ratio=40"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_KEEPA_LIVE_ENABLED=true"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_KEEPA_LIVE_RESPONSE_TIMEOUT=30000"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DDSTORE_IO_SOCKET_READ_TIMEOUT=90000"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -DRSECOMM_LOGFILE_MAX=0"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dlock.daemon.port=$RSE_LOCKD_PORT"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -Dlock.daemon.cleanup.interval=1440"
_RSE_JAVAOPTS="$RSE_JAVAOPTS -showversion"
_RSE_SERVER_CLASS=org.eclipse.dstore.core.server.Server
_RSE_DAEMON_CLASS=com.ibm.etools.zos.server.RseDaemon
_RSE_POOL_SERVER_CLASS=com.ibm.etools.zos.server.ThreadPoolProcess
_RSE_LOCKD_CLASS=com.ibm.ftt.rse.mvs.server.miners.MVSLockDaemon
_RSE_SERVER_TIMEOUT=120000
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CMDSEV_WORK_HOME
CGI_DTWORK=$_SCLMDT_WORK_HOME
#=====
# (6) zusätzliche Umgebungsvariablen

```

Abbildung 8. (Fortsetzung)

Anmerkung: Für die Angabe von Verzeichnissen in `rsed.envvars` können Sie symbolische Links verwenden.

Folgende Definitionen sind erforderlich:

JAVA_HOME

Java-Home-Verzeichnis. Die Standardeinstellung ist `/usr/lpp/java/J5.0`. Passen Sie das Verzeichnis an Ihre Java-Installation an.

RSE_HOME

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

_RSE_LOCKD_PORT

Portnummer des RSE-Sperrdämons. Die Standardeinstellung ist 4036. Bei Bedarf können Sie diesen Port ändern.

Anmerkung:

- Überprüfen Sie vor Auswahl eines Ports, ob der Port auf Ihrem System verfügbar ist. Verwenden Sie dazu die TSO-Befehle **NETSTAT** und **NETSTAT PORTL**.
- Die gesamte Kommunikation über diesen Port ist auf Ihre z/OS-Hostmaschine beschränkt.

_RSE_HOST_CODEPAGE

Die Host-Codepage. Die Standardeinstellung ist IBM-1047. Passen Sie den Wert an Ihre Host-Codepage an.

TZ

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

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

LANG

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

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

_CEE_DMPTARG

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

STEPLIB

Zugriff auf MVS-Dateien, 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 Bereitstellung von (erforderlichen) Bibliotheken in LINKLIST/LPALIB umgehen möchten. Weitere Informationen zur Verwendung der nachfolgend aufgelisteten Bibliotheken enthält der Abschnitt „PARMLIB-Änderungen“ auf Seite 16.

```
STEPLIB=$STEPLIB:CEE.SCEERUN:CEE.SCEERUN2:CBCLBDDL
STEPLIB=$STEPLIB:ISP.SISPLoad:ISP.SISPLPA:SYS1.LINKLIB
STEPLIB=$STEPLIB:FEK.SFEKAUTH:FEK.SFEKLOAD
```

Anmerkung:

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

RSE_SAF_CLASS

Gibt die Java-Schnittstellen zu Ihrem Sicherheitsprodukt an. Die Standardeinstellung ist /usr/include/java_classes/IRRRacf.jar. Passen Sie die Einstellung an die Konfiguration Ihrer Sicherheitssoftware an.

Anmerkung: Ab z/OS 1.10 ist /usr/include/java_classes/IRRRacf.jar Teil der System Authorization Facility (SAF), die zum Lieferumfang des Basisprodukts z/OS gehört. Damit ist die JAR-Datei auch für Kunden verfügbar, die kein RACF verwenden.

RSE_JAVAOPTS

Zusätzliche RSE-spezifische Java-Optionen. . Weitere Informationen zu dieser Definition enthält der Abschnitt „Zusätzliche Java-Startparameter mit _RSE_JAVAOPTS definieren“ auf Seite 42.

Developer for System z verwendet standardmäßig das TSO/ISPF-Client-Gateway von ISPF für TSO Commands Service. Wenn die folgende _RSE_JAVAOPTS-Option nicht auf Kommentar gesetzt ist, wird stattdessen eine APPC-Transaktion verwendet:

```
RSE_JAVAOPTS="$ _RSE_JAVAOPTS -DTSO_SERVER=APPC"
```

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

_CMDSERV_BASE_HOME

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

_CMDSERV_CONF_HOME

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

_CMDSERV_WORK_HOME

Basisarbeitsverzeichnis für ISPF. Die Standardeinstellung ist /var/rdz. Pas-

sen Sie die Position an das vom TSO/ISPF-Client-Gateway verwendete Verzeichnis WORKAREA an. Diese Anweisung ist nur erforderlich, wenn das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

Anmerkungen:

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

STEPLIB

STEPLIB ist im Abschnitt über die erforderlichen Definitionen beschrieben.

RSE_CMDSERV_OPTS

Zusätzliche, für das TSO/ISPF-Client-Gateway spezifische Java-Optionen. Die Standardeinstellung ist "". Weitere Informationen zu dieser Definition enthält der Abschnitt „Zusätzliche Java-Startparameter mit _RSE_CMDSERV_OPTS definieren“ auf Seite 47. Diese Anweisung ist nur erforderlich, wenn das TSO/ISPF-Client-Gateway von ISPF verwendet wird.

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

SCLMDT_CONF_HOME

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

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

STEPLIB

STEPLIB ist im Abschnitt über die erforderlichen Definitionen beschrieben.

_SCLMDT_TRANTABLE

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

ANT_HOME

Ausgangsverzeichnis für Ihre Ant-Installation. Die Standardeinstellung ist /usr/lpp/apache/Ant/apache-ant-1.7.1. Passen Sie das Verzeichnis an Ihre Ant-Installation an. Diese Anweisung ist nur erforderlich, wenn SCLM Developer Toolkit mit der JAVA/J2EE-Buildunterstützung verwendet wird.

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

_RSE_PORTRANGE

Gibt den Bereich der Ports an, die der RSE-Server für die Kommunikation

mit einem Client öffnen kann. Standardmäßig kann jeder Port verwendet werden. Weitere Informationen zu dieser Definition enthält der Abschnitt „Für RSE-Server verfügbaren PORTRANGE definieren“ auf Seite 41. Diese Anweisung ist optional.

_BPXK_SETIBMOPT_TRANSPORT

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

Anmerkung:

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

_FEKFSCMD_TP_NAME

Name des APPC-Transaktionsprogramms. Die Standardeinstellung ist FEKFRSRV. Entfernen Sie das Kommentarzeichen vor dieser Definition und ändern Sie sie, wenn Sie beim Definieren der APPC-Transaktion nicht den Standardnamen verwendet haben. Diese Anweisung ist optional.

_FEKFSCMD_PARTNER_LU

Zwingt den RSE-Server, diese APPC-Partner-LU zu verwenden. Standardmäßig wird die während der APPC-Konfiguration angegebene Basis-LU verwendet. Diese Anweisung ist optional.

GSK_CRL_SECURITY_LEVEL

Gibt die Sicherheitsstufe an, die von SSL beim Herstellen einer Verbindung zu LDAP-Servern während der Zertifikatsüberprüfung verwendet wird, um CRLs auf widerrufen Zertifikate zu überprüfen. Die Standardeinstellung ist MEDIUM. Entfernen Sie das Kommentarzeichen und ändern Sie sie, damit der angegebene Wert angewendet wird. Diese Anweisung ist optional. Folgende Werte sind gültig:

- LOW - Die Zertifikatsüberprüfung scheitert nicht, wenn keine Verbindung mit dem LDAP-Server hergestellt werden kann.
- MEDIUM - Die Zertifikatsüberprüfung erfordert die Erreichbarkeit des LDAP-Servers, jedoch keine Definition einer CRL. Dies ist die Standardeinstellung.
- HIGH - Die Zertifikatsüberprüfung erfordert die Erreichbarkeit des LDAP-Servers und die Definition einer CRL.

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

GSK_LDAP_SERVER

Gibt einen oder mehrere durch Leerzeichen getrennte Hostnamen der LDAP-Server an. Entfernen Sie das Kommentarzeichen und ändern Sie sie, um die Verwendung der angegebenen LDAP-Server zu erzwingen, damit diese ihre CRL erhalten. Diese Anweisung ist optional.

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

GSK_LDAP_PORT

Gibt den LDAP-Server-Port an. Der Standardport ist 389. Entfernen Sie das Kommentarzeichen und ändern Sie den Wert, um die Verwendung des angegebenen Werts zu erzwingen. Diese Anweisung ist optional.

GSK_LDAP_USER

Gibt den definierten Namen an, der bei der Verbindung mit dem LDAP-Server verwendet wird. Entfernen Sie das Kommentarzeichen und ändern Sie ihn, damit der angegebene Wert angewendet wird. Diese Anweisung ist optional.

GSK_LDAP_PASSWORD

Gibt das Kennwort an, das bei der Verbindung mit dem LDAP-Server verwendet wird. Entfernen Sie das Kommentarzeichen und ändern Sie den Wert, um die Verwendung des angegebenen Werts zu erzwingen. Diese Anweisung ist optional.

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

_CEE_RUNOPTS

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

_BPX_SHAREAS

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

_BPX_SPAWN_SCRIPT

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

JAVA_PROPAGATE

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

RSE_LIB

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

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

LIBPATH

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

CLASSPATH

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

_RSE_CMDSERV_OPTS

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

_RSE_JAVAOPTS

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

_RSE_SERVER_CLASS

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

_RSE_DAEMON_CLASS

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

_RSE_POOL_SERVER_CLASS

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

_RSE_LOCKD_CLASS

Java-Klassen für den RSE-Sperrdämon. Die Standardeinstellung ist `com.ibm.ftt.rse.mvs.server.miners.MVSLockDaemon`. Modifizieren Sie diese Einstellung nicht.

_RSE_SERVER_TIMEOUT

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

SCLMDT_BASE_HOME

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

SCLMDT_WORK_HOME

Basisarbeitsverzeichnis von SCLM Developer Toolkit. Die Standardeinstellung ist `$_CMDSERV_WORK_HOME`. Modifizieren Sie diese Einstellung nicht.

CGI_DTWORK

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

Für RSE-Server verfügbaren PORTRANGE definieren

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

Nachfolgend sehen Sie eine kurze Beschreibung des RSE-Verbindungsprozesses, die Ihnen helfen soll, die Portverwendung zu verstehen.

1. Der Client stellt über den Host-Port 4035 eine Verbindung mit dem RSE-Dämon her.
2. Der RSE-Dämon erstellt einen RSE-Server-Thread.
3. Der RSE-Server öffnet einen Host-Port, zu dem der Client eine Verbindung herstellen kann. Der Benutzer kann die Auswahl dieses Ports auf dem Client auf der Eigenschaftenregisterkarte für das Subsystem (nicht zu empfehlen) oder mit der Definition `_RSE_PORTRANGE` in `rsed.envvars` konfigurieren.
4. Der RSE-Dämon gibt die Portnummer an den Client zurück.
5. Der Client stellt eine Verbindung mit dem Host-Port her.

Anmerkung:

- Dieser Prozess ist mit der (optionalen) alternativen Verbindungsmethode unter Verwendung von REXEC/SSH vergleichbar.
- Weitere Informationen hierzu enthält Kapitel 11, „Wissenswertes zu Developer for System z“, auf Seite 193.

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

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

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

PORTRANGE hat folgendes Format: `_RSE_PORTRANGE=min-max`. (Die Angabe max gilt nicht einschließlich. Die Einstellung `_RSE_PORTRANGE=8108-8118` bedeutet beispielsweise, dass die Portnummern von 8108 bis einschließlich 8117 verwendet werden können.) Die vom RSE-Server verwendete Portnummer wird wie folgt ermittelt:

1. Wenn in den Subsystemeigenschaften auf dem Client eine Portnummer ungleich null angegeben ist, wird diese Portnummer verwendet. Ist der Port nicht verfügbar, scheitert die Verbindung. Diese Konfiguration wird nicht empfohlen.

Anmerkung: Der Host kann diesen Verbindungsanforderungstyp zurückweisen, wenn Sie die Anweisung `deny.nonzero.port=true` in `rsed.envvars` angeben. Weitere Informationen zu dieser Anweisung erhalten Sie unter „Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren“.

2. Wenn die Portnummer in den Subsystemmerkmalen null ist und in `rsed.envvars` `_RSE_PORTRANGE` enthalten ist, wird der von `_RSE_PORTRANGE` angegebene Portbereich verwendet. Falls kein Port aus dem Bereich verfügbar ist, scheitert die Verbindung.
3. Wenn die Portnummer in den Subsystemmerkmalen null ist und `_RSE_PORTRANGE` nicht in `rsed.envvars` enthalten ist, wird jeder verfügbare Port verwendet.

Anmerkung: Wenn ein Webserver einen Port öffnet und auf den Empfang von Daten wartet, kann dieser Port nicht von einem anderen Server verwendet werden. Ist die Verbindung jedoch einmal hergestellt, können mehrere Server denselben Port verwenden. Die Portnummern im Portbereich bedeuten also keine Einschränkung hinsichtlich der Anzahl gleichzeitig verbundener Benutzer.

Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren

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

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

```
_RSE_JAVAOPTS=""
```

Variableninitialisierung. Modifizieren Sie diese Einstellung nicht.

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms1m -Xmx256m"`

Festlegen der anfänglichen Heapgröße (Xms) und der maximalen Heapgröße (Xmx). Die Standardwerte sind jeweils 1M und 256M. Ändern Sie diese, um die gewünschten Werte der Heapgröße zu erzwingen. Wenn diese Anweisung in Kommentarzeichen gesetzt ist, werden die Java-Standardwerte verwendet. Diese sind 4M beziehungsweise 512M (1M und 64M für Java 5.0).

Anmerkung: Weitere Informationen zum Ermitteln der optimalen Werte für diese Anweisung enthält „Definitionen von wichtigen Ressourcen“ auf Seite 235.

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

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

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

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

Verzeichnis für die benutzerspezifischen Protokolle. Die Standardeinstellung ist /var/rdz/logs. Ändern Sie diese, um die gewünschte Position zu erzwingen. Wenn diese Anweisung in Kommentarzeichen gesetzt ist, wird das Ausgangsverzeichnis der Client-Benutzer-ID verwendet. Das Ausgangsverzeichnis ist im Segment für die OMVS-Sicherheit der Benutzer-ID definiert.

Anmerkung:

- Wenn diese Anweisung (oder das entsprechende Ausgangsverzeichnis) keinen absoluten Pfad angibt, (der Pfad beginnt nicht mit einem Schrägstrich (/)), ist die aktuelle Position des Protokolls relativ zum Konfigurationsverzeichnis (standardmäßig /etc/rdz).
- Der vollständige Pfad zu den Benutzerprotokollen ist userlog/dstorelog/\$LOGNAME/. Dabei ist userlog der Wert der Anweisung user.log, dstorelog ist der Wert der Anweisung DSTORE_LOG_DIRECTORY und \$LOGNAME ist die Benutzer-ID des Clients in Großbuchstaben.
- Stellen Sie sicher, dass die Berechtigungsbits für userlog/dstorelog so festgelegt sind, dass jeder Client \$LOGNAME erstellen kann.

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY=`

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

Anmerkung:

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

Die folgenden Anweisungen sind standardmäßig auf Kommentar gesetzt.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.clients=60"

Maximale Anzahl der Clients, die ein Thread-Pool bedienen kann. Die Standardeinstellung ist 60. Entfernen Sie das Kommentarzeichen und passen Sie die Option an, um die Anzahl der Clients pro Thread-Pool zu begrenzen. Beachten Sie, dass andere Grenzwerte möglicherweise verhindern, dass RSE diese Begrenzung erreicht.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dmaximum.threads=1000"

Maximale Anzahl von aktiven Threads in einem Thread-Pool, um neue Clients zuzulassen. Die Standardeinstellung ist 1000. Entfernen Sie das Kommentarzeichen und passen Sie den Wert an, um die Anzahl der Clients pro Thread-Pool auf Basis der Threads in Gebrauch zu begrenzen. Beachten Sie, dass jede Clientverbindung mehrere Threads (16 oder mehr) verwendet und dass andere Grenzwerte verhindern können, dass RSE diese Begrenzung erreicht.

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

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

Minimale Anzahl aktiver Thread-Pools. Die Standardeinstellung ist 1. Entfernen Sie das Kommentarzeichen und passen Sie die Option an, damit mindestens die angegebene Anzahl von Thread-Pool-Prozessen gestartet wird. Thread-Pool-Prozesse werden für die Lastverteilung der RSE-Server-Threads verwendet. Weitere neue Prozesse werden bei Bedarf gestartet. Wenn die neuen Prozesse vorab gestartet werden, werden Verzögerungen bei Verbindungen verhindert. Das System verwendet in Leerlaufzeiten allerdings mehr Ressourcen.

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

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

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

TCP/IP-Version. Die Standardeinstellung ist `false`. Dies bedeutet, dass eine IPv4-Schnittstelle verwendet wird. Entfernen Sie das Kommentarzeichen und geben Sie `true` an, damit eine IPv6-Schnittstelle verwendet wird.

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

Eine Kopie der Hostprotokolldateien aufbewahren, die zur vorherigen Sit-

zung gehören. Die Standardeinstellung ist false. Entfernen Sie das Kommentarzeichen und geben Sie true an, um die vorherigen Protokolldateien während des Serverstarts und dem Clientverbindungsaufbau in *.last umzubenennen.

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

Die Datenströme 'stdout' und 'stderr' des Thread-Pools in eine Protokolldatei schreiben. Die Standardeinstellung ist false. Entfernen Sie das Kommentarzeichen und geben Sie true an, um die Datenströme 'stdout' und 'stderr' zu speichern. Die Protokolldateien befinden sich in dem Verzeichnis, auf das die Anweisung daemon.log verweist.

Anmerkung:

- Mit dem Bedienerbefehl **MODIFY RSESTANDARDLOG** können Sie die Aktualisierung der Datenstromprotokolldateien dynamisch stoppen oder starten.
- Wenn die Anweisung enable.standard.log aktiv ist, gibt es keine benutzerspezifischen Protokolldateien stdout.log und stderr.log. Die benutzerspezifischen Daten werden jetzt in den entsprechenden RSE-Thread-Pool-Datenstrom geschrieben.

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

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

Anmerkung:

- Die Überprüfung des Eingangsports muss auch in Ihrem Sicherheitsprodukt aktiviert sein.
- Wenn Sie die Überprüfung des Eingangsports aktiviert haben, ist sie auch für andere z/OS UNIX-Services wie IN-ETD aktiviert.

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

Ihre Sicherheitssoftware verwenden, um eine Anmeldung mit einem X.509-Zertifikat zu authentifizieren. Die Standardeinstellung ist true. Entfernen Sie das Kommentarzeichen und geben Sie false an, damit die Authentifizierung vom RSE-Dämon unabhängig von der X.509-Unterstützung Ihrer Sicherheitssoftware erfolgen kann.

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

Unterstützung für Ausgangsverzeichnisse, die von dem z/OS UNIX-Automount erstellt wurden. Die Standardeinstellung ist false. Entfernen Sie das Kommentarzeichen und geben Sie true an, um sicherzustellen, dass der z/OS UNIX-Automount die Client-ID als Eigentümer des Verzeichnisses verwendet.

Anmerkung: Der z/OS UNIX-Automount verwendet die Benutzer-ID des Prozesses, der den Service beim Erstellen eines Dateisystems aufgerufen hat. Wenn diese Option inaktiviert ist, ist dieser Prozess der RSE-Thread-Pool-Server (Benutzer-ID: STCRSE).

Wenn diese Option aktiviert ist, wird mithilfe der Benutzer-ID ein neuer, temporärer Prozess erstellt, bevor der Service aufgerufen wird.

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

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

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

Anzahl der in einer Prüfprotokolldatei gespeicherten Tage mit Prüfinformationen. Die Standardeinstellung ist 30. Entfernen Sie das Kommentarzeichen, wenn Sie steuern möchten, wie viele Prüfdaten in eine Prüfprotokolldatei geschrieben werden. Der Maximalwert ist 365.

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

Anzahl der geführten Prüfprotokolle. Die Standardeinstellung ist 0 (keine Begrenzung). Entfernen Sie das Kommentarzeichen und passen Sie die Einstellung an, wenn Protokolle nach einer bestimmten Anzahl von Tagen gelöscht werden sollen. Der Maximalwert ist 365.

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

Nicht zulassen, dass der Client die Portnummer für die Kommunikation auswählt. Die Standardeinstellung ist false. Entfernen Sie das Kommentarzeichen und geben Sie true an, damit Verbindungen zurückgewiesen werden, bei denen der Client angibt, welcher Host-Port vom RSE-Server für die Verbindung verwendet werden soll. Weitere Informationen hierzu enthält „Für RSE-Server verfügbaren PORTRANGE definieren“ auf Seite 41.

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

Nicht zulassen, dass sich ein Benutzer mit derselben Benutzer-ID mehrere Male anmeldet. Die Standardeinstellung ist true. Entfernen Sie das Kommentarzeichen und geben Sie false an, damit sich ein Benutzer mit derselben ID auf einem einzelnen RSE-Dämon mehrere Male anmelden kann.

Anmerkung: Wenn diese Anweisung nicht aktiv oder auf false gesetzt ist, wird bei einem zweiten Anmeldeversuch die erste Anmeldung vom Host beendet. Gleichzeitig wird die Konsolnachricht "FEK210I" angezeigt.

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

RSE-Thread-Pools mit nicht behebbaren Fehlern automatisch entfernen. Standardmäßig werden fehlerhafte RSE-Thread-Pools nicht automatisch entfernt. Entfernen Sie das Kommentarzeichen und passen den Wert an, damit fehlerhafte RSE-Thread-Pool-Server in bestimmten Intervallen (Intervalleinheit: Sekunden) automatisch entfernt werden. Mit 0 wird die Funktion inaktiviert.

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

ID der RSE-Serveranwendung. Die Standardeinstellung ist FEKAPPL. Entfernen Sie das Kommentarzeichen und passen Sie diese Option an, um die Verwendung der gewünschten Anwendungs-ID zu aktivieren.

Anmerkung:

- Die Anwendungs-ID muss für Ihre Sicherheitssoftware definiert sein. Wenn dies nicht der Fall ist, wird die Clientanmeldung verhindert.

- Weitere Informationen zu den Sicherheitsauswirkungen bei Änderungen dieses Werts erhalten Sie unter „PassTickets verwenden“ auf Seite 164.
- Die Anwendungs-ID muss mit der von JES Job Monitor verwendeten Anwendungs-ID übereinstimmen. Weitere Informationen zum Definieren der Anwendungs-ID für JES Job Monitor erhalten Sie unter „Konfigurationsdatei für JES Job Monitor (FEJJCNFG)“ auf Seite 29.

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

Option für die Kennwortspeicherung. Die Standardeinstellung ist false. Entfernen Sie das Kommentarzeichen und geben Sie true an, um zu verhindern, dass Benutzer ihr Hostkennwort auf dem Client speichern. Bereits gespeicherte Kennwörter werden dann entfernt. Diese Option funktioniert nur bei Clients ab der Version 7.1.

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

Option zum Ausblenden von z/OS UNIX. Die Standardeinstellung ist false. Entfernen Sie das Kommentarzeichen und geben Sie true an, damit Benutzer keine z/OS UNIX-Elemente (Verzeichnisstruktur und Befehlszeile) im Client sehen können. Diese Option funktioniert nur bei Clients ab Version 7.6.

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

Trennung der Verbindung inaktiver Clients. Standardmäßig wird die Verbindung inaktiver Clients nicht unterbrochen. Entfernen Sie das Kommentarzeichen und passen Sie die Einstellung an, um die Verbindung von Clients zu trennen, die die angegebene Zahl von Millisekunden (3600000 entspricht 1 Stunde) inaktiv sind.

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

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

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

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

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DTSO_SERVER=APPC"

Verwendung einer APPC-Transaktion für TSO Commands Service. Standardmäßig wird das TSO/ISPF-Client-Gateway von ISPF verwendet. Entfernen Sie das Kommentarzeichen, um stattdessen eine APPC-Transaktion zu verwenden. Ändern Sie die zugeordneten Werte nicht.

Zusätzliche Java-Startparameter mit **_RSE_CMDSERV_OPTS** definieren

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

Die **_RSE_CMDSERV_OPTS**-Anweisungen sind RSE-spezifische Java-Optionen, die nur wirksam sind, wenn für Developer for System z das TSO/ISPF-Client-Gateway von ISPF verwendet wird. (Dies ist die Standardeinstellung.)

`_RSE_CMDSERV_OPTS=""`

Variableninitialisierung. Modifizieren Sie diese Einstellung nicht.

`_RSE_CMDSERV_OPTS="$_RSE_CMDSERV_OPTS &ISPROF=&SYSUID..IS-
PROF=""`

Verwendung eines vorhandenen ISPF-Profiles für die ISPF-Initialisierung.
Entfernen Sie das Kommentarzeichen und ändern Sie den Dateinamen, um
das angegebene ISPF-Profil zu verwenden.

Im Dateinamen können die folgenden Variablen verwendet werden:

- &SYSUID als Ersatz für die Benutzer-ID des Entwicklers
- &SYSPREF als Ersatz für das TSO-Präfix des Entwicklers

Konfigurationsdatei TSO/ISPFISPF.conf des TSO/ISPF-Client-Gateways von ISPF

Das TSO/ISPF-Client-Gateway von ISPF erstellt ausgehend von den Definitionen in ISPF.conf eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen. Developer for System z führt in dieser Umgebung einige MVS-basierte Services aus. Dazu gehören TSO Commands Service, der Service von SCLM Developer Toolkit und eine alternative CARMA-Startmethode.

Die Datei ISPF.conf befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

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

Sie müssen nicht nur die korrekten Namen für die ISPF-Dateien angeben, sondern auch den Dateinamen für TSO Commands Service, FEK.SFEKPROC, zur Anweisung SYSPROC oder SYSEXEC hinzufügen. Vergleichen Sie hierzu das folgende Beispiel.

```
* ERFORDERLICH:  
sysproc=ISP.SISPCLIB,FEK.SFEKPROC  
ispmllib=ISP.SISPMENU  
isptlib=ISP.SISPTENU  
ispplib=ISP.SISPPENU  
ispslib=ISP.SISPSLIB  
ispllib=ISP.SISPLOAD  
  
* OPTIONAL:  
*allocjob = FEK.#CUST.CNTL(CRAISPRX)  
*ISPF_timeout = 900
```

Abbildung 9. ISPF-Konfigurationsdatei ISPF.conf

Anmerkung:

- Sie können Ihre eigenen DD-Anweisungen und Dateiverkettungen hinzufügen, um die TSO-Umgebung anzupassen und so eine TSO-Anmeldeprozedur zu imitieren. Weitere Details enthält Kapitel 16, „TSO-Umgebung anpassen“, auf Seite 269.

- Das TSO/ISPF-Client-Gateway funktioniert möglicherweise nicht ordnungsgemäß, wenn Sie ein Produkt (eines Fremdanbieters) verwenden, das ISPF-Befehle wie **ISPSTART** abfängt. Stellen Sie anhand der Dokumentation zu diesem Produkt fest, wie es für Developer für System z inaktiviert werden kann. Wenn das Produkt die Zuordnung einer bestimmten DD-Anweisung zu DUMMY erfordert, können Sie dies in ISPF.conf simulieren, indem Sie die DD-Anweisung nullfile zuordnen.

Beispiel:

```
ISPTRACE=nullfile
```

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

Optionale Komponenten

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

- Kapitel 3, „Common Access Repository Manager (optional)“, auf Seite 51
- Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73
- Kapitel 5, „SCLM Developer Toolkit (optional)“, auf Seite 81
- „Gespeicherte DB2-Prozedur (optional)“ auf Seite 89
- „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 92
- „RSE-SSL-Verschlüsselung (optional)“ auf Seite 93
- „RSE-Trace (optional)“ auf Seite 96
- „Hostbasierte Eigenschaftsgruppe (optional)“ auf Seite 98
- „Hostbasierte Projekte (optional)“ auf Seite 99
- „File Manager-Integration (optional)“ auf Seite 100
- „Nicht editierbare Zeichen (optional)“ auf Seite 101
- „REXEC (oder SSH) verwenden (optional)“ auf Seite 102
- „APPC-Transaktion für TSO Commands Service (optional)“ auf Seite 104
- „WORKAREA-Bereinigung (optional)“ auf Seite 108

Installationsprüfung

Eine Beschreibung der verschiedenen Installationsprüfprogramme (IVPs) finden Sie unter Kapitel 7, „Installationsprüfung“, auf Seite 109, da einige IVPs für optionale Komponenten gelten.

Kapitel 3. Common Access Repository Manager (optional)

Common Access Repository Manager (CARMA) ist eine Produktivitätshilfe für Entwickler, die RAM (Repository Access Managers) erstellen. Ein RAM ist eine Anwendungsprogrammierschnittstelle (API) für z/OS-basierte SCMs (Software Configuration Managers).

Vom Benutzer geschriebene Anwendungen können einen CARMA-Server starten, der die RAM lädt und eine Standardschnittstelle für den Zugriff auf den SCM bereitstellt.

Developer for System z unterstützt mehrere Startmethoden für einen CARMA-Server, die jeweils spezielle Vor- und Nachteile haben.

- Die Methode der Batchübergabe startet den CARMA-Server durch Übergabe eines Jobs. Dies ist die in den bereitgestellten Beispielkonfigurationsdateien 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 pro Entwickler, der einen CARMA-Server startet, ein JES-Initiator verwendet.
- Die Methode "CRASTART" startet den CARMA-Server als Subtask innerhalb von RSE. Bei dieser sehr flexiblen Konfiguration wird eine gesonderte Konfigurationsdatei verwendet, 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.
- Bei der Methode "TSO/ISPF-Client-Gateway" wird mit dem TSO/ISPF-Client-Gateway von ISPF eine TSO- oder ISPF-Umgebung erstellt, in der der CARMA-Server gestartet wird. Diese Methode erlaubt flexible Dateizuordnungen mithilfe der Datei ISPF.conf. Sie ist jedoch nicht für den Zugriff auf SCMs geeignet, die mit normalen TSO- oder ISPF-Operationen in Konflikt geraten.

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-Portbereich für interne Kommunikation
- Sicherheitsregel, die Entwicklern die Aktualisierung der CARMA-VSAMs-Dateien erlaubt
- Sicherheitsregel, die Benutzern die Übergabe von CRA*-Jobs erlaubt (optional)
- Aktualisierung des LPA (optional)

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

1. Erstellen Sie die erforderlichen CARMA-Komponenten. Weitere Details enthält der Abschnitt „CARMA-Komponenten“ auf Seite 52.
2. 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 „RSE-Schnittstelle zu CARMA“ auf Seite 53.

3. Wählen Sie eine Startmethode für CARMA aus und führen Sie die erforderliche Anpassung der entsprechenden Konfigurationsdateien aus. Weitere Details enthalten die folgenden Abschnitte:
 - „CARMA-Serverstart mit Batchübergabe“ auf Seite 56
 - „Alternativer CARMA-Serverstart mit CRASTART (optional)“ auf Seite 57
 - „Alternativer CARMA-Serverstart mit TSO/ISPF-Client-Gateway (optional)“ auf Seite 60
4. Aktivieren Sie die Beispiel-RAM (Repository Access Manager) (optional). Weitere Details enthält der Abschnitt „Beispiel-RAM (Repository Access Manager) aktivieren (optional)“ auf Seite 62.
5. Aktivieren Sie den CA Endevor®-RAM (optional). Weitere Details enthält der Abschnitt „CA Endevor® SCM-RAM aktivieren (optional)“ auf Seite 64.
6. Erstellen Sie CRAXJCL, um IRXJCL zu ersetzen (optional). Weitere Details enthält der Abschnitt „IRXJCL oder CRAXJCL (optional)“ auf Seite 72.

Anmerkung: Die Beispielmembere aus diesem Kapitel sind in FEK.#CUST.* und /etc/rdz enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

CARMA-Komponenten

Die folgenden CARMA-Komponenten müssen unabhängig von der gewählten Startmethode angepasst werden. Die unten angegebenen Beispielmembere befinden sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

1. Passen Sie die JCL FEK.#CUST.JCL(CRA\$VDEF) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA\$VDEF enthaltenen Dokumentation. CRA\$VDEF erstellt die VSAM-Datei mit der CARMA-Konfiguration CRADEF und setzt die erforderlichen Daten ein.
2. Passen Sie die JCL FEK.#CUST.JCL(CRA\$VMSG) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA\$VMSG enthaltenen Dokumentation. CRA\$VMSG erstellt die VSAM-Datei für CARMA-Nachrichten CRAMSG und setzt die erforderlichen Daten ein.
3. Passen Sie die JCL FEK.#CUST.JCL(CRA\$VSTR) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA\$VSTR enthaltenen Dokumentation. CRA\$VSTR erstellt die VSAM-Datei mit benutzerdefinierten CARMA-Informationen CRASTRS und setzt die erforderlichen Daten ein.

Anmerkung:

- Die CARMA-VSAMs, die mit diesen Jobs erstellt werden, definieren die Beispiel-RAMs. Im Abschnitt „CA Endevor® SCM-RAM aktivieren (optional)“ auf Seite 64 ist beschrieben, wie der CA Endevor®-RAM definiert wird.
- Richten Sie sich nach dem Beispieljob FEK.#CUST.JCL(CRA#UADD), wenn Sie die Definitionen für einen (benutzerdefinierten) RAM mit einer vorhandenen VSAM-Konfiguration zusammenführen müssen. Dieser Job muss für jede geänderte CARMA-VSAM angepasst und übergeben werden. Weitere Informationen zur Satzstruktur, die von den verschiedenen CARMA-VSAMs verwendet wer-

den, finden Sie im *Rational Developer for System z Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).

- Verwenden Sie den Beispieljob FEK.#CUST.JCL(CRA#UQRY), um die aktiven Definitionen aus einer VSAM in eine sequenzielle Datei zu extrahieren.

Migrationshinweise für CARMA-VSAM

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

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

Developer for System z Version 7.6.1 unterstützt ein neues Datenstrukturlayout mit variabler Länge für die VSAM-Datei für angepasste CARMA-Informationen (CRASTRS), bei dem Zeichenfolgen durch ein Begrenzungszeichen getrennt werden, anstatt einer festen Länge zu unterliegen.

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

Anmerkung:

- Ab Version 7.6.1 wird die VSAM-Beispieldatei für angepasste CARMA-Informationen im Format mit variabler Länge ausgeliefert.
- Ab Version 7.6.1 unterstützt das CARMA-Lademodul (CRASERV) sowohl Formate mit festen Längen als mit variablen Längen für die VSAM-Datei für angepasste CARMA-Informationen.
- Ältere Versionen des CARMA-Lademoduls unterstützen keine Formate mit variabler Länge. Wenn trotzdem eine VSAM-Datei für angepasste CARMA-Informationen mit einem Format mit variabler Länge verwendet wird, werden die Zeichenfolgen beschädigt.

RSE-Schnittstelle zu CARMA

Der CARMA-Server stellt für andere hostbasierte Produkte eine Standard-API für den Zugriff auf Software Configuration Manager (SCM) bereit. CARMA bietet jedoch keine Methoden für die direkte Kommunikation mit einem Client-PC an. Aus diesem Grund ist CARMA auf andere Produkte wie den RSE-Server angewiesen. Der RSE-Server verwendet die Einstellungen in CRASRV.properties zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server.

Die Datei CRASRV.properties befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl OEDIT bearbeiten.

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

```
# CRASRV.properties - CARMA-Konfigurationsoptionen
#
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname='FEK.#CUST.CNTL(CRASUBMT)'
crastart.stub=/usr/lpp/rdz/bin/CRASTART
crastart.configuration.file=/etc/rdz/crastart.conf
crastart.syslog=Partial
crastart.timeout=420
#crastart.steplib=FEK.SFEKLPA
#crastart.tasklib=TASKLIB
```

Abbildung 10. CARMA-Konfigurationsdatei CRASRV.properties

port.start

Der erste Port, der für die Kommunikation zwischen CARMA und dem RSE-Server verwendet wird. Der Standardport ist 5227. Die Kommunikation über diesen Port ist auf Ihre Hostmaschine 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 hierzu enthält der Abschnitt „Reservierte TCP/IP-Ports“ auf Seite 152.

port.range

Portbereich, der mit port.start beginnt und für die CARMA-Kommunikation verwendet wird. Die Standardeinstellung ist 100. Wenn Sie die Standardeinstellungen verwenden, kann CARMA beispielsweise die Ports 5227 bis einschließlich 5326 nutzen.

startup.script.name

Definiert den absoluten Pfad des CARMA-Start-Scripts. Die Standardeinstellung ist /usr/lpp/rdz/bin/carma.startup.rex. Diese REXX-Exec löst den Start eines CARMA-Servers aus.

clist.dsname

Definiert die Startmethode für den CARMA-Server.

- *CRASTART gibt an, dass der CARMA-Server innerhalb von RSE mit CRASTART als Subtask gestartet werden soll. Weitere Details hierzu enthält der Abschnitt „Alternativer CARMA-Serverstart mit CRASTART (optional)“ auf Seite 57. Wenn Sie *CRASTART angeben, müssen Sie auch die crastart.*-Anweisungen angeben.
- *ISPF gibt an, dass der CARMA-Server mit dem TSO/ISPF-Client-Gateway von ISPF gestartet werden soll. Weitere Informationen erhalten Sie unter „Alternativer CARMA-Serverstart mit TSO/ISPF-Client-Gateway (optional)“ auf Seite 60.
- Alle anderen Werte definieren die Position der CLIST CRASUBMT mit Namenskonventionen wie in TSO. 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. Für diesen letztgenannten Fall müssen alle CARMA-Benutzer eine eigene CLIST CRASUBMT führen.

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

crastart.stub

z/OS UNIX-Stub zum Aufrufen von CRASTART. Die Standardeinstellung ist /usr/lpp/rdz/bin/CRASTART. Dieser Stub macht das MVS-basierte Lademodul CRASTART für z/OS UNIX-Prozesse verfügbar. Diese Anweisung wird nur verwendet, wenn für die Anweisung clist.dsname der Wert *CRASTART angegeben ist.

crastart.configuration.file

Gibt den Namen der CRASTART-Konfigurationsdatei an. Die Standardeinstellung ist /etc/rdz/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.

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:

A (All)	Alle Trace-Informationen 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.

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). Diese Anweisung wird nur verwendet, wenn für die Anweisung clist.dsname der Wert *CRASTART angegeben ist.

crastart.steplib

Die Position des Moduls CRASTART, wenn der Zugriff über die Anweisung STEPLIB in rsed.envvars erfolgt. Die Standardeinstellung ist FEK.S-FEKLPA. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn das Modul CRASTART keine Komponente von LPA oder LINKLIST sein kann. 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. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn der DD-Name TASKLIB für Ihren SCM oder RAM eine spezielle Bedeutung hat und nicht als Ersatz für STEPLIB verwendet werden kann. Diese Anweisung wird nur verwendet, wenn für die Anweisung clist.dsname der Wert *CRASTART angegeben ist.

CARMA-Serverstart mit Batchübergabe

In diesem Abschnitt ist die Konfiguration der Standardmethode von Developer for System z für das Starten eines CARMA-Servers beschrieben. Wenn Sie eine andere Startmethode verwenden, können Sie diesen Anpassungsschritt auslassen.

Developer for System z verwendet standardmäßig eine Batchübergabemethode für den CARMA-Serverstart, die nicht vom TSO/ISPF-Client-Gateway abhängt und bei der sich das Modul CRASRT nicht im LPA befinden muss. Die Methode übergibt den CARMA-Server als einen Batch-Job mit langer Laufzeit in Ihrem JES.

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in `/etc/rdz/CRASRV.properties` zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Lesen Sie hierzu die Informationen unter „RSE-Schnittstelle zu CARMA“ auf Seite 53. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Setzen Sie den Wert der Anweisung `clist.dsname` wie im folgenden Beispiel auf den Datei- und Membernamen der CLIST CRASUBMT für den CARMA-Serverstart. Weitere Informationen zu den verschiedenen Anweisungen erhalten Sie unter „RSE-Schnittstelle zu CARMA“ auf Seite 53.

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname='FEK.#CUST.CNTL(CRASUBMT)'
```

Abbildung 11. CRASRV.properties - CARMA-Start mit Batchübergabe

CRASUBMT anpassen

Passen Sie die CLIST CRASUBMT wie im folgenden Codebeispiel an. Anpassungsanweisungen finden Sie in der in CRASUBMT enthaltenen Dokumentation. Die CLIST CRASUBMT übergibt einen CARMA-Server.

CRASUBMT befindet sich in `FEK.#CUST.CNTL`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.


```

PROC 1 PORT TIMEOUT(420)
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=1024K,TIME=NOLIMIT
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//*        DD DISP=SHR,DSN=FEK.#CUST.LOAD
//CRADEF   DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG   DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS  DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//*CRARAM1 DD DISP=SHR,DSN=FEK.#CUST.CRARAM1
//*
//ISPPROF DD DISP=(NEW,DELETE,DELETE),
//        SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB,UNIT=SYSAALLDA
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPEXEC DD DISP=SHR,DSN=ISP.SISPEXEC
//SYSPROC DD DISP=SHR,DSN=ISP.SISPCLIB
//*
//CARMALOG DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ISPSTART PGM(CRASERV) PARM(&PORT &TIMEOUT)
//*
$$
EXIT CODE(0)

```

Abbildung 12. CRASUBMT - CARMA-Start mit Batchübergabe

Anmerkung:

- Sie können Ihre eigenen DD-Anweisungen und Dateiverkettungen hinzufügen, um die CARMA-TSO-Umgebung anzupassen und so eine TSO-Anmeldeprozedur zu imitieren.
- Wenn Sie möchten, können Sie das CARMA-Zeitlimit ändern. Dazu müssen Sie die Zeile PROC 1 PORT TIMEOUT(420) in der CLIST FEK.#CUST.CNTL(CRASUBMT) 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-Startvorgangs sehen Sie in der Datei rsecomm.log. Weitere Informationen zum Festlegen des Detaillierungsgrades von rsecomm.log enthält der Abschnitt „RSE-Trace (optional)“ auf Seite 96.
- Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

Alternativer CARMA-Serverstart mit CRASTART (optional)

In diesem Abschnitt ist die Konfiguration einer alternativen Methode von Developer for System z für das Starten eines CARMA-Servers beschrieben. Wenn Sie eine andere Startmethode verwenden, können Sie diesen Anpassungsschritt auslassen.

Developer for System z unterstützt eine alternative Methode für den CARMA-Serverstart, die keinen JES-Initiator verwendet, um einen Server-Job zu übergeben. Diese Methode startet den CARMA-Server mit CRASTART als Subtask innerhalb von RSE und ist mit dem TSO/ISPF-Client-Gateway-Service vergleichbar.

Anmerkung: Details des CARMA-Startvorgangs sehen Sie in der Datei `rsecomm.log`. Weitere Informationen zum Festlegen des Detaillierungsgrades von `rsecomm.log` enthält der Abschnitt „RSE-Trace (optional)“ auf Seite 96.

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in `/etc/rdz/CRASRV.properties` zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Lesen Sie hierzu die Informationen unter „RSE-Schnittstelle zu CARMA“ auf Seite 53. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Setzen Sie den Wert der Anweisung `clist.dsname` auf `*CRASTART` und geben Sie für die Anweisungen `crastart.*` die richtigen Werte an. Vergleichen Sie dazu das folgende Beispiel. Weitere Informationen zu den verschiedenen Anweisungen erhalten Sie unter „RSE-Schnittstelle zu CARMA“ auf Seite 53.

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname=*CRASTART
crastart.stub=/usr/lpp/rdz/bin/CRASTART
crastart.configuration.file=/etc/rdz/crastart.conf
crastart.syslog=Partial
crastart.timeout=420
#crastart.steplib=FEEK.SFEKLPA
#crastart.tasklib=TASKLIB
```

*Abbildung 13. CRASRV.properties - *CRASTART für alternativen CARMA-Start*

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

crastart.conf anpassen

Während dieses Anpassungsschrittes sollten Sie einen Ausdruck des angepassten Members `SCRASUBMT` als Referenz zur Hand haben. (Lesen Sie hierzu den Abschnitt „CARMA-Serverstart mit Batchübergabe“ auf Seite 56.) Auch wenn Sie das Member nicht angepasst haben, kann der Ausdruck von Vorteil sein.

`CRASTART` erstellt ausgehend von den Definitionen in `crastart.conf` eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen. Developer for System z kann in dieser Umgebung den CARMA-Server CRASERV ausführen.

Die Datei `crastart.conf` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

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

Die folgenden Anpassungsschritte sind erforderlich, um die Konfigurationsdatei wie im nachstehenden Beispiel anzupassen.

- Fügen Sie die der STEPLIB-Verkettung der Prozedur CRASUBMT zugeordneten Dateien zur Anweisung TASKLIB in crastart.conf hinzu.
- Erstellen Sie Einträge für die obligatorischen DD-Anweisungen CRADEF, CRAMSG und CRSTRS der CARMA-VSAM. Verwenden Sie die in der (angepassten) Prozedur CRASUBMT angegebenen Dateinamen.
- Fügen Sie alle angepassten DD-Anweisungen und die zugehörigen Dateiverkettungen hinzu, die in der (angepassten) Prozedur CRASUBMT verfügbar sind. Fügen Sie beispielsweise die DD-Anweisung und den Dateinamen CRARAM1 hinzu, wenn Sie den Beispiel-PDS-RAM verwenden. Sie können Namen von Dateien (die mit DISP=SHR angelegt wurden) sowie SYSOUT- und DUMMY-Konstrukte verwenden.
- Fügen Sie alle BPXWDYN-Befehle mit der Anweisung -COMMAND hinzu (optional). Es kann mehrere Anweisungen -COMMAND geben. Mit BPXWDYN können Sie sehr komplexe Zuordnungen vornehmen. Sie können beispielsweise temporäre Dateien erstellen, andere Dispositionen als SHR angeben, eine Zuordnung zu anderen Subsystemen angeben usw. Weitere Informationen zu BPXWDYN enthält die Veröffentlichung *Using REXX and z/OS UNIX System Services* (IBM Form SA22-7806).
- Wählen Sie mit der Anweisung PROGRAM die gewünschte Programmaufrufmethode aus. Empfohlen wird die Methode "PROGRAM=IKJEFT01 CRASERV &CRAPRM1. &CRAPRM2.", mit der eine TSO-Umgebung erstellt wird, die eine Mischung aus APF-Dateien und Nicht-APF-Dateien bearbeiten kann. Weitere Methoden sind in der Beispieldatei crastart.conf angegeben.

Anmerkung: Die Definitionen in crastart.conf müssen jeweils in nur einer Zeile angegeben sein.

* crastart.conf - CARMA-Zuordnungsoptionen

```
TASKLIB = FEK.SFEKLOAD
CRADEF = FEK.#CUST.CRADEF
CRAMSG = FEK.#CUST.CRAMSG
CRSTRS = FEK.#CUST.CRSTRS
*CRARAM1 = FEK.#CUST.CRARAM1
*
CARMALOG = SYSOUT(H)
SYSTSPRT = SYSOUT(H)
SYSTSIN = DUMMY
-COMMAND=ALLOC FI(SCRATCH) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80) UNIT(VIO)
*
PROGRAM=IKJEFT01 CRASERV &CRAPRM1. &CRAPRM2.
```

Abbildung 14. crastart.conf - *CRASTART für alternativen CARMA-Start

In der Konfigurationsdatei können die folgenden Variablen verwendet werden:

Tabelle 10. Variablen für crastart.conf

&CRAUSER.	Anmeldebenutzer-ID des Clients
&CRADATE.	Aktuelles Datum im (siebenstelligen julianischen) Format Djjjjddd
&CRATIME.	Aktuelle Zeit im Format Thhmmss (Stunden, Minuten und Sekunden)

Tabelle 10. Variablen für *crastart.conf* (Forts.)

&CRAPRM3. bis &CRAPRM9.	<p>Zusätzliche Variablen mit benutzerdefinierten Werten. Wenn Sie diese Variablen verwenden, muss die von <code>startup.script.name</code> in der Datei <code>CRASRV.properties</code> referenzierte CARMA-Start-REXX angepasst werden.</p> <p>Wenn Sie diese Variablen verwenden, sollten Sie eine Kopie des Standardstarts REXX (<code>/usr/lpp/rdz/bin/carma.startup.rex</code>) anpassen und mit <code>startup.script.name</code> auf diese Kopie verweisen. Damit verhindern Sie den Verlust Ihrer Arbeitsergebnisse bei Wartungsaktualisierungen für den Standard-REXX.</p>
Systemsymbol	Jedes in <code>SYS1.PARMLIB(IEASYMxx)</code> definierte Symbol
-<DD>	Ein bereits definierter DD-Name mit einem vorangestellten Bindestrich (-) fungiert in JCL als Rückbezug auf <code>*.ddname</code> . Die ursprüngliche DD muss mit der Anweisung <code>-COMMAND</code> zugeordnet werden.

Anmerkung: Für das TSO-Präfix gibt es keine Variable, weil TSO während der Interpretation der Konfigurationsdatei nicht aktiv ist.

Alternativer CARMA-Serverstart mit TSO/ISPF-Client-Gateway (optional)

In diesem Abschnitt ist die Konfiguration einer alternativen Methode von Developer for System z für das Starten eines CARMA-Servers beschrieben. Wenn Sie eine andere Startmethode verwenden, können Sie diesen Anpassungsschritt auslassen.

Developer for System z unterstützt eine alternative Methode für den CARMA-Serverstart, die keinen JES-Initiator verwendet, um einen Server-Job zu übergeben, und bei der sich das Modul CRASTART nicht im LPA befinden muss. Die Methode nutzt das TSO/ISPF-Client-Gateway von ISPF und ist mit der Standardmethode für den Zugriff auf TSO Commands Service vergleichbar.

Anmerkung: Details des CARMA-Startvorgangs sehen Sie in der Datei `rsecomm.log`. Weitere Informationen zum Festlegen des Detaillierungsgrades von `rsecomm.log` enthält der Abschnitt „RSE-Trace (optional)“ auf Seite 96.

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in `/etc/rdz/CRASRV.properties` zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Lesen Sie hierzu die Informationen unter „RSE-Schnittstelle zu CARMA“ auf Seite 53. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Setzen Sie den Wert der Anweisung `clist.dsname` wie im folgenden Beispiel auf `*ISPF`. Weitere Informationen zu den verschiedenen Anweisungen erhalten Sie unter „RSE-Schnittstelle zu CARMA“ auf Seite 53.

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname=*ISPF
```

Abbildung 15. *CRASRV.properties* - *ISPF für alternativen CARMA-Start

ISPF.conf anpassen

Während dieses Anpassungsschrittes sollten Sie einen Ausdruck des angepassten Members `SCRASUBMT` als Referenz zur Hand haben. (Lesen Sie hierzu den Abschnitt „CARMA-Serverstart mit Batchübergabe“ auf Seite 56.) Auch wenn Sie das Member nicht angepasst haben, kann der Ausdruck von Vorteil sein.

Das TSO/ISPF-Client-Gateway von ISPF erstellt ausgehend von den Definitionen in `ISPF.conf` eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen. Developer for System z führt in dieser Umgebung den CARMA-Server aus.

Die Datei `ISPF.conf` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

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

Die folgenden Anpassungsschritte sind erforderlich, um die Konfigurationsdatei wie im nachstehenden Beispiel anzupassen.

- Geben Sie für die obligatorischen ISPF-Dateien die korrekten Namen an. (Ordnen Sie jedoch nicht die Anweisung `ISPPROF` zu, da diese dynamisch zugeordnet wird.)
- Hängen Sie die `PROCLIB` von Developer for System z, `FEK.SFEKPROC`, an die Anweisung `SYSPROC` oder `SYSEXEC` an, damit das System die `CRASRVI-Exec` finden kann. Diese Exec startet den CARMA-Server (und ersetzt damit die `DD SYSTSIN` von `CRASUBMT`).
- Hängen Sie die `STEPLIB-DD`-Verkettung der Prozedur `CRASUBMT` an die Anweisung `ispllib` an.
- Erstellen Sie Einträge für die obligatorischen DD-Anweisungen `CRADef`, `CRAMSG` und `CRSTRS` der CARMA-VSAM. Verwenden Sie die in der (angepassten) Prozedur `CRASUBMT` angegebenen Dateinamen.
- Fügen Sie alle angepassten DD-Anweisungen und die zugehörigen Dateiverkettungen hinzu, die in der (angepassten) Prozedur `CRASUBMT` verfügbar sind. Fügen Sie beispielsweise die DD-Anweisung und den Dateinamen `CRARAM1` hinzu, wenn Sie den Beispiel-PDS-RAM verwenden. Sie können nur Namen von Dateien (die mit `DISP=SHR` angelegt wurden) verwenden.
- Wenn Sie weitere Zuordnungen mit einer Exec vornehmen möchten, können Sie das Kommentarzeichen vor der Anweisung `allocexec` entfernen und die Anweisung anpassen.

Anmerkung: Nehmen Sie nicht die DD-Anweisung SYSTSIN, SYSTSOUT oder CARMALOG auf und auch keine anderen DD-Anweisungen, die JES-Konstrukte wie Eingabedatenströme und SYSOUT= verwenden. Diese Einträge müssen so konvertiert werden, dass sie Dateien verwenden.

```
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
ispllib=FEK.SFEKLOAD
ispmllib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
CRADEF =FEK.#CUST.CRADEF
CRAMSG =FEK.#CUST.CRAMSG
CRASTRS=FEK.#CUST.CRASTRS
*CRARAM1=FEK.#CUST.CRARAM1
allocjob=FEK.#CUST.CNTL(CRAISPRX)
```

Abbildung 16. ISPF.conf - *ISPF für alternativen CARMA-Start

Die DD-Anweisung CARMALOG verweist standardmäßig auf den Wert SYSOUT=*, der in ISPF.conf nicht zugeordnet werden kann. Eine direkte Zuordnung der DD-Anweisung zu einer Datei ist auch nicht möglich, da alle Benutzer von Developer for System z dieselbe Datei ISPF.conf verwenden.

Sie können jedoch ausgehend von der aktiven Benutzer-ID eine Datei anlegen. Verwenden Sie dazu eine Zuordnungs-Exec wie in Kapitel 16, „TSO-Umgebung anpassen“, auf Seite 269 im Abschnitt „Erweitert – Zuordnungs-Exec verwenden“ auf Seite 271 beschrieben. Im Beispielmember CRAISPRX der Datei FEK.#CUST.CNTL sehen Sie ein Beispiel für die Zuordnung der DD-Anweisung zum Dateinamen TSOPREFIX'. 'USERID'.CRA.'TIMESTAMP'.CARMALOG'.

Anmerkung:

- Wenn Sie die Anweisung allocjob verwenden, müssen Sie darauf achten, dass Sie nicht die DD-Definitionen aufheben, die Sie bereits in ISPF.conf festgelegt haben.
- Wenn der Parameter JWT im Parmlib-Member SMFPRMxx auf einen niedrigeren Wert als der Wert ISPF_timeout in ISPF.conf gesetzt ist, müssen Sie mit einem Systemabbruch 522 für Modul CRASERV rechnen. Dies hat keine Auswirkungen auf CARMA-Operationen, da der Server automatisch erneut gestartet wird, falls dies erforderlich ist.

Beispiel-RAM (Repository Access Manager) aktivieren (optional)

Repository Access Managers (RAM) sind vom Benutzer geschriebene APIs für die Anbindung an z/OS Software Configuration Manager (SCM). Führen Sie für die Beispiel-RAM, die Sie aktivieren möchten, die Anweisungen in den folgenden Abschnitten aus.

Anmerkung: Die Beispiel-RAM werden zum Testen der Konfiguration Ihrer CARMA-Umgebung und für die Entwicklung Ihrer eigenen RAM bereitgestellt. Verwenden Sie die zur Verfügung gestellten Beispiel-RAM NICHT in einer Produktionsumgebung.

Weitere Informationen zu den bereitgestellten Beispiel-RAM und zum bereitgestellten Beispielquellcode finden Sie im *Rational Developer for System z Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).

Die unten angegebenen Beispielmuster befinden sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP (FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

PDS-RAM aktivieren

Der PDS-RAM gibt eine Dateiliste ähnlich zu **MVS-Dateien** -> **Meine Dateien** in der Ansicht **Ferne Systeme** zurück. Der PDS-RAM verwendet standardmäßig die RAM-ID '0'.

Anmerkung: Der PDS-RAM setzt voraus, dass CARMA (mithilfe von ISPSTART) in ISPF gestartet wird.

1. Passen Sie die JCL FEK.#CUST.JCL(CRA#VPDS) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA#VPDS enthaltenen Dokumentation. CRA#VPDS erstellt die VSAM-Datei für PDS-RAM-Nachrichten und setzt die erforderlichen Daten ein.
2. Fügen Sie die DD-Anweisung CRARAM1 zur ausgewählten CARMA-Startmethode hinzu und geben Sie den Namen der Datei mit der VSAM für PDS-RAM-Nachrichten an.

SCLM-RAM aktivieren

Der SCLM-RAM gibt einen Basiseintrag in SCLM (Software Configuration Manager von ISPF) zurück. Der SCLM-RAM verwendet standardmäßig die RAM-ID '1'.

Anmerkung: Der SCLM-RAM setzt voraus, dass CARMA (mithilfe von ISPSTART) in ISPF gestartet wird.

1. Passen Sie die JCL FEK.#CUST.JCL(CRA#VSLM) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA#VSLM enthaltenen Dokumentation. CRA#VSLM erstellt die VSAM-Datei für SCLM-RAM-Nachrichten und setzt die erforderlichen Daten ein.
2. Fügen Sie die DD-Anweisung CRARAM2 zur ausgewählten CARMA-Startmethode hinzu und geben Sie den Namen der Datei mit der VSAM für SCLM-RAM-Nachrichten an.
3. Passen Sie die JCL FEK.#CUST.JCL(CRA#ASLM) an. Anpassungsanweisungen finden Sie in der in CRA#ASLM enthaltenen Dokumentation. CRA#ASLM legt die für SCLM-RAM-Clients erforderlichen Dateien an.

Anmerkung: Vor der Verwendung von CARMA mit dem SCLM-RAM muss jeder Benutzer FEK.#CUST.JCL(CRA#ASLM) einmal übergeben. Andernfalls kommt es zu einem Zuordnungsfehler.

Skeleton-RAM aktivieren

Der Skeleton-RAM gibt ein Skeleton-Gerüst zurück, das für die Entwicklung Ihrer eigenen RAMs verwendet werden kann. Der Skeleton-RAM verwendet standardmäßig die RAM-ID '3'.

1. Passen Sie die JCL FEK.#CUST.JCL(CRA#CRAM) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA#CRAM enthaltenen Dokumentation. CRA#CRAM kompiliert den Skeleton-RAM.
2. Fügen Sie die Ladebibliothek mit dem kompilierten Skeleton-RAM CRARAMSA zur DD-Anweisung STEPLIB der ausgewählten CARMA-Startmethode hinzu (DD TASKLIB für die Methode CRASTART).

CA Endeavor® SCM-RAM aktivieren (optional)

Die Schnittstelle für CA Endeavor® Software Configuration Manager in IBM® Rational® Developer for System z ermöglicht Clients mit Developer for System z direkten Zugriff auf CA Endeavor® SCM. In diesem Handbuch wird die Schnittstelle für CA Endeavor® in IBM® Rational® Developer for System z mit 'CA Endeavor® SCM-RAM' (Repository Access Manager) abgekürzt.

Im Gegensatz zu den Beispiel-RAMs, die in dieser Veröffentlichung dokumentiert werden, ist CA Endeavor® SCM RAM ein Produktions-RAM. Sie sollten nicht beide RAM-Typen in derselben Konfiguration aktivieren.

Achtung: Die zur Verfügung gestellten Konfigurationsjobs für den CA Endeavor® SCM-RAM ersetzen die aktive CARMA-Konfiguration durch eine Konfiguration, die nur den CA Endeavor® SCM-RAM enthält.

Anmerkung: Die Startmethode für das TSO/ISPF-Client-Gateway kann nicht zusammen mit dem CA Endeavor® SCM-RAM verwendet werden.

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-Portbereich für interne Kommunikation
- Sicherheitsregel, die Benutzern die Übergabe von CRA*-Jobs erlaubt (optional)
- Aktualisierung des LPA (optional)

Vor der Verwendung des CA Endeavor® SCM-RAMs an Ihrem Standort müssen Sie die folgenden Tasks ausführen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Ordnen Sie CARMA VSAM-Dateien zu, die den CA Endeavor® SCM-RAM definieren, und bereiten Sie diese vor. Weitere Details enthält der Abschnitt „CA Endeavor® SCM-RAM definieren“ auf Seite 65.
2. Wählen Sie Ihre bevorzugte Startmethode (Batchübergabe oder CRASTART) aus und führen Sie die erforderliche Anpassung der zugehörigen Konfigurationsdateien durch. Weitere Details enthalten die folgenden Abschnitte:
 - „CA Endeavor® SCM-RAM-Start mit Batchübergabe“ auf Seite 65
 - „CA Endeavor® SCM-RAM-Start mit CRASTART“ auf Seite 68
3. (Optional) Passen Sie die Zuordnungs-Exec an, die für die dynamische Zuordnung von benutzerspezifischen Dateien verwendet wird. Weitere Details enthält der Abschnitt „(Optional) CRANDVRA anpassen“ auf Seite 69.
4. (Optional) Passen Sie die CA Endeavor® SCM-RAM-spezifischen Konfigurationsdateien an. Weitere Details enthält der Abschnitt „(Optional) CA Endeavor® SCM-RAM anpassen“ auf Seite 70.

CA Endevor® SCM-RAM definieren

Die folgenden CARMA-Komponenten müssen unabhängig von der gewählten Startmethode angepasst werden. Die unten angegebenen Beispielmuster befinden sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

1. Passen Sie die JCL FEK.#CUST.JCL(CRA#VCAD) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA\$VDEF enthaltenen Dokumentation. CRA#VCAD erstellt die VSAM-Datei mit der CARMA-Konfiguration CRADEF und setzt die erforderlichen Daten ein.
2. Passen Sie die JCL FEK.#CUST.JCL(CRA\$VMSG) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA\$VMSG enthaltenen Dokumentation. CRA\$VMSG erstellt die VSAM-Datei für CARMA-Nachrichten CRAMSG und setzt die erforderlichen Daten ein.

Anmerkung: Dies ist derselbe Job wie für die Beispiel-RAMs.

3. Passen Sie die JCL FEK.#CUST.JCL(CRA#VCAS) an und übergeben Sie sie. Anpassungsanweisungen finden Sie in der in CRA\$VSTR enthaltenen Dokumentation. CRA#VCAS erstellt die VSAM-Datei mit benutzerdefinierten CARMA-Informationen CRASTRS und setzt die erforderlichen Daten ein.

Anmerkung:

- Der CA Endevor® SCM-RAM verwendet standardmäßig die RAM-ID '0'.
- Richten Sie sich nach dem Beispieljob FEK.#CUST.JCL(CRA#UADD), wenn Sie die Definitionen für einen (benutzerdefinierten) RAM mit einer vorhandenen VSAM-Konfiguration zusammenführen müssen. Dieser Job muss für jede geänderte CARMA-VSAM angepasst und übergeben werden. Weitere Informationen zur Satzstruktur, die von den verschiedenen CARMA-VSAMs verwendet werden, finden Sie im *Rational Developer for System z Common Access Repository Manager Developer's Guide* (IBM Form SC23-7660).
- Verwenden Sie den Beispieljob FEK.#CUST.JCL(CRA#UQRY), um die aktiven Definitionen aus einer VSAM in eine sequenzielle Datei zu extrahieren.

CA Endevor® SCM-RAM-Start mit Batchübergabe

Führen Sie diesen Schritt nicht aus, wenn Sie die Methode 'CRASTART' zum Starten des CARMA-Servers mit dem CA Endevor® SCM-RAM verwenden.

Developer for System z kann die Batchübergabemethode für den CARMA-Serverstart verwenden, um den CA Endevor® SCM-RAM zu starten. Die Methode übergibt den CARMA-Server als einen Batch-Job mit langer Laufzeit in Ihrem JES.

Weitere Informationen zur Startmethode mit Batchübergabe erhalten Sie unter „CARMA-Serverstart mit Batchübergabe“ auf Seite 56.

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in `/etc/rdz/CRASRV.properties` zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Lesen Sie hierzu die Informationen unter „RSE-Schnittstelle zu CARMA“ auf Seite 53. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Setzen Sie den Wert für die Anweisung `clist.dsname` wie im folgenden Beispiel auf den Datei- und Membernamen der CLIST CRASUBCA für den CARMA-Serverstart. Weitere Informationen zu den verschiedenen Anweisungen erhalten Sie unter „RSE-Schnittstelle zu CARMA“ auf Seite 53.

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname='FEK.#CUST.CNTL(CRASUBCA)'
```

Abbildung 17. Abbildung x1. CRASRV.properties - CA Endevor® SCM-RAM-Start mit Batchübergabe

CRASUBCA anpassen

Passen Sie die CLIST CRASUBCA wie im folgenden Codebeispiel an. Anpassungsanweisungen finden Sie in der in CRASUBCA enthaltenen Dokumentation. Die CLIST CRASUBCA übergibt einen CARMA-Server für CA Endevor® SCM.

CRASUBCA befindet sich in `FEK.#CUST.CNTL`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

```

PROC 1 PORT TIMEOUT(420)
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT,
//          PARM='%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&PORT &TIMEOUT)'
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//          DD DISP=SHR,DSN=CA.NDVR.AUTHLIB
//          DD DISP=SHR,DSN=CA.NDVU.AUTHLIB
//CRADEF DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//*
//SYSPROC DD DISP=SHR,DSN=ISP.SISPCLIB
//          DD DISP=SHR,DSN=FEK.SFEKPROC
//ISPEXEC DD DISP=SHR,DSN=ISP.SISPEXEC
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DUMMY
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CONLIB
//JCLOUT DD SYSOUT=(A,INTRDR),DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
//EXT1ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT1DEP DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//MSG3FILE DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//C1MSGSG1 DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//C1EXMSGSG DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//TYPEMAP DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRASHOW)
$$
EXIT CODE(0)

```

Abbildung 18. Abbildung x2. CRASUBCA - CA Endevor® SCM-RAM-Start mit Batchübergabe

Anmerkung:

- Sie können Ihre eigenen DD-Anweisungen und Dateiverkettungen hinzufügen, um die CARMA-TSO-Umgebung anzupassen und so eine TSO-Anmeldeprozedur zu imitieren.
- (Optional) Sie können den Wert für das CARMA-Zeitlimit ändern, indem Sie die Zeile PROC 1 PORT TIMEOUT(420) in der CLIST 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-Startvorgangs sehen Sie in der Datei rsecomm.log. Weitere Informationen zum Festlegen des Detaillierungsgrades von rsecomm.log enthält der Abschnitt „RSE-Trace (optional)“ auf Seite 96.
- Änderungen werden für alle CARMA-Server wirksam, die nach der Aktualisierung gestartet werden.

CA Endeavor® SCM-RAM-Start mit CRASTART

Führen Sie diesen Schritt nicht aus, wenn Sie die Batchübergabemethode zum Starten des CARMA-Servers mit dem CA Endeavor® SCM-RAM verwenden.

Developer for System z kann die Methode 'CRASTART' für den CARMA-Serverstart verwenden, um den CA Endeavor® SCM-RAM zu starten. Diese Methode startet den CARMA-Server mit CRASTART als Subtask innerhalb von RSE.

Weitere Informationen zur Startmethode mit CRASTART erhalten Sie unter „Alternativer CARMA-Serverstart mit CRASTART (optional)“ auf Seite 57.

Anmerkung: Details des CARMA-Startvorgangs sehen Sie in der Datei rsecomm.log. Weitere Informationen zum Festlegen des Detaillierungsgrades von rsecomm.log enthält der Abschnitt „RSE-Trace (optional)“ auf Seite 96.

CRASRV.properties anpassen

Der RSE-Server verwendet die Einstellungen in /etc/rdz/CRASRV.properties zum Starten eines CARMA-Servers und für den Zugriff auf diesen Server. Lesen Sie hierzu die Informationen unter „RSE-Schnittstelle zu CARMA“ auf Seite 53. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Setzen Sie den Wert der Anweisung `clist.dsname` auf `*CRASTART` und geben Sie für die Anweisungen `crastart.*` die richtigen Werte an. Vergleichen Sie dazu das folgende Beispiel. Weitere Informationen zu den verschiedenen Anweisungen erhalten Sie unter „RSE-Schnittstelle zu CARMA“ auf Seite 53.

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname=*CRASTART
crastart.stub=/usr/lpp/rdz/bin/CRASTART
crastart.configuration.file=/etc/rdz/crastart.endevor.conf
crastart.syslog=Partial
crastart.timeout=420
#crastart.steplib=FEK.SFEKLPA
#crastart.tasklib=TASKLIB
```

Abbildung 19. Abbildung x3. CRASRV.properties - CA Endeavor® SCM-RAM-Start mit CRASTART

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

crastart.endevor.conf anpassen

CRASTART erstellt ausgehend von den Definitionen in `crastart.endevor.conf` eine gültige (TSO/ISPF-)Umgebung, um CA Endeavor® SCM aufzurufen. Developer for System z führt in dieser Umgebung den CA Endeavor® SCM-RAM aus.

`crastart.endevor.conf` befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

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

```

TASKLIB = FEK.SFEKLOAD,CA.NDVR.AUTHLIB,CA.NDVRU.AUTHLIB
CRADEF  = FEK.#CUST.CRADEF
CRAMSG  = FEK.#CUST.CRAMSG
CRASTRS = FEK.#CUST.CRASTRS

SYSPROC = ISP.SISPCLIB,FEK.SFEKPROC
SYSEXEC = ISP.SISPEXEC
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
BLKSIZE(6160) SPACE(2,2) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
BLKSIZE(6160) SPACE(2,2) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) DIR(5) RECFM(F,B) LRECL(80)
BLKSIZE(6160) SPACE(2,2) TRACKS UNIT(SYSALLDA)
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPTABL = -ISPPROF

CARMALOG = SYSOUT(H)
SYSPRINT= SYSOUT(H)
SYSTSPRT = SYSOUT(H)
SYSTSIN  = DUMMY

TYPEMAP = FEK.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEK.#CUST.PARMLIB(CRASHOW)
CONLIB   = CA.NDVR.CONLIB
-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(EXT1DEP) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(MSG3FILE) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(C1EXMSG5) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(C1MSG51) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)

PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1.
&CRAPRM2.)

```

Abbildung 20. crastart.conf - CA Endeavor® SCM-RAM-Start mit CRASTART

(Optional) CRANDVRA anpassen

Die Startmethoden mit Batchübergabe und mit CRASTART rufen beide die REXX-Exec CRANDVRA auf, um benutzerspezifische Dateien zuzuordnen, die vom CA Endeavor® SCM-RAM verwendet werden.

DD-Anweisung	Dateiname	Typ
DEPEND	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.DEPEND	Permanent
BROWSE	&SYSPREF..&SYSUID..&SYSNAME..CRA\$NDVR OWSE	Temporär
C1PRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	Temporär

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

Anpassungsanweisungen finden Sie in der in CRANDVRA enthaltenen Dokumentation.

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

(Optional) CA Endevor® SCM-RAM anpassen

Die folgenden CARMA-Komponenten können unabhängig von der gewählten Startmethode angepasst werden. Die unten angegebenen Beispielmuster befinden sich in FEK.#CUST.PARMLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

1. (Optional) Passen Sie FEK.#CUST.PARMLIB(CRASHOW) an. Anpassungsanweisungen finden Sie in der in CRASHOW enthaltenen Dokumentation. CRASHOW definiert Standardfilter für CA Endevor® SCM-Umgebungen, -Systeme usw.
2. (Optional) Passen Sie FEK.#CUST.PARMLIB(CRATMAP) an. Anpassungsanweisungen finden Sie in der in CRATMAP enthaltenen Dokumentation. CRATMAP überschreibt den CA Endevor® SCM-Typ mit Dateierweiterungszuordnungen.

(Optional) Unterstützung mehrerer RAMs

CARMA lässt die Definition und gleichzeitige Ausführung mehrerer RAMs zu. Da jedoch auch bei mehreren RAMs nur ein CARMA-Server pro Benutzer aktiv ist, müssen möglicherweise einige Änderungen an der Konfiguration vorgenommen werden, damit sie funktioniert.

RAMs 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 RAMs und übergibt die Informationen an den CARMA-Client. Der Benutzer kann dann einen oder mehrere RAMs auswählen, die in den CARMA-Server geladen werden.

Da RAMs als Plug-ins des CARMA-Servers aktiv sind, müssen Sie sicherstellen, dass alle Voraussetzungen (wie Dateizuordnungen) für jeden der RAMs im Adressraum des CARMA-Servers verfügbar sind. Dies kann Änderungen an den CARMA-Konfigurationsbeispielen, wie CRASUBMT oder crastart.conf, erfordern, die im Lieferumfang von Developer for System z enthalten sind.

Beispiel

Im folgenden Beispiel wird eine vorhandene Konfiguration des CA Endevor® SCM-RAMs mit der Startmethode 'CRASTART' gestartet und anschließend der Beispiel-PDS-RAM hinzugefügt.

Definitionen für den CA Endevor® SCM-RAM:

- FEK.SFEKVSM2(CRA0VCAD) - CRADEF-Definitionen
- FEK.SFEKVSM2(CRA0VCAS) - CRASTRS-Definitionen
- /etc/rdz/crastart.endevor.conf - CRASTART-Konfigurationsdatei

Definitionen für den PDS-RAM:

- FEK.SFEKVSM2(CRA0VDEF) - CRADEF-Definitionen
- FEK.SFEKVSM2(CRA0VSTR) - CRASTRS-Definitionen
- FEK.#CUST.CRARAM1 - CRARAM1-Definitionen

Als erstes sammelt der RAM-Entwickler alle Daten und Informationen, die der Systemprogrammierer benötigt, um die Konfiguration abzuschließen.

1. Extrahieren Sie die für den PDS-RAM spezifischen Daten aus den SFEKVSM2-Membren (diese Member enthalten Definitionen für alle Beispiel-RAMs, nicht nur für den PDS-RAM).
2. Führen Sie diese Daten mit den SFEKVSM2-Membren des Endevor® SCM-RAMs zusammen.
3. Erstellen Sie eine Liste mit den Voraussetzungen für den PDS-RAM:
 - DD-Anweisung 'CRARAM1', verweist auf FEK.#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 RAMs 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.
2. Fügen Sie crastart.endevor.conf eine CRARAM1-Definition hinzu:

```
CRARAM1 = FEK.#CUST.CRARAM1
```

3. Überprüfen Sie die PROGRAM-Anweisung in crastart.endevor.conf, um sicherzustellen, dass sie die Umgebung bereitstellen kann, die von beiden RAMs benötigt wird:

```
PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV)
      PARM(&CRAPRM1. &CRAPRM2.)
```

- IKJEFT01: TSO wird verwendet, um bestimmte autorisierte Aufrufe in einer nicht-autorisierten Umgebung zuzulassen. Des Weiteren wird TSO als Umgebung für die Ausführung der CA Endevor® SCM-RAM-Vorzuordnungs-Exec verwendet.
- %CRANDVRA: Vorzuordnungs-Exec des CA Endevor® SCM-RAMs (befindet sich in FEK.SFEKPROC), die temporäre (und permanente) benutzerspezifische Arbeitsdateien zuordnet
- NDVRC1: CA Endevor®-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 Endevor® SCM-RAM ist in einer ISPF-Umgebung aktiv, die voraussetzt, dass die vom PDS-RAM benötigte TSO-Umgebung ebenfalls verfügbar ist.

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-Anweisung SYSTSIN oder SYSTSPRT dynamisch zuordnen. Zur Umgehung dieses Problems wird das Beispielprogramm CRAXJCL bereitgestellt.

Ein Versuch Ihres Prozessors, die (für IRXJCL erforderliche) Anweisung SYSTSIN oder SYSTSPRT zuzuordnen, könnte scheitern, 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.

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. Weitere Informationen erhalten Sie in *TSO/E REXX Reference* (IBM Form SA22-7790). SYSTSIN kann somit sicher von CARMA verwendet werden. Alle Ausgaben, die IRXJCL an SYSTSPRT sendet, erscheinen im CARMA-Protokoll.

Prozessoren, die das Ersatzmodul CRAXJCL aufrufen, sollten nicht versuchen, die DD-Anweisung SYSTSIN oder SYSTSPRT vor dem Aufruf von CRAXJCL zuzuordnen.

CRAXJCL erstellen

Das Ersatzmodul CRAXJCL wird im Quellenformat bereitgestellt, denn Sie müssen es an die spezifischen Zuordnungen anpassen, die Sie für SYSTSPRT verwenden möchten. SYSTSIN wird in der Regel einer Pseudodatei (DUMMY) zugeordnet.

Der Assemblerbeispiel Quellcode und der Beispieljob für Kompilierung/Bindung werden als FEK.#CUST.ASM(CRAXJCL) und FEK.#CUST.JCL(CRA#CIRX) bereitgestellt, sofern Sie beim Anpassen und Übergeben des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 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. Anpassungsanweisungen finden Sie in der in CRA#CIRX enthaltenen Dokumentation.

Kapitel 4. Application Deployment Manager (optional)

Developer for System z verwendet bestimmte Funktionen von Application Deployment Manager als allgemeine Deploymentmethode für verschiedene Komponenten. Die in diesem Kapitel aufgelisteten Anpassungsschritte sind erforderlich, wenn Sie Entwickler sind und die folgenden Funktionen komplett oder teilweise nutzen:

- Enterprise Service Tools (EST)
- BMS Screen Designer
- MFS Screen Designer
- CICSTS-Codegenerierung

Anmerkung: Die Enterprise Service Tools (EST) umfassen mehrere Tools, zu denen SFM (Service Flow Modeler) und XSE (XML Services for the Enterprise) gehören.

Durch die Anpassung von Application Deployment Manager wird der CRD-Server (CICS Resource Definition) hinzugefügt, der als CICS-Anwendung unter z/OS ausgeführt wird, um die folgenden Funktionen zu unterstützen:

- CICS-Ressourcenabfragen
- CRD-Installationsanforderungen und -Deinstallationsanforderungen in Umgebungen mit und ohne CICSplex SM
- Anforderungen nach einer schrittweisen Aktualisierung von Programmen und Maskengruppen
- Prüfanforderungen für Pipelines
- Export-, Import- und Aktualisierungsanforderungen für Manifeste

Kapitel 15, „CICSTS-Aspekte“, auf Seite 257 enthält weitere Informationen für CICS-Administratoren zum CRD-Server.

Voraussetzungen und Prüfliste

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

- TCP/IP-Port für externe Kommunikation
- JCL für die CICS-Region aktualisieren
- CSD für die CICS-Region aktualisieren
- Gruppe für CICS-Region definieren
- Sicherheitsregel für die Aktualisierung einer ADM-VSAM durch Administratoren
- CICSTS-Sicherheit konfigurieren
- CICS-Transaktionsname definieren (optional)
- Sicherheitsregel für die Aktualisierung einer ADM-VSAM durch Benutzer (optional)

Vor der Verwendung von Application Deployment Manager an Ihrem Standort müssen Sie die folgenden Tasks ausführen. Sofern nicht anders angegeben, sind alle Tasks obligatorisch.

1. Erstellen Sie das CRD-Repository. Weitere Details enthält der Abschnitt „CRD-Repository“.
2. Wählen Sie die zu verwendende CICS-Schnittstelle (RESTful oder Web-Service) aus. (Die Schnittstellen können gemeinsam verwendet werden.) Weitere Details enthält der Abschnitt „RESTful oder Web-Service“ auf Seite 75.
3. Führen Sie, falls gewünscht, die spezifischen RESTful-Anpassungen durch. Weitere Details enthält der Abschnitt „CRD-Server mit der RESTful-Schnittstelle“ auf Seite 75.
 - Definieren Sie den CRD-Server für die primäre CICS-Verbindungsregion.
 - Definieren Sie den CRD-Server für die nicht primäre CICS-Verbindungsregion (optional).
 - Passen Sie die CRD-Servertransaktions-IDs an (optional).
4. Führen Sie, falls gewünscht, die spezifischen Web-Service-Anpassungen durch. Weitere Details enthält der Abschnitt „CRD-Server mit der Web-Service-Schnittstelle“ auf Seite 77.
 - Fügen Sie der CICS-RPL-Verknüpfung den (möglicherweise angepassten) Pipelinennachrichtenhandler hinzu.
 - Definieren Sie den CRD-Server für die primäre CICS-Verbindungsregion.
 - Definieren Sie den CRD-Server für die nicht primäre CICS-Verbindungsregion (optional).
5. Erstellen Sie das Manifestrepository (optional). Weitere Details enthält der Abschnitt „Manifestrepository (optional)“ auf Seite 79.

CRD-Repository

Passen Sie den Job ADNVCRD an und übergeben Sie ihn, um die VSAM-Datei für das CRD-Repository anzulegen und zu initialisieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.

ADNVCRD ist in FEK.#CUST.JCL enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Sie sollten für jede primäre CICS-Verbindungsregion ein gesondertes Repository erstellen. Eine gemeinsame Nutzung des Repositories impliziert, dass alle zugehörigen CICS-Regionen dieselben im Repository gespeicherten Werte verwenden.

Anmerkung:

- Um die Unterstützung von URIMAP zu aktivieren, muss ein vorhandenes CRD-Server-Repository vergrößert werden. Die Unterstützung von URIMAP wurde dem Verwaltungsdienstprogramm in Developer for System z Version 7.6.1 hinzugefügt. Weitere Details hierzu enthält der Abschnitt „Migrationshinweise zum Verwaltungsdienstprogramm“ auf Seite 266.
- Sofern Sie keine anders lautende Benachrichtigung erhalten, können Sie Ihr aktuelles CRD-Server-Repository (mit Ihren angepassten Werten) für alle Releases von Developer for System z wiederverwenden.

Benutzer müssen das Zugriffsrecht READ für das CRD-Repository haben und CICS-Administratoren das Zugriffsrecht UPDATE.

CICS-Verwaltungsdienstprogramm

Mit dem von Developer for System z bereitgestellten Verwaltungsdienstprogramm können CICS-Administratoren die Standardwerte für CICS-Ressourcendefinitionen vorgeben. Diese Standardwerte können schreibgeschützt oder für den Anwendungsentwickler editierbar sein.

Das Verwaltungsdienstprogramm wird vom Beispieljob ADNJSAPU aufgerufen. Für die Verwendung dieses Dienstprogramms ist das Zugriffsrecht UPDATE für das CRD-Repository erforderlich.

ADNJSAPU befindet sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Weitere Informationen hierzu finden Sie in Kapitel 15, „CICSTS-Aspekte“, auf Seite 257.

RESTful oder Web-Service

CICS Transaction Server stellt ab Version 4.1 Unterstützung für eine HTTP-Schnittstelle zur Verfügung, die mithilfe von RESTful-Prinzipien (Representational State Transfer) entworfen wurde. Diese RESTful-Schnittstelle ist jetzt die strategische CICSTS-Schnittstelle, die von Clientanwendungen verwendet wird. Die ältere Web-Service-Schnittstelle wurde eingefroren. Erweiterungen werden nur für die RESTful-Schnittstelle entwickelt.

Application Deployment Manager hält diese Absichtserklärung ein. Für alle Services, die ab Developer for System z Version 7.6 neu sind, ist der RESTful-CRD-Server erforderlich.

Falls gewünscht, können die RESTful- und Web-Service-Schnittstellen gleichzeitig in einer CICS-Region aktiv sein. In diesem Fall sind in der Region zwei CRD-Server aktiv. Beide Server verwenden gemeinsam dasselbe CRD-Repository. Beachten Sie, dass CICS einige Warnungen zu doppelten Definitionen ausgibt, wenn die zweite Schnittstelle in der Region definiert wird.

CRD-Server mit der RESTful-Schnittstelle

Die Informationen in diesem Abschnitt beschreiben, wie Sie den CRD-Server definieren, der die RESTful-Schnittstelle für die Kommunikation mit dem Client von Developer for System z verwendet.

Falls gewünscht, können die RESTful- und Web-Service-Schnittstellen gleichzeitig in einer CICS-Region aktiv sein. In diesem Fall sind in der Region zwei CRD-Server aktiv. Beide Server verwenden gemeinsam dasselbe CRD-Repository. Beachten Sie, dass CICS einige Warnungen zu doppelten Definitionen ausgibt, wenn die zweite Schnittstelle in der Region definiert wird.

Primäre CICS-Verbindungsregion

Der CRD-Server muss für die primäre Verbindungsregion definiert werden. Diese WOR (Web Owning Region) verarbeitet die Web-Service-Anforderungen von Developer for System z.

- Stellen Sie die Lademodule FEK.SFEKLOAD(ADNCRD*, ADNANAL und ADNREST) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.
- Passen Sie den Job ADNCSDRS an und übergeben Sie ihn, um die CICS-Systemdefinition (CSD) für die primäre CICS-Verbindungsregion zu aktualisieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.
ADNCSDRS ist in FEK.#CUST.JCL enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.
- Installieren Sie die ADM-Gruppe für diese Region mit dem entsprechenden CEDA-Befehl. Beispiel:
CEDA INSTALL GROUP(ADNPCRGP)

Nicht primäre CICS-Verbindungsregionen

Der CRD-Server kann auch mit zusätzlichen, nicht primären Verbindungsregionen verwendet werden. Dabei handelt es sich in der Regel um AOR-Regionen (Application Owning Regions).

Anmerkung: Sie müssen diese Schritte nicht ausführen, wenn Ihre CICS-Ressourcendefinitionen mit BAS (Business Application Services) von CICSplex SM verwaltet werden.

- Stellen Sie das ADM-Lademodul FEK.SFEKLOAD(ADNCRD*) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) dieser nicht primären Verbindungsregionen. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.
- Passen Sie den Job ADNCS DAR an und übergeben Sie ihn, um CSD für diese nicht primären Verbindungsregionen zu aktualisieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.
ADNCS DAR ist in FEK.#CUST.JCL enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.
- Installieren Sie die ADM-Gruppe für diese Regionen mit dem entsprechenden CEDA-Befehl. Beispiel:
CEDA INSTALL GROUP(ADNARRGP)

CRD-Servertransaktions-IDs anpassen (optional)

Developer for System z stellt mehrere Transaktionen bereit, die der CRD-Server beim Definieren und Abfragen von CICS-Ressourcen verwendet.

Tabelle 11. Standard-Transaktions-IDs des CRD-Servers

Transaktion	Beschreibung
ADMS	Für Änderungen an CICS-Ressourcen, die vom Manifestverarbeitungstool angefordert werden. Diese Transaktion ist normalerweise für CICS-Administratoren bestimmt.
ADMI	Für Anforderungen, die CICS-Ressourcen definieren, installieren oder deinstallieren.
ADMR	Für alle anderen Anforderungen, die CICS-Umgebungsinformationen oder -Ressourceninformationen abrufen.

Indem Sie die folgenden Schritte ausführen, können Sie die Transaktions-IDs ändern, damit sie mit den Standardwerten Ihres Standorts übereinstimmen:

1. Passen Sie den Job ADNTXNC an und übergeben Sie ihn, um das Lademodul ADNRCUST zu erstellen. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.
2. Stellen Sie das resultierende Lademodul ADNRCUST in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der CICS-Verbindungsregion, in der der CRD-Server definiert ist.
3. Passen Sie den Job ADNCSCTX an und übergeben Sie ihn, um ADNRCUST als Programm in den CICS-Regionen zu definieren, in denen der CRD-Server definiert ist. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.

Anmerkung: Der RESTful-CRD-Server versucht immer, das Lademodul ADNRCUST zu laden. Sie erzielen daher einen kleinen Leistungsvorteil, wenn Sie das Lademodul ADNRCUST erstellen und definieren, auch wenn Sie die Transaktions-IDs nicht ändern.

CRD-Server mit der Web-Service-Schnittstelle

Die Informationen in diesem Abschnitt beschreiben, wie Sie den CRD-Server definieren, der die Web-Service-Schnittstelle für die Kommunikation mit dem Client von Developer for System z verwendet.

Falls gewünscht, können die RESTful- und Web-Service-Schnittstellen gleichzeitig in einer CICS-Region aktiv sein. In diesem Fall sind in der Region zwei CRD-Server aktiv. Beide Server verwenden gemeinsam dasselbe CRD-Repository. Beachten Sie, dass CICS einige Warnungen zu doppelten Definitionen ausgibt, wenn die zweite Schnittstelle in der Region definiert wird.

Handler für Pipelinenachrichten

Der Pipelinenachrichtenhandler (ADNTMSGH) wird für die Sicherheit verwendet. Er verarbeitet die Benutzer-ID und das Kennwort im SOAP-Header. ADNTMSGH wird von der Beispielpipelinekonfigurationsdatei referenziert und muss deshalb in die CICS-RPL-Kette gestellt werden. Weitere Informationen zum Handler für Pipelinenachrichten und die erforderliche Sicherheitskonfiguration enthält Kapitel 15, „CICSTS-Aspekte“, auf Seite 257.

Developer for System z stellt mehrere Transaktionen bereit, die der CRD-Server beim Definieren und Abfragen von CICS-Ressourcen verwendet. Die Transaktions-IDs legt ADNTMSGH je nach erforderlicher Operation fest. Für standortspezifische Anpassungen von ADNTMSGH steht COBOL-Beispielquellcode zur Verfügung.

Tabelle 12. Standard-Transaktions-IDs des CRD-Servers

Transaktion	Beschreibung
ADMS	Für Änderungen an CICS-Ressourcen, die vom Manifestverarbeitungstool angefordert werden. Diese Transaktion ist normalerweise für CICS-Administratoren bestimmt.
ADMI	Für Anforderungen, die CICS-Ressourcen definieren, installieren oder deinstallieren.
ADMR	Für alle anderen Anforderungen, die CICS-Umgebungsinformationen oder -Ressourceninformationen abrufen.

Standardmodul verwenden:

- Stellen Sie das Lademodul FEK.SFEKLOAD(ADNTMSGH) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.

ADNTMSGH anpassen:

Die Beispielmembers ADNMSGH* befinden sich in FEK.#CUST.JCL und FEK.#CUST.COBO, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

- Passen Sie den Beispielquellcode (COBOL) FEK.#CUST.COBO(ADNMSGHS) des Pipelinenachrichtenhandlers an die Standards Ihres Standorts an.
- Passen Sie den Job FEK.#CUST.JCL(ADNMSGHC) an und übergeben Sie ihn, um die angepasste Quelle ADNMSGHS zu kompilieren. Anpassungsanweisungen finden Sie in der in ADNMSGHC enthaltenen Dokumentation. Beachten Sie, dass das resultierende Lademodul den Namen ADNTMSGH haben muss.
- Stellen Sie das resultierende Lademodul ADNTMSGH in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion.

Anmerkung: Stellen Sie sicher, dass das angepasste Lademodul ADNTMSGH vor allen Verweisen auf FEK.SFEKLOAD angegeben ist, da andernfalls das Standardmodul verwendet wird.

CICS, primäre Verbindungsregion

Der CRD-Server muss für die primäre Verbindungsregion definiert werden. Diese Region verarbeitet die Serviceanforderungen von Developer for System z.

- Stellen Sie die Lademodule FEK.SFEKLOAD(ADNCRD*, ADNANAL und ADNREST in die CICS-RPL-Kette (DD-Anweisung DFHRPL) der primären CICS-Verbindungsregion. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind. Das Lademodul des Pipelinenachrichtenhandlers ADNTMSGH muss ebenfalls in der RPL-Verkettung enthalten sein. Lesen Sie hierzu die Informationen unter „Handler für Pipelinenachrichten“ auf Seite 77.
- Passen Sie den Job ADNCSDWS an und übergeben Sie ihn, um die CICS-Systemdefinition (CSD) für die primäre CICS-Verbindungsregion zu aktualisieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation. Die in diesem Job verwendeten Transaktions-IDs müssen mit den vom (möglicherweise angepassten) Pipelinenachrichtenhandler verwendeten übereinstimmen.

ADNCSDWS ist in FEK.#CUST.JCL enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

- Installieren Sie die ADM-Gruppe für diese Region mit dem entsprechenden CEDA-Befehl. Beispiel:
CEDA INSTALL GROUP(ADNPCRGP)

CICS, nicht primäre Verbindungsregionen

Der CRD-Server kann auch mit zusätzlichen, nicht primären Verbindungsregionen verwendet werden. Dabei handelt es sich in der Regel um AOR-Regionen (Application Owning Regions).

Anmerkung: Sie müssen diese Schritte nicht ausführen, wenn Ihre CICS-Ressourcendefinitionen mit den BAS (Business Application Services) von CICSplex SM verwaltet werden.

- Stellen Sie die ADM-Lademodule FEK.SFEKLOAD(ADNCRD*) in die CICS-RPL-Kette (DD-Anweisung DFHRPL) dieser nicht primären Verbindungsregionen. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.
- Passen Sie den Job ADNCS DAR an und übergeben Sie ihn, um CSD für diese nicht primären Verbindungsregionen zu aktualisieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.

ADNCS DAR ist in FEK.#CUST.JCL enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

- Installieren Sie die ADM-Gruppe für diese Regionen mit dem entsprechenden CEDA-Befehl. Beispiel:

```
CEDA INSTALL GROUP(ADNARRGP)
```

Manifestrepository (optional)

Developer for System z gibt Clients die Möglichkeit, Manifeste mit Beschreibungen ausgewählter CICS-Ressourcen anzuzeigen und ggf. zu ändern. Änderungen können je nach den vom CICS-Administrator festgelegten Berechtigungen direkt vorgenommen oder in das Manifestrepository zur weiteren Verarbeitung durch einen CICS-Administrator exportiert werden.

Anmerkung:

- Diesen Schritt müssen nur Kunden ausführen, die Manifeste von Developer for System z exportieren, um sie mit dem Manifestverarbeitungstool zu verarbeiten.
- Das Manifestverarbeitungstool ist ein Plug-in zum IBM CICS-Explorer.

Passen Sie den Job ADNVMFST an und übergeben Sie ihn, um die VSAM-Datei für das Manifestrepository anzulegen und zu initialisieren und um diese Datei für die primäre CICS-Verbindungsregion zu definieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation. Für jede primäre CICS-Verbindungsregion muss ein gesondertes Manifestrepository erstellt werden. Alle Benutzer benötigen das Zugriffsrecht UPDATE für das Manifestrepository.

ADNVMFST ist in FEK.#CUST.JCL enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Kapitel 5. SCLM Developer Toolkit (optional)

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

SCLM Developer Toolkit enthält ein auf Eclipse basierendes Plug-in mit Anbindung an SCLM, das den Zugriff auf alle SCLM-Prozesse für die herkömmliche Codeentwicklung ermöglicht. Das Plug-in unterstützt auch die vollständige Java- und J2EE-Entwicklung auf der Workstation. Dazu gehören die Synchronisation mit SCLM auf dem Mainframe-Computer sowie die Builderstellung, die Assemblierung und das Deployment des J2EE-Codes vom Mainframe-Computer.

Voraussetzungen und Prüfliste

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

- APF und LINKLIST aktualisieren
- SCLM-Sprachumsetzer für JAVA/J2EE-Unterstützung definieren
- SCLM-Typen für JAVA/J2EE-Unterstützung definieren
- Sicherheitsregel für die Aktualisierung einer SCLM-VSAM durch Benutzer (optional)
- Ant installieren (optional)

Vor der 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 82.
2. Passen Sie die Konfigurationsdateien für Developer for System z an. Weitere Details enthalten die folgenden Abschnitte:
 - „Aktualisierung von ISPF.conf für SCLMDT“ auf Seite 82
 - „Aktualisierung von rsed.envvars für SCLMDT“ auf Seite 83
3. Definieren Sie die Unterstützung für die Umsetzung langer/kurzer Namen (optional). Weitere Details enthält der Abschnitt „Umsetzung langer/kurzer Namen (optional)“ auf Seite 84.
4. Installieren Sie Ant und passen Sie es an, um die JAVA/J2EE-Buildunterstützung zu verwenden. Weitere Details enthält der Abschnitt „Ant installieren und anpassen (optional)“ auf Seite 86.
5. Aktualisieren Sie SCLM, um spezifische Komponenten für SCLMDT zu definieren. Weitere Details enthält der Abschnitt „Aktualisierung von SCLM für SCLMDT“ auf Seite 87.
6. Richten Sie eine automatisierte, regelmäßige Bereinigung des SCLMDT-Arbeitsbereichs ein (optional). Weitere Details enthält der Abschnitt „Alte Dateien aus WORKAREA entfernen“ auf Seite 88.

Voraussetzungen

Eine Liste der erforderlichen SCLM-Wartung finden Sie in Anhang E, „Voraussetzungen“, auf Seite 337.

In diesem Anhang sind auch die für JAVA/J2EE-Builds in SCLM Developer Toolkit notwendigen Ant-Spezifikationen dokumentiert.

Achtung:

SCLM Developer Toolkit erfordert die Verwendung des TSO/ISPF-Client-Gateways von ISPF. Damit ist implizit z/OS ab Version 1.8 erforderlich.

SCLM Developer Toolkit erfordert zusätzliche Anpassungsschritte für Systemeinstellungen. Lesen Sie hierzu die Beschreibung im Abschnitt „PARMLIB-Änderungen“ auf Seite 16. Einige der Änderungen sind:

- 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 SCLM Developer Toolkit den Fertigstellungsstatus von Jobs und Jobausgaben ab. Beide Methoden erfordern zusätzliche Aufmerksamkeit:

- SDSF muss separat bestellt, installiert und konfiguriert werden und erfordert die Verwendung von JES2.
- Mit den Standardeinstellungen für den TSO-Befehl **OUTPUT** kann ein Benutzer nur Jobausgaben abrufen, die mit seiner Benutzer-ID beginnen. Wenn Sie die Funktion **OUTPUT** vollständig nutzen möchten, 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 seiner Benutzer-ID beginnen. Weitere Informationen zu diesem Exit enthält die Veröffentlichung *TSO/E Customization* (IBM Form SA22-7783).

Benutzer müssen die Zugriffsrechte READ, WRITE und EXECUTE für die z/OS UNIX-Verzeichnisse /tmp/ und /var/rdz/WORKAREA/ haben. Das Verzeichnis WORKAREA/ ist in /var/rdz/ enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

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/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** 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 (,). Weitere Details zur Anpassung von `ISPF.conf` enthält der Abschnitt „Konfigurationsdatei TSO/ISPFISPF.conf des TSO/ISPF-Client-Gateways von ISPF“ auf Seite 48.

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

* OPTIONAL:
*allocjob = FEK.#CUST.CNTL(CRAISPRX)
*ISPF_timeout = 900
```

Abbildung 21. 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 Details enthält Kapitel 16, „TSO-Umgebung anpassen“, auf Seite 269.
- Wenn Sie Batch-Builds ausführen, stellen Sie sicher, dass die angepasste Version des Skeleton FLMLIBS vor der ISPF/SCLM-Skeleton-Bibliothek verknüpft ist.

```
ispslib=hlq.USERSKEL,ISP.SISPSLIB
```

Aktualisierung von `rsed.envvars` für SCLMDT

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

Die Datei `rsed.envvars` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

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

Das folgende Codebeispiel zeigt die SCLMDT-Anweisungen in `rsed.envvars`, die Sie an Ihre Systemumgebung anpassen müssen. Weitere Details zur Anpassung von `rsed.envvars` enthält der Abschnitt „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 33.

```

_SCLMDT_CONF_HOME=/var/rdz/sclmdt
#STEPLIB=$STEPLIB:FEK.SFEKAUTH:FEK.SFEKLOAD
#_SCLMDT_TRANTABLE=FEK.#CUST.LSTRANS.FILE
#ANT_HOME=/usr/lpp/apache/Ant/apache-ant-1.7.1
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CMDSEV_WORK_HOME
CGI_DTWORk=$_SCLMDT_WORK_HOME

```

Abbildung 22. Aktualisierung von *rsed.envvars* für SCLMDT

Umsetzung langer/kurzer Namen (optional)

Vom SCLM Developer Toolkit aus können Sie Dateien mit langen Namen (mit mehr als acht Zeichen oder gemischter Groß-/Kleinschreibung) in SCLM speichern. Zu diesem Zweck wird eine VSAM-Datei verwendet, die die Zuordnung des langen Dateinamens zu dem in SCLM verwendeten und aus acht Zeichen bestehenden 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, die VSAM für die Umsetzung langer/kurzer Namen, erstellen

Passen Sie das Beispielmember FLM02LST in der ISPF-Beispielbibliothek ISP.SISPSAMP an und übergeben Sie es, um die VSAM für die Umsetzung langer/kurzer Namen zu erstellen. Bei den Konfigurationsschritten in dieser Veröffentlichung wird davon ausgegangen, dass Sie die VSAM wie in der folgenden Beispielkonfigurations-JCL FEK.#CUST.LSTRANS.FILE nennen.

```

//FLM02LST    JOB <Jobparameter>
//*
/* ACHTUNG: Dies ist keine JCL-Prozedur und kein vollständiger Job.
/* Vor Verwendung dieses Beispiels müssen Sie die folgenden
/* Änderungen vornehmen:
/* 1. Passen Sie die Jobparameter an Ihre Systemanforderungen an.
/* 2. Ersetzen Sie ***** durch die Platteneinheit für die VSAM.
/* 3. Ersetzen Sie alle Verweise auf FEK.#CUST.LSTRANS.FILE durch
/* Ihre Namenskonvention für die SCLM-Umsetzungs-VSAM.
/*
//CREATE      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE FEK.#CUST.LSTRANS.FILE
SET MAXCC=0
DEFINE CLUSTER(NAME(FEK.#CUST.LSTRANS.FILE) -
               VOLUMES(*****)) -
               RECORDSIZE(58 2048) -
               SHAREOPTIONS(3 3) -
               CYLINDERS(1 1) -
               KEYS(8 0) -
               INDEXED) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.INDEX))

/* ALTERNATIVEN INDEX MIT NICHT EINDEUTIGEN SCHLÜSSELN DEFINIEREN -> ESDS */

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

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

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

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

```

Abbildung 23. FLM02LST - Konfigurations-JCL für Umsetzung langer/kurzer Namen

Anmerkung: Benutzer benötigen für diese VSAM-Datei die Zugriffsberechtigung UPDATE. Lesen Sie hierzu die Beschreibung in Kapitel 10, „Sicherheitsaspekte“, auf Seite 159.

Aktualisierung von rsed.envvars für die Umsetzung langer/kurzer Namen

Entfernen Sie vor Verwendung der Umsetzung langer/kurzer Namen das Kommentarzeichen und setzen Sie die Umgebungsvariable `_SCLMDT_TRANTABLE` in `rsed.envvars`, damit der Name der VSAM für die Umsetzung langer Namen in Kurznamen übereinstimmt.

Die Datei `rsed.envvars` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

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

Ant installieren und anpassen (optional)

Dieser Schritt ist nur erforderlich, wenn Sie in SCLM die JAVA/J2EE-Buildunterstützung verwenden möchten.

Apache Ant ist ein Open-Source-Build-Tool von Java, das Sie von der Webseite <http://ant.apache.org/> herunterladen können. Ant umfasst Textdateien und Scripts, die im ASCII-Format verteilt werden. Für die Ausführung unter z/OS UNIX ist daher eine ASCII-EBCDIC-Umsetzung erforderlich.

Führen Sie die folgenden Schritte aus, um Ant unter z/OS zu implementieren und für Developer for System z zu definieren:

- Laden Sie die neueste Ant-Datei in Binärform in das z/OS UNIX-Dateisystem herunter. Sie sollten die ZIP-Version von ANT herunterladen, da es unter z/OS beim Entpacken von Formaten mit dem Suffix `tar.gz` oder `tar.bz2` zu Problemen kommen kann.
- 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 legen Sie es als aktuelles Verzeichnis mit dem Befehl **cd /home-dir** fest.
- Extrahieren Sie die Datei mit dem JAR-Extraktionsbefehl **jar -xf apache-ant-1.7.1.zip** im aktuellen Verzeichnis. Für die Verwendung des Befehls **jar** muss Ihr lokaler z/OS UNIX-PATH ein Java-Verzeichnis `bin` enthalten. Ist dies nicht der Fall, verwenden Sie den Befehl mit dem vollständig qualifizierten Pfad zur Java-Position `bin` (z. B. **/usr/lpp/java/J5.0/bin/jar -xf apache-ant-1.7.1.zip**).
- Konvertieren Sie alle Ant-Textdateien in EBCDIC. Führen Sie dazu das (ggf. angepasste) Beispielscript `/usr/lpp/rdz/samples/BWBTRANT` aus.

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.
- Aktivieren Sie mit dem Befehl **chmod -R 755 *** für alle Benutzer die Zugriffsrechte READ und EXECUTE für Dateien im ANT-Verzeichnis.
- Setzen Sie vor der Verwendung von Ant in `rsed.envvars` die Umgebungsvariablen `JAVA_HOME` und `ANT_HOME`.

- JAVA_HOME muss auf das Java-Ausgangsverzeichnis zeigen. Beispiel:
JAVA_HOME=/usr/lpp/java/IBM/J5.0
- ANT_HOME 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/rdz/samples/BWBTRANT
- cat ./apache-ant-1.7.1/README
- chmod -R 755 *
- oedit /etc/rsed.envvars

Testen Sie wie folgt, ob die Ant-Initialisierung erfolgreich war:

- Fügen Sie das Ant- und Java-Verzeichnis bin zur Umgebungsvariablen PATH hinzu.

Beispiel:

```
export PATH=/usr/lpp/Apache/Ant/apache-ant-1.7.1/bin:$PATH
export PATH=/usr/lpp/java/IBM/J5.0/bin:$PATH
```

- Führen Sie den **ant**-Befehl **-version** aus, um nach einer erfolgreichen Installation die Version anzuzeigen.

Beispiel:

```
ant -version
```

Anmerkung: Das Setzen der Anweisung PATH auf diese Weise ist nur für Tests erforderlich, nicht aber für den regulären Betrieb.

Aktualisierung von SCLM für SCLMDT

SCLM selbst erfordert für eine Zusammenarbeit mit SCLM Developer Toolkit einige Anpassungsschritte. Weitere Informationen zu den erforderlichen Anpassungstasks enthält das *Administratorhandbuch für SCLM Developer Toolkit* (IBM Form SC12-4344-01):

- Sprachumsetzer für JAVA/J2EE-Unterstützung definieren
- SCLM-Typen für JAVA/J2EE-Unterstützung definieren

Der SCLM-Administrator muss verschiedene anpassbare Werte von Developer for System z kennen, die in Tabelle 13 beschrieben sind, um die Anpassungs- und Projektdefinitionstasks ausführen zu können.

Tabelle 13. Prüfliste für den SCLM-Administrator

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Beispielbibliothek von Developer for System z	<ul style="list-style-type: none"> • FEK.SFEKSAMV • SMP/E-Installation 	
Beispielverzeichnis von Developer for System z	<ul style="list-style-type: none"> • /usr/lpp/rdz/samples • SMP/E-Installation 	

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

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Java-Verzeichnis bin	<ul style="list-style-type: none"> • /usr/lpp/java/J5.0/bin • rsed.envvars - \$JAVA_HOME/bin 	
Ant-Verzeichnis bin	<ul style="list-style-type: none"> • /usr/lpp/Apache/Ant/apache-ant-1.7.1/bin • rsed.envvars - \$ANT_HOME/bin 	
WORKAREA-Ausgangsverzeichnis	<ul style="list-style-type: none"> • /var/rdz • rsed.envvars - \$_CMDSERV_CONF_HOME 	
Ausgangsverzeichnis für SCLMDT-Projektkonfigurationen	<ul style="list-style-type: none"> • /var/rdz/sclmdt • rsed.envvars - \$_SCLMDT_CONF_HOME 	
VSAM für Umsetzung langer/kurzer Namen	<ul style="list-style-type: none"> • FEK.#CUST.LSTRANS.FILE • rsed.envvars - \$_SCLMDT_TRANTABLE 	

Alte Dateien aus WORKAREA entfernen

SCLM Developer Toolkit nutzt einen WORKAREA gemeinsam mit dem TSO/ISPF-Client-Gateway von ISPF. Daher könnte eine regelmäßige Bereinigung angezeigt sein. Weitere Informationen hierzu enthält der Abschnitt „WORKAREA-Bereinigung (optional)“ auf Seite 108.

Kapitel 6. Weitere Anpassungstasks (optional)

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

- „Gespeicherte DB2-Prozedur (optional)“
- „EST-Unterstützung (Enterprise Service Tools) (optional)“ auf Seite 91
- „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 92
- „IRZ-Diagnosefehlernachrichten (optional)“ auf Seite 93
- „RSE-SSL-Verschlüsselung (optional)“ auf Seite 93
- „RSE-Trace (optional)“ auf Seite 96
- „Hostbasierte Eigenschaftsgruppe (optional)“ auf Seite 98
- „Hostbasierte Projekte (optional)“ auf Seite 99
- „File Manager-Integration (optional)“ auf Seite 100
- „Nicht editierbare Zeichen (optional)“ auf Seite 101
- „REXEC (oder SSH) verwenden (optional)“ auf Seite 102
- „APPC-Transaktion für TSO Commands Service (optional)“ auf Seite 104
- „WORKAREA-Bereinigung (optional)“ auf Seite 108

Gespeicherte DB2-Prozedur (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines WLM-Administrators und eines DB2-Administrators:

- WLM aktualisieren
- Neues PROCLIB-Member erstellen
- DB2 aktualisieren

Developer for System z stellt ein Beispiel für eine gespeicherte DB2-Prozedur (Stored Procedure Builder für PL/I und COBOL) bereit, sodass Sie innerhalb des Clients von Developer for System z gespeicherte COBOL- und PL/I-Prozeduren erstellen können.

Anmerkung: Die Beispielmember ELAXM* befinden sich in FEK.#CUST.JCL und FEK.#CUST.PROCLIB, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

WLM-Änderungen (Workload Manager)

Ordnen Sie der JCL-Prozedur des WLM-Adressraums für den Stored Procedure Builder für PL/I und COBOL in den WLM-Anzeigen eine Anwendungsumgebung zu. Informationen hierzu finden Sie in der Veröffentlichung *MVS Planning Workload Management* (IBM Form SA22-7602).

Anmerkung: Sie können in WLM eine neue Umgebung für den Stored Procedure Builder für PL/I und COBOL erstellen oder die erforderlichen Definitionen zu einer vorhandenen Umgebung hinzufügen.

PROCLIB-Änderungen

Passen Sie die Task FEK.#CUST.PROCLIB(ELAXMSAM) der gespeicherten Beispielprozedur wie innerhalb des Members beschrieben an und kopieren Sie sie in SYS1.PROCLIB. Sie müssen wie im nachstehenden Codebeispiel Folgendes angeben:

- den Namen der im WLM definierten Anwendungsumgebung für diese gespeicherte Prozedur
- den Namen des DB2-Subsystems
- das übergeordnete Qualifikationsmerkmal verschiedener Dateien

```
//ELAXMSAM PROC RGN=0M,
//          NUMTCB=1,
//          APPLENV=#w1mwd4z,
//          DB2SSN=#ssn,
//          DB2PRFX='DSN810',
//          COBPRFX='IGY.V3R4M0',
//          PLIPRFX='IBMZ.V3R6M0',
//          LIBPRFX='CEE',
//          LODPRFX='FEK'
//*
//DSNX9WLM EXEC PGM=DSNX9WLM,REGION=&RGN,TIME=NOLIMIT,DYNAMNBR=10,
//          PARM='&DB2SSN,&NUMTCB,&APPLENV'
//STEPLIB DD DISP=SHR,DSN=&DB2PRFX..SDSNEXIT
//          DD DISP=SHR,DSN=&DB2PRFX..SDSNLOAD
//          DD DISP=SHR,DSN=&LIBPRFX..SCEERUN
//          DD DISP=SHR,DSN=&COBPRFX..SIGYCOMP
//          DD DISP=SHR,DSN=&PLIPRFX..SIBMZCMP
//SYSEXEC DD DISP=SHR,DSN=&LODPRFX..SFEKPROC
//SYSTSPRT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSABEND DD DUMMY
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSUT5 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSUT6 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSUT7 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//*
```

Abbildung 24. ELAXMSAM - Task für gespeicherte DB2-Prozedur

Anmerkung:

- Die gespeicherte DB2-Prozedur verwendet die REXX-Exec ELAXMREX aus FEK.SFEKPROC. Ändern Sie diese Position nicht, wenn Sie möchten, dass die SMP/E-Wartung automatisch aktiviert wird.
- Falls Sie das Member ELAXMSAM oder ELAXMREX umbenennen möchten, lesen Sie Kapitel 17, „Mehrere Instanzen ausführen“, auf Seite 277.

DB2-Änderungen

Passen Sie das Beispielmembers ELAXMJCL in der Datei FEK.#CUST.JCL an und übergeben Sie es, um die gespeicherte Prozedur für DB2 zu definieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.

```

//ELAXMJCL JOB <Jobparameter>
//JOBPROC JCLLIB ORDER=(#h1q.SDSNPROC)
//JOBLIB DD DISP=SHR,DSN=#h1q.SDSNEXIT
//        DD DISP=SHR,DSN=#h1q.SDSNLOAD
//*
//RUNTIAD EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
DSN S(#ssn) R(1) T(1)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIAD) -
LIB('#h1q.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
CREATE PROCEDURE SYSPROC.ELAXMREX
( IN FUNCTION_REQUEST VARCHAR(20) CCSID EBCDIC
, IN SQL_ROUTINE_NAME VARCHAR(27) CCSID EBCDIC
, IN SQL_ROUTINE_SOURCE VARCHAR(32672) CCSID EBCDIC
, IN BIND_OPTIONS VARCHAR(1024) CCSID EBCDIC
, IN COMPILE_OPTIONS VARCHAR(255) CCSID EBCDIC
, IN PRECOMPILE_OPTIONS VARCHAR(255) CCSID EBCDIC
, IN PRELINK_OPTIONS VARCHAR(32672) CCSID EBCDIC
, IN LINK_OPTIONS VARCHAR(255) CCSID EBCDIC
, IN ALTER_STATEMENT VARCHAR(32672) CCSID EBCDIC
, IN SOURCE_DATASETNAME VARCHAR(80) CCSID EBCDIC
, IN BUILDOWNER VARCHAR(8) CCSID EBCDIC
, IN BUILDUTILITY VARCHAR(18) CCSID EBCDIC
, OUT RETURN_VALUE VARCHAR(255) CCSID EBCDIC )
PARAMETER STYLE GENERAL RESULT SETS 1
LANGUAGE REXX EXTERNAL NAME ELAXMREX
COLLID DSNREXCS WLM ENVIRONMENT ELAXMSAM
PROGRAM TYPE MAIN MODIFIES SQL DATA
STAY RESIDENT NO COMMIT ON RETURN NO
ASUTIME NO LIMIT SECURITY USER;

COMMENT ON PROCEDURE SYSPROC.ELAXMREX IS
'PLI & COBOL PROCEDURE PROCESSOR (ELAXMREX), INTERFACE LEVEL 0.01';

GRANT EXECUTE ON PROCEDURE SYSPROC.ELAXMREX TO PUBLIC;
//*
```

Abbildung 25. ELAXMJCL – Definition einer gespeicherten DB2-Prozedur

Anmerkung: Stellen Sie sicher, dass die Klausel WLM ENVIRONMENT der Anweisung CREATE PROCEDURE den Namen der WLM-Umgebungsprozedur angibt, die für den Stored Procedure Builder für PL/I und COBOL definiert wurde (standardmäßig ELAXMSAM).

EST-Unterstützung (Enterprise Service Tools) (optional)

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

Im Client von Developer for System z gibt es eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools' (EST). Abhängig vom generierten Codetyp liegen diesem Code Funktionen zugrunde, die durch die Hostinstallation von Developer for System z bereitgestellt werden. In den folgenden Abschnitten wird beschrieben, wie Sie diese Hostfunktionen verfügbar machen:

- Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73
- „Unterstützung bidirektionaler Sprachen für CICS (optional)“ auf Seite 92
- „IRZ-Diagnosefehlernachrichten (optional)“ auf Seite 93

Anmerkung: Die Enterprise Service Tools (EST) umfassen mehrere Tools, zu denen SFM (Service Flow Modeler) und XSE (XML Services for the Enterprise) gehören.

Unterstützung bidirektionaler Sprachen für CICS (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines CICS-Administrators:

- JCL für die CICS-Region aktualisieren
 - Programm für CICS definieren
-

Die Komponente Enterprise Service Tools (EST) von Developer for System z unterstützt verschiedene Formate für arabische und hebräische Schnittstellennachrichten und die bidirektionale Datendarstellung und -bearbeitung in allen Editoren und Ansichten. In Terminalanwendungen werden Anzeigen von links nach rechts und von rechts nach links sowie numerische Felder und Felder mit entgegengesetzter Anzeigeausrichtung unterstützt.

Zu den zusätzlichen bidirektionalen Features und Funktionen gehören unter anderem:

- Der EST-Service-Requester gibt dynamisch bidirektionale Attribute von Schnittstellennachrichten an.
- Die bidirektionale Datenverarbeitung in Service-Flows basiert auf bidirektionalen Attributen (Texttyp, Textausrichtung, numerische Ersetzung und symmetrische Ersetzung). Diese Attribute können in verschiedenen Stadien der Erstellung von Schnittstellen- und Terminal-Flows angegeben werden.
- Der von EST generierte Laufzeitcode umfasst die Umsetzung von Daten in Feldern von Nachrichten mit verschiedenen bidirektionalen Attributen.

Von EST generierter Code kann die BIDI-Konvertierung auch in anderen Umgebungen als CICS SFR (Service Flow Runtime) unterstützen. Ein Beispiel sind Batchanwendungen. Sie können die EST-Generatoren veranlassen, alle Aufrufe bidirektionaler Umsetzungsroutinen aufzunehmen, indem Sie in den EST-Generierungsassistenten die entsprechenden BIDI-Konvertierungsattribute angeben und die generierten Programme mit der entsprechenden Bibliothek für bidirektionale Umsetzung (FEK.SFEKLOAD) verknüpfen.

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

1. Stellen Sie die FEK.SFEKLOAD-Lademodule 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.

Anmerkung: 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 der entsprechenden CEDA-Befehle als Programme für CICS. Beispiel:

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

IRZ-Diagnosefehlernachrichten (optional)

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

- LINKLIST aktualisieren
 - JCL für die CICS-Region aktualisieren
-

Im Client von Developer for System z gibt es eine Codegenerierungskomponente mit der Bezeichnung 'Enterprise Service Tools' (EST). Alle IRZ*- und IIRZ*-Module in der Ladebibliothek FEK.SFEKLOAD müssen dem generierten Code verfügbar gemacht werden, damit der von EST generierte Code Diagnosefehlernachrichten ausgeben kann. EST kann Code für die folgenden Umgebungen generieren:

- CICS
- IMS
- MVS-Batch

Wenn der generierte Code in einer CICS-Transaktion ausgeführt wird, fügen Sie alle Module IRZ* und IIRZ* in FEK.SFEKLOAD zur DFHRPL-DD der CICS-Region hinzu. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.

Machen Sie in allen anderen Situationen alle Module IRZ* und IIRZ* in der Ladebibliothek FEK.SFEKLOAD mithilfe von STEPLIB oder LINKLIST verfügbar. Zu diesem Zweck sollten Sie die Installationsdatei zur Kette hinzufügen, damit angewendete Wartungen automatisch für CICS verfügbar sind.

Wenn Sie sich für die Verwendung von STEPLIB entscheiden, müssen Sie die nicht über LINKLIST verfügbaren Module in der Anweisung STEPLIB der Task definieren, die den Code ausführt.

Wenn die Lademodule nicht verfügbar sind und durch den generierten Code ein Fehler festgestellt wird, wird die folgende Nachricht ausgegeben:

IRZ9999S Abruf des Texts der Laufzeitnachricht 'Language Environment' ist gescheitert. Überprüfen Sie, ob das Laufzeitnachrichtenmodul 'Language Environment' für Facility-IRZ in DFHRPL oder STEPLIB installiert ist.

RSE-SSL-Verschlüsselung (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines Sicherheitsadministrators:

- LINKLIST aktualisieren
 - Sicherheitsregel für das Hinzufügen programmgesteuerter Dateien
 - Sicherheitsregel für das Hinzufügen von Zertifikaten für SSL (optional)
-

Die externe Kommunikation (Client-Host) kann mit SSL (Secure Socket Layer) verschlüsselt werden. Dieses Feature ist standardmäßig inaktiviert und wird von den Einstellungen in `ssl.properties` gesteuert.

Die Datei `ssl.properties` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Der Client kommuniziert während des Verbindungsaufbaus mit dem RSE-Dämon und während der aktuellen Sitzung mit dem RSE-Server. Wenn SSL aktiviert ist, sind beide Datenströme verschlüsselt.

Aufgrund unterschiedlicher Architektur unterstützen der RSE-Dämon und der RSE-Server verschiedene Mechanismen zum Speichern von Zertifikaten. Dies impliziert, dass für den RSE-Dämon sowie den RSE-Server SSL-Definitionen erforderlich sind. Ein gemeinsam genutztes Zertifikat kann verwendet werden, wenn der RSE-Dämon und der RSE-Server dieselbe Zertifikatsverwaltungsmethode verwenden.

Tabelle 14. Mechanismen für den SSL-Zertifikatsspeicher

Zertifikatsspeicher	Erstellt und verwaltet von	RSE-Dämon	RSE-Server
Schlüsseldatei	SAF-kompatibles Sicherheitsprodukt	unterstützt	unterstützt
Schlüsseldatenbank	gskkyman von z/OS UNIX	unterstützt	/
Keystore	Java-Keytool	/	unterstützt

Anmerkung:

- Für die Verwaltung von Zertifikaten sind SAF-kompatible Schlüsseldateien die bevorzugte Methode.
- SAF-kompatible Schlüsseldateien speichern den privaten Schlüssel des Zertifikats in der Sicherheitsdatenbank oder mithilfe von ICSF, der Schnittstelle zur Verschlüsselungshardware in System z. Der Zugriff auf ICSF wird durch Profile in der Sicherheitsklasse CSFSERV geschützt.

Zum Verwalten von SSL verwendet der RSE-Dämon System SSL-Funktionen. Dies impliziert, dass SYS1.SIEALNKE von Ihrer Sicherheitssoftware programmgesteuert und RSE über LINKLIST oder die STEPLIB-Anweisung in rsed.envvars verfügbar sein muss.

Das folgende Codebeispiel zeigt die Datei `ssl.properties`, die Sie an Ihre Systemumgebung anpassen müssen. Wenn eine US-Codepage verwendet wird, beginnen die Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare sind nicht in derselben Zeile zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
# ssl.properties – SSL-Konfigurationsdatei
enable_ssl=false

# Dämonmerkmale

#daemon_keydb_file=
#daemon_keydb_password=
#daemon_key_label=

# Servermerkmale

#server_keystore_file=
#server_keystore_password=
#server_keystore_label=
#server_keystore_type=JCERACFKS
```

Abbildung 26. SSL-Konfigurationsdatei *ssl.properties*

Die Dämon- und Servermerkmale müssen nur gesetzt werden, wenn Sie SSL aktivieren. Weitere Informationen zur SSL-Konfiguration enthält Anhang A, „SSL- und X.509-Authentifizierung konfigurieren“, auf Seite 299.

enable_ssl

Aktivieren oder Inaktivieren der SSL-Kommunikation. Die Standardeinstellung ist `false`. Die einzigen gültigen Optionen sind `true` und `false`.

daemon_keydb_file

Name der Schlüsseldatei von RACF (oder eines ähnlichen Sicherheitsprodukts). Wenn Sie anstelle einer Schlüsseldatei **gskkyman** verwendet haben, um eine Schlüsseldatenbank zu erstellen, geben Sie den Namen der Schlüsseldatenbank an. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn SSL aktiviert ist.

daemon_keydb_password

Lassen Sie die Anweisung auf Kommentar gesetzt bzw. geben Sie keinen Wert an, wenn Sie eine Schlüsseldatei verwenden. Geben Sie andernfalls das Kennwort für die Schlüsseldatenbank an. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn SSL aktiviert ist und Sie eine **gskkyman**-Schlüsseldatenbank verwenden.

daemon_key_label

Die in der Schlüsseldatei oder Schlüsseldatenbank verwendete Bezeichnung des Zertifikats, sofern es sich nicht um das Standardzertifikat handelt. Wenn die Standardbezeichnung verwendet wird, muss diese Angabe auf Kommentar gesetzt werden. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn SSL aktiviert ist und Sie nicht das Standardsicherheitszertifikat verwenden.

server_keystore_file

Name des vom Java-Befehl **keytool** erstellten Keystore oder der Name der RACF-Schlüsseldatei (oder eines ähnlichen Sicherheitsprodukts), wenn `server_keystore_type=JCERACFKS` angegeben ist. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn SSL aktiviert ist.

server_keystore_password

Lassen Sie die Anweisung auf Kommentar gesetzt bzw. geben Sie keinen Wert an, wenn Sie eine Schlüsseldatei verwenden. Geben Sie andernfalls das Kennwort für den Keystore an. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn SSL aktiviert ist und Sie einen **keytool**-Keystore verwenden.

server_keystore_label

Die in der Schlüsseldatei oder dem Keystore verwendete Bezeichnung des Zertifikats. Standardmäßig wird das erste gültige Zertifikat ermittelt. Entfernen Sie das Kommentarzeichen und passen Sie diese Anweisung an, wenn SSL aktiviert ist und Sie nicht das Standardsicherheitszertifikat verwenden.

server_keystore_type

Keystoretyp. Die Standardeinstellung ist JKS. Gültige Werte sind:

Tabelle 15. Gültige Keystoretypen

Schlüsselwort	Keystoretyp
JKS	Java-Keystore
JCERACFKS	SAF-kompatible Schlüsseldatei, wobei der private Schlüssel des Zertifikats in der Sicherheitsdatenbank gespeichert ist
JCECCARACFKS	SAF-kompatible Schlüsseldatei, wobei der private Schlüssel des Zertifikats mithilfe von ICSF gespeichert wird, der Schnittstelle zur Verschlüsselungshardware in System z

Anmerkung: Zum Zeitpunkt der Veröffentlichung ist eine Aktualisierung der Datei /usr/lpp/java/J5.0/lib/security/java.security von IBM z/OS Java erforderlich, damit JCECCARACFKS unterstützt wird. Die folgende Zeile muss hinzugefügt werden:

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
```

Die Ergebnisdatei sieht wie folgt aus:

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA
security.provider.2=com.ibm.jsse2.IBMJSSEProvider2
security.provider.3=com.ibm.crypto.provider.IBMJCE
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider
security.provider.5=com.ibm.security.cert.IBMCertPath
security.provider.6=com.ibm.security.sasl.IBMSASL
```

RSE-Trace (optional)

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

Developer for System z unterstützt zur Problemlösung verschiedene Tracestufen für den internen Programmablauf. RSE und einige von RSE aufgerufene Services ermitteln anhand der Einstellungen in rsecomm.properties den gewünschten Detaillierungsgrad der Ausgabeprotokolle.

Achtung: Änderungen an diesen Einstellungen können zu Leistungseinbußen führen und sollten nur auf Anweisung des IBM Support Center vorgenommen werden.

Die Datei rsecomm.properties befindet sich in /etc/rdz/, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten.

Das folgende Codebeispiel zeigt die Datei `rsecomm.properties`, die Sie an Ihre Trace-Anforderungen anpassen können. Wenn eine US-Codepage verwendet wird, beginnen die Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare sind nicht in derselben Zeile zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
# server.version - NICHT ÄNDERN!
server.version=5.0.0

# Protokollstufe
# 0 - Fehlernachrichten protokollieren
# 1 - Fehlernachrichten und Warnungen protokollieren
# 2 - Fehlernachrichten, Warnungen und Informationen protokollieren
debug_level=1

# Protokollposition
# Log_To_StdOut
# Log_To_File
log_location=Log_To_File
```

Abbildung 27. Konfigurationsdatei für Protokollierung `rsecomm.properties`

server.version

Version des Protokollierungsservers. Die Standardeinstellung ist 5.0.0. Modifizieren Sie diese Einstellung nicht.

debug_level

Detaillierungsgrad für Ausgabeprotokolle. Die Standardeinstellung ist 1 (Protokollierung von Fehlernachrichten und Warnungen). Beachten Sie, dass `debug_level` den Detaillierungsgrad mehrerer Services (und damit zahlreicher Ausgabedateien) steuert. Die Erhöhung des Detaillierungsgrades kann zu Leistungseinbußen führen und sollte nur auf Anweisung des IBM Support Center erfolgen. Weitere Informationen zu den von dieser Anweisung gesteuerten Protokollen enthält der Abschnitt „Traceerstellung für RSE“ auf Seite 146.

Gültige Werte sind:

0	Nur Fehlernachrichten protokollieren
1	Fehlernachrichten und Warnungen protokollieren
2	Fehlernachrichten, Warnungen und Informationsnachrichten protokollieren

Anmerkung: Mit dem Bedienerbefehl **modify rsecommlog** kann `debug_level` dynamisch geändert werden, wie in Kapitel 8, „Bedienerbefehle“, auf Seite 125 beschrieben.

log_location

Ausgabemedium für die RSE-bezogene Protokollierung. Die Standardeinstellung ist `Log_To_File`. Wenn Sie die RSE-Dämonverbindungsmethode (Standardeinstellung) verwenden, nehmen Sie keine Änderung vor, sofern Sie keine Anweisungen durch das zuständige IBM Support Center erhalten haben.

Gültige Werte sind:

Log_To_File	Nachrichten werden an eine eigene Datei im Protokollausgabeverzeichnis gesendet. <ul style="list-style-type: none">• RSE-Dämon: rsedaemon.log in daemonlog• RSE-Thread-Pools: rseserver.log in daemonlog• Benutzer: rsecomm.log in userlog/.eclipse/RSE/\$LOGNAME
Log_To_StdOut	Nachrichten werden an stdout gesendet. <ul style="list-style-type: none">• RSE-Dämon: weitergeleitet an DD STDOUT in der gestarteten Task RSED• RSE-Thread-Pools: nicht definiert• Benutzer: weitergeleitet an stdout.log in userlog/.eclipse/RSE/\$LOGNAME

daemonlog ist der Wert der Anweisung daemon.log in rsed.envvars. Wenn die Anweisung daemon.log auf Kommentar gesetzt oder nicht vorhanden ist, wird der Ausgangspfad der Benutzer-ID verwendet, die der gestarteten Task RSED zugeordnet ist. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert.

Benutzerspezifische Protokolle werden in userlog/.eclipse/RSE/\$LOGNAME gespeichert. Dabei ist userlog der Wert der Anweisung user.log in rsed.envvars und \$LOGNAME ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung user.log in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert.

Hostbasierte Eigenschaftsgruppe (optional)

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

Die Clientkomponente von Developer for System z kann Eigenschaftsgruppen mit Standardwerten für mehrere Eigenschaften definieren (z. B. die bei der Kompilierung von COBOL-Quellcode zu verwendenden COBOL-Compileroptionen). In Developer for System z gibt es integrierte Standardwerte, aber auch die Möglichkeit, angepasste, systemspezifische Standardwerte zu definieren.

Die Position der Konfigurationsdateien mit benutzerdefinierten Eigenschaftsgruppen und Standardwerten ist in der Datei propertiescfg.properties festgelegt, die sich in /etc/rdz/ befindet, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl OEDIT bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Das folgende Codebeispiel zeigt die Datei propertiescfg.properties, die Sie an Ihre Systemumgebung anpassen müssen. Wenn eine US-Codepage verwendet wird, beginnen die Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare sind nicht in derselben Zeile zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
#
# Hostbasierte Eigenschaftsgruppen - Stammkonfigurationsdatei
#
ENABLED=FALSE
RDZ-VERSION=7.5.0.0
PROPERTY-GROUP=/var/rdz/properties
DEFAULT-VALUES=/var/rdz/properties
```

Abbildung 28. propertiescfg.properties - Konfigurationsdatei für hostbasierte Eigenschaftsgruppen

ENABLED

Gibt an, ob Developer for System z die Konfigurationsdateien für Eigenschaftsgruppen und Standardwerte verwendet. Die Standardeinstellung ist FALSE. Die einzigen gültigen Optionen sind TRUE und FALSE.

RDZ-VERSION

Mindestversion der Clientkomponente von Developer for System z für die Verwendung hostbasierter Eigenschaftsgruppen. Die Standardeinstellung ist 7.5.0.0. Modifizieren Sie diese Einstellung nicht.

PROPERTY-GROUP

Position der Konfigurationsdatei für Eigenschaftsgruppen. Die Standardeinstellung ist /var/rdz/properties.

DEFAULT-VALUES

Position der Konfigurationsdatei für Standardwerte. Die Standardeinstellung ist /var/rdz/properties.

Im Information Center für Developer for System z (<http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp>) finden Sie weitere Informationen zur Erstellung der Konfigurationsdatei für Eigenschaftsgruppen (propertygroups.xml) und der Konfigurationsdatei für Standardwerte (defaultvalues.xml).

Hostbasierte Projekte (optional)

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

z/OS-Projekte können in der Perspektive für z/OS-Projekte auf dem Client einzeln definiert werden. Sie können z/OS-Projekte aber auch zentral auf dem Host definieren und dann benutzerabhängig auf dem Client replizieren. Solche hostbasierten Projekte sind vom Aussehen und von der Funktionsweise her mit auf dem Client definierten Projekten identisch. Die Struktur, die Member und die Eigenschaften dieser Projekte können jedoch nicht vom Client geändert werden und sind nur bei bestehender Verbindung mit dem Host verfügbar.

Die Position der Projektdefinitionen ist in der Datei projectcfg.properties festgelegt, die sich in /etc/rdz/ befindet, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Das folgende Codebeispiel zeigt die Datei projectcfg.properties, die Sie an Ihre Systemumgebung anpassen müssen. Wenn eine US-Codepage verwendet wird, beginnen die Kommentarzeilen mit dem Nummernzeichen (#). Datenzeilen dürfen

nur eine Anweisung und ihren zugeordneten Wert haben.
Kommentare sind nicht in derselben Zeile zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
#
# Stammkonfigurationsdatei für hostbasierte Projekte
#
# WSED-VERSION – nicht ändern!
WSED-VERSION=7.0.0.0
# Position der Definitionsdateien für das hostbasierte Projekt angeben
PROJECT-HOME=/var/rdz/projects
```

Abbildung 29. *projectcfg.properties* – Konfigurationsdatei für hostbasierte Projekte

WSED-VERSION

Mindestversion der Clientkomponente von Developer for System z für die Verwendung hostbasierter Projekte. Die Standardeinstellung ist 7.0.0.0. Modifizieren Sie diese Einstellung nicht.

PROJECT-HOME

Basisverzeichnis für die Projektdefinitionen. Die Standardeinstellung ist /var/rdz/projects.

Anmerkung: Für die Aktivierung hostbasierter Projekte muss das Verzeichnis /var/rdz/projects/USERID eine Datei *project.instance* enthalten. Hier gibt /var/rdz/projects die Position der Projektdefinitionsdateien an und USERID ist die Benutzer-ID, mit der sich der Entwickler anmeldet (in Großbuchstaben).

Im Information Center für Developer for System z (<http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp>) finden Sie weitere Informationen zu hostbasierten Projekten.

File Manager-Integration (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines Sicherheitsadministrators:

- Sicherheitsregel für das Hinzufügen programmgesteuerter Dateien

Developer for System z unterstützt den direkten Zugriff von einem Client auf eine begrenzte Gruppe von Funktionen von IBM File Manager für z/OS. IBM File Manager for z/OS stellt umfassende Tools für die Arbeit mit MVS-Dateien und z/OS UNIX-Dateien sowie mit DB2-, IMS- und CICS-Daten bereit. Zu diesen Tools gehören die bekannten Anzeige-, Bearbeitungs-, Kopier- und Druckdienstprogramme von ISPF, die erweitert wurden, um den Anforderungen von Anwendungsentwicklern besser gerecht zu werden. In der aktuellen Version von Developer for System z werden nur das Anzeigen und Bearbeiten von MVS-Dateien (einschließlich aller VSAM-Typen), das Erstellen und Bearbeiten von Schablonen für MVS-Dateien (einschließlich dynamischer Schablonen) sowie erweiterte Kopierprogramme unterstützt.

Beachten Sie, dass Sie das Produkt IBM File Manager for z/OS gesondert bestellen, installieren und konfigurieren müssen. Welche Version von File Manager für Ihre Version von Developer for System z erforderlich ist, können Sie der Veröffentlichung *Rational Developer for System z Prerequisites* (IBM Form SC23-7659) entnehmen. Die Installation und Anpassung dieses Produkts ist nicht in diesem Handbuch beschrieben.

Beachten Sie, dass Developer for System z und File Manager die Schnittstelle für Batchverarbeitung für den Zugriff auf File Manager-Services nicht mehr unterstützen. Sie müssen den File Manager-Listener verwenden.

Anmerkung: Zusätzlich zu den normalen Listener-Konfigurationstasks, die in Ihrer File Manager-Dokumentation beschrieben sind, erfordert Developer for System z, dass die Dateien STEPLIB des Servers programmiert sind.

Die von Developer for System z benötigten File Manager-Integrationsdefinitionen sind in der Datei `FMIEXT.properties` gespeichert, die sich in `/etc/rdz/` befindet, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss neu gestartet werden, damit diese Änderungen wirksam werden.

Das folgende Codebeispiel zeigt die Datei `FMIEXT.properties`, die Sie an Ihre Systemumgebung anpassen müssen. Wenn eine US-Codepage verwendet wird, beginnen die Kommentarzeilen mit dem Nummernzeichen (`#`). Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare sind nicht in derselben Zeile zulässig. Zeilenfortsetzungen werden nicht unterstützt.

```
# Erweiterungsmerkmale für File Manager-Integration (FMI)
#
enabled=false
fmlistenport=1960
```

Abbildung 30. File Manager-Konfigurationsdatei `FMIEXT.properties`

enabled

Gibt an, ob der File Manager-Listener auf demselben Hostsystem verfügbar ist. Der Standardwert ist `false`. Die einzigen zulässigen Werte sind `true` und `false`.

fmlistenport

Vom File Manager-Listener verwendeter Port. Der Standardwert ist 1960. Die Kommunikation über diesen Port ist auf Ihre Hostmaschine beschränkt.

Nicht editierbare Zeichen (optional)

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

Einige Zeichen werden nicht richtig zwischen den Codepages des Hosts (EBCDIC-basiert) und den Client-Codepages (ASCII-basiert) umgesetzt. Der Editor der Clientkomponente von Developer for System z kann diese nicht editierbaren Zeichen anhand der Definitionen in der Datei `uchars.settings` identifizieren. Wenn beim Öffnen einer Datei ein Zeichen in `uchars.settings` identifiziert wird, erzwingt der Editor den schreibgeschützten Modus, damit die Datei durch Speichern nicht beschädigt werden kann.

Die Datei `uchars.settings` befindet sich in `/etc/rdz/`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15. Sie können die Datei mit dem TSO-Befehl **OEDIT** bearbeiten. RSE muss

neu gestartet werden, damit diese Änderungen wirksam werden. Es ist nicht ratsam, diese Datei zu modifizieren. Nehmen Sie Änderungen nur auf Anweisung des IBM Support Center vor.

```
# uchars.settings - Nicht editierbare Codepunkte
#
*          *          0D 15 25;

# DBCS (Japanisch, Koreanisch und Chinesisch)
IBM-930    *          0D 15 1E 1F 25;
IBM-933    *          0D 15 1E 1F 25;
IBM-935    *          0D 15 1E 1F 25;
IBM-937    *          0D 15 1E 1F 25;
IBM-939    *          0D 15 1E 1F 25;
IBM-1390   *          0D 15 1E 1F 25;
IBM-1399   *          0D 15 1E 1F 25;
IBM-1364   *          0D 15 1E 1F 25;
IBM-1371   *          0D 15 1E 1F 25;
IBM-1388   *          0D 15 1E 1F 25;
# UNICODE
UTF-8      *          0D 0A;
UTF-16BE   *          0D 0A;
UTF-16LE   *          0D 0A;
UTF-16     *          0D 0A;
```

Abbildung 31. *uchars.settings* - Konfigurationsdatei für nicht editierbaren Zeichen

Die Datei besteht aus mehreren Einträgen im folgenden Format:

```
HOST-CODEPAGE  LOKALE_CODEPAGE  HEX-CODEPUNKTE ;
```

HEX-CODEPUNKTE steht hier für eine Liste von zweistelligen hexadezimalen Codepunkten, die jeweils durch ein Leerzeichen voneinander getrennt sind und die nicht editierbaren Zeichen bezeichnen. Die Liste muss mit einem Semikolon (;) enden.

Es gelten die folgenden Syntaxregeln:

- Wenn eine US-Codepage verwendet wird, beginnen die Kommentarzeilen mit dem Nummernzeichen (#).
- Datenzeilen dürfen nur eine Anweisung und ihren zugeordneten Wert haben. Kommentare sind nicht in derselben Zeile zulässig.
- Für HOST-CODEPAGE und/oder LOKALE_CODEPAGE kann ein Stern (*) verwendet werden, der als Platzhalter alle Codepages repräsentiert.
- Konkrete Einträge haben Vorrang vor generischen Einträgen (Platzhaltern).
- Die Angabe HOST-CODEPAGE * hat Vorrang, wenn sowohl HOST-CODEPAGE * als auch * LOKALE_CODEPAGE angegeben ist und beide nicht durch HOST-CODEPAGE LOKALE_CODEPAGE außer Kraft gesetzt werden.
- Ist ein Codepagepaar mehrfach angegeben, wird der letzte Eintrag verwendet.

REXEC (oder SSH) verwenden (optional)

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 Host ausführen können. SSH (Secure Shell) ist ein ähnlicher Service, bei

dem jedoch die gesamte Kommunikation mit SSL (Secure Sockets Layer) verschlüsselt wird. Developer for System z nutzt beide Services für ferne (hostbasierte) Aktionen in z/OS UNIX-Unterprojekten.

Developer for System z kann auch für die Verwendung von REXEC (oder SSH) zum Starten eines RSE-Servers auf dem Host konfiguriert werden. Beachten Sie jedoch, dass jede auf diese Weise gestartete Verbindung in einem gesonderten RSE-Server mündet und jeder dieser Server einen gewissen Anteil an den Systemressourcen benötigt. Diese alternative Verbindungsmethode kann daher nur für eine kleine Anzahl von Verbindungen funktionieren.

Da die alternative Verbindungsmethode mit REXEC (oder SSH) den RSE-Dämon umgeht, kann sie nicht auf alle in dieser Veröffentlichung beschriebenen Services zugreifen, z. B. die Einzelserververarbeitung und -überprüfung. Informieren Sie sich bei der IBM Unterstützungsfunktion, ob ein bestimmter Hostservice von der alternativen Verbindungsmethode mit REXEC unterstützt wird.

Anmerkung: Developer for System z verwendet die z/OS UNIX-Version von REXEC und nicht die TSO-Version.

Ferne (hostbasierte) Aktionen für z/OS UNIX-Unterprojekte

Ferne (hostbasierte) Aktionen für z/OS UNIX-Unterprojekte erfordern, dass auf dem Host REXEC oder SSH aktiv ist. Wenn REXEC/SSH nicht für die Verwendung des Standardports konfiguriert ist, muss die Clientkomponente von Developer for System z den korrekten Port für z/OS UNIX-Unterprojekte definieren. Hierfür können Sie die Vorgabenseite **Fenster >Benutzervorgaben... > z/OS-Lösungen >USS-Unterprojekte >Optionen für ferne Aktionen** auswählen. Welcher Port verwendet wird, erfahren Sie im Abschnitt „REXEC (oder SSH) konfigurieren“ auf Seite 104.

Alternative RSE-Verbindungsmethode

Die Clientkomponente von Developer for System z muss die beiden folgenden Werte kennen, um über REXEC (oder SSH) eine RSE-Verbindung starten zu können:

- Das Verzeichnis, in dem sich das Start-Skript `server.zseries` befindet.

Das Skript `server.zseries` ist in `/etc/rdz/` enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Anmerkung: Um die Verwendung von REXEC (oder SSH) als alternative Anmeldemethode zu vermeiden, wird `server.zseries` nicht mehr automatisch in `/etc/rdz` kopiert. Wenn Sie diese Funktion verwenden möchten, müssen Sie die Datei manuell aus `/usr/lpp/rdz/bin` kopieren, wie in folgendem Beispielbefehl gezeigt:

```
cp /usr/lpp/rdz/bin/server.zseries /etc/rdz
```

- Der verwendete Port.

Welcher Port verwendet wird, erfahren Sie im Abschnitt „REXEC (oder SSH) konfigurieren“ auf Seite 104.

REXEC (oder SSH) konfigurieren

Im *Communications Server IP Configuration Guide* (IBM Form SC31-8775) sind die erforderlichen Konfigurationsschritte für REXEC (oder SSH) beschrieben. Anhang C, „INETD konfigurieren“, auf Seite 321 enthält Konfigurationshinweise für Developer for System z. (Spezifische Konfigurationsschritte für Developer for System z sind nicht erforderlich.)

Ein von REXEC verwendeter allgemeiner Port ist 512. Zur Verifizierung können Sie in `/etc/inetd.conf` und `/etc/services` die verwendete Portnummer überprüfen.

- 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`.

Anmerkung: Die Dateien `/etc/inetd.conf` und `/etc/services` können auch andere Namen haben. Weitere Informationen hierzu enthält Anhang C, „INETD konfigurieren“, auf Seite 321.

APPC-Transaktion für TSO Commands Service (optional)

Für diese Anpassungstask, für die die folgenden Ressourcen oder speziellen Anpassungstasks erforderlich sind, benötigen Sie die Unterstützung eines APPC-Administrators und eines WLM-Administrators:

- LINKLIST aktualisieren
- APPC-Transaktion
- WLM aktualisieren

TSO Commands Service kann als ein APPC-Transaktionsprogramm FEKFRSRV implementiert werden. Diese Transaktion fungiert als ein Hostserver, der die von der Workstation abgesetzten TSO- und ISPF-Befehle ausführt. Auf der Workstation ist APPC nicht erforderlich. Der Client kommuniziert über RSE mit FEKFRSRV. Jeder Client kann gleichzeitig eine aktive Verbindung zu mehreren Hosts haben.

Anmerkung:

- Developer for System z verwendet standardmäßig das TSO/ISPF-Client-Gateway von ISPF, um auf TSO Commands Service zuzugreifen.
- Falls Sie sich nicht mit APPC auskennen, lesen Sie Anhang D, „APPC konfigurieren“, auf Seite 329, bevor Sie mit den Informationen in diesem Abschnitt fortfahren.
- Für die Kommunikation mit FEKFRSRV verwendet RSE die TCP/IP-REXX-Socket-API. Dies setzt voraus, dass die TCP/IP-Ladebibliothek (Standardeinstellung `TCPIP.SEZALOAD`) über LINKLIST oder die Anweisung `STEPLIB` in `rsed.envvars` verfügbar sein muss.
- RSE muss neu gestartet werden, damit die beschriebenen Änderungen wirksam werden.

Vorbereitungen

- Vor dem Konfigurieren des TSO Commands-Servers wird vorausgesetzt, dass Sie die folgenden Tasks abgeschlossen haben. Die genannten Tasks sind in den angegebenen Veröffentlichungen beschrieben.
 1. Installieren, konfigurieren und starten Sie VTAM auf Ihrem z/OS-System. Weitere Informationen hierzu enthält der *Communications Server IP SNA Network Implementation Guide* (IBM Form SC31-8777).
 2. Installieren, konfigurieren und starten Sie TCP/IP auf Ihrem z/OS-System. Weitere Informationen hierzu enthält Anhang B, „TCP/IP konfigurieren“, auf Seite 313.
 3. Konfigurieren und starten Sie APPC und das ASCH-Subsystem (APPC-Transaktionsscheduler). Weitere Informationen hierzu enthält Anhang D, „APPC konfigurieren“, auf Seite 329.
- Mit der folgenden Beispiel-REXX können Sie APPC in ISPF-Anzeigen verwalten.

```
/* REXX – APPC-Verwaltung in ISPF-Anzeigen*/
address ISPEXEC
"LIBDEF ISPMLIB DATASET ID('ICQ.ICQMLIB') STACK"
"LIBDEF ISPPLIB DATASET ID('ICQ.ICQPLIB') STACK"
"LIBDEF ISPSLIB DATASET ID('ICQ.ICQSLIB') STACK"
"LIBDEF ISPTLIB DATASET ID('ICQ.ICQTLIB') STACK"
address TSO "ALTLIB ACT APPLICATION(CLIST)",
            "DSN('ICQ.ICQCCLIB') UNCOND QUIET"
"SELECT CMD(%ICQASRM0) NEWAPPL(ICQ) PASSLIB"
address TSO "ALTLIB DEACT APPLICATION(CLIST) QUIET"
"LIBDEF ISPMLIB"
"LIBDEF ISPPLIB"
"LIBDEF ISPSLIB"
"LIBDEF ISPTLIB"
exit
```

Abbildung 32. REXX für APPC-ISPF-Anzeigen

Anmerkung: Mit diesem Tool können Sie die APPC-Transaktion inaktivieren. Die Transaktion ist unverändert vorhanden, akzeptiert dann jedoch keine Verbindungen mehr.

- Das Definieren der APPC-Transaktion erfordert Know-how zu verschiedenen Bereichen des Betriebssystems MVS. Gehen Sie die folgende Prüfliste mit erfahrenen Administratoren durch, bevor Sie Ihre Arbeit fortsetzen.

Tabelle 16. Prüfliste für APPC-Transaktionen

Fachwissen	Erforderliche Informationen:	
	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
APPC-Administrator	Dateiname von TPDATA <ul style="list-style-type: none"> • Standardwert: SYS1.APPCTP • Der Wert ist in SYS1.PARMLIB(APPCPMxx) enthalten. 	
APPC-Administrator	Zu verwendender Transaktionsname (möglicherweise nicht vorhanden) <ul style="list-style-type: none"> • Standardwert: FEKFRSRV • Vorhandene Transaktionen können durch Auswahl von "TP Profile Administration" im ISPF-Menü 'APPC' abgefragt werden. 	

Tabelle 16. Prüfliste für APPC-Transaktionen (Forts.)

Fachwissen	Erforderliche Informationen:	Wert
	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	
APPC-Administrator	Zu verwendende APPC-Transaktionsklasse <ul style="list-style-type: none"> • Standardwert: A • APPC-Klassen sind in SYS1.PARMLIB(ASCHPMxx) definiert. 	
WLM/SRM	TSO-Leistungsgruppe und -Domäne <ul style="list-style-type: none"> • Kein IBM Standardwert (standortabhängig) 	
RACF	Jeder Benutzer von Developer for System z hat Zugriff auf ein OMVS-Segment. (Dies ist erforderlich.) <ul style="list-style-type: none"> • Kein IBM Standardwert (standortabhängig) • Der TSO-RACF-Befehl LU userid OMVS zeigt ein vorhandenes persönliches OMVS-Segment an. 	
RACF	Jeder Benutzer von Developer for System z muss Lesezugriff (READ) auf HLQ.SFEKPROC(FEKFRSRV) haben. <ul style="list-style-type: none"> • Kein IBM Standardwert (standortabhängig) • Der TSO-RACF-Befehl LD AUTHUSER DATASET('HLQ.SFEKPROC.**') zeigt Benutzer und Gruppen mit der Zugriffsebene für die Dateien dieses Dateiprofils an. 	

Weitere Informationen zur Verwaltung von WLM/SRM finden Sie in der Veröffentlichung *MVS Planning Workload Management* (IBM Form SA22-7602). Weitere Informationen zu OMVS-Segmenten und Profilen für Dateischutz enthält der *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Implementierung

Anmerkung: Die Beispielmember FEKAPPC* befinden sich in FEK.#CUST.JCL, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

1. Definieren Sie die Planungsinformationen (Klasse) für den APPC-Transaktions-scheduler, wenn Sie keine externe Transaktionsklasse verwenden. Nehmen Sie eine Definition für die Klasse in SYS1.PARMLIB(ASCHPMxx) auf, damit sie vom Transaktionsprogramm FEKFRSRV verwendet wird. Diese Klasse wird in der Beispiel-JCL FEK.#CUST.JCL(FEKAPPCC) verwendet. Die Klasse in FEKAPPCC muss deshalb mit der in SYS1.PARMLIB(ASCHPMxx) definierten Klasse übereinstimmen. Beispiel:

```
CLASSADD
  CLASSNAME(A)
  MAX(20)
  MIN(1)
  MSGLIMIT(200)
```

Anmerkung:

- TSO Commands Service erfordert außerdem die Angabe der Standardspezifikationen in den Abschnitten OPTIONS und TPDEFAULT von SYS1.PARMLIB(ASCHPMxx). Weitere Informationen hierzu enthält Anhang D, „APPC konfigurieren“, auf Seite 329.
 - Die verwendete APPC-Transaktionsklasse muss genug APPC-Initiatoren haben, damit für jeden gleichzeitig angemeldeten Benutzer von Developer for System z ein Initiator verfügbar ist.
2. Definieren Sie die APPC-Transaktion, die als Befehlsserver verwendet werden soll. Zum Definieren dieser Transaktion können Sie die Beispiel-JCL FEK.#CUST.JCL(FEKAPPCC) verwenden. Anweisungen zum Anpassen dieser JCL finden Sie direkt in der JCL.

Anmerkung:

- a. Falls Sie den Namen des Transaktionsprogramms in diesem Schritt (standardmäßig FEKFRSRV) geändert haben, müssen Sie den neuen Namen _FEKFSCMD_TP_NAME_ in rsed.envvars zuordnen. Lesen Sie hierzu die Beschreibung im Abschnitt „RSE-Konfigurationsdatei rsed.envvars“ auf Seite 33.
 - b. Die APPC-Transaktion verwendet die REXX-Exec FEKFRSRV aus FEK.SFEKPROC. Ändern Sie diese Position nicht, wenn Sie möchten, dass die SMP/E-Wartung automatisch aktiviert wird.
 - c. Zum Anzeigen der Transaktion steht die Beispiel-JCL FEK.#CUST.JCL(FEKAPPCL) und zum Löschen der Transaktion FEK.#CUST.JCL(FEKAPPCX) zur Verfügung.
3. Aktivieren Sie die Verwendung von APPC durch RSE, indem Sie das Kommentarzeichen vor der Anweisung RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DTSO_SERVER=APPC" in rsed.envvars entfernen (siehe Abschnitt „RSE-Konfigurationsdatei rsed.envvars“ auf Seite 33).
4. Steuern Sie die Zuteilungspriorität des Transaktionsprogramms, indem Sie FEKFRSRV in WLM (Workload Manager) eine Domäne und eine Leistungsgruppe zuordnen. Da FEKFRSRV TSO-Befehle absetzt, sollte das Transaktionsprogramm einer TSO-Leistungsgruppe zugeordnet werden.
5. Definieren Sie ein Standard-OMVS-Segment für das System oder ein individuelles Segment für jeden Benutzer von Developer for System z.
6. Gewähren Sie Benutzern von Developer for System z die Zugriffsberechtigung READ für die ausführbare TSO-Bibliothek FEK.SFEKPROC von Developer for System z.

Hinweise zur Verwendung von APPC

- Wenn Sie APPC für TSO Commands Service verwenden, ist Developer for System z bei der Initialisierung darauf angewiesen, dass TCP/IP mit dem richtigen Hostnamen konfiguriert ist. Dies impliziert, dass die verschiedenen TCP/IP- und Resolverkonfigurationsdateien ordnungsgemäß definiert sein müssen. Informationen zur TCP/IP- und Resolver-Anpassung finden Sie in Anhang B, „TCP/IP konfigurieren“, auf Seite 313 und im Abschnitt *TCPIP.DATA configuration statements* der Veröffentlichung *Communications Server IP Configuration Reference* (IBM Form SC31-8776).

Sie können Ihre TCP/IP-Konfiguration testen, indem Sie den RSE-Dämon mit dem Parameter IVP=IVP starten oder das Installationsprüfprogramm fekfivpt verwenden. Lesen Sie hierzu Kapitel 7, „Installationsprüfung“, auf Seite 109.

- Wenn Sie APPC für TSO Commands Service verwenden, erfordert Developer for System z pro geöffneten MVS-Datei für die hostinterne Kommunikation ein gesondertes Socket (TCP/IP-Port). Jeder verfügbare Port kann verwendet werden. Dieser Portauswahlmechanismus kann nicht geändert werden.
- Wenn Sie APPC für TSO Commands Service verwenden, muss für das Lesen und Schreiben einer MVS-Datei eine Dateisystemdomäne mit physischen Sockets verwendet werden. Wenn das Dateisystem nicht ordnungsgemäß definiert ist oder nicht genug Sockets hat, verhindert der Sperrenmanager (FFS) Lese-/Schreibanforderungen. Lesen Sie hierzu den Abschnitt „MVS-Dateien können nicht geöffnet werden“ auf Seite 157.
- Wenn Sie APPC für TSO Commands Service verwenden, um die Konfiguration des TSO/ISPF-Client-Gateways von ISPF zu vermeiden, denken Sie daran, dass andere Services (z. B. SCLM Developer Toolkit) auf das TSO/ISPF-Client-Gateway angewiesen sind.
- Allgemeine Hinweise zur Verwendung von APPC enthält Anhang D, „APPC konfigurieren“, auf Seite 329.

WORKAREA-Bereinigung (optional)

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

Das TSO/ISPF-Client-Gateway von ISPF und SCLM Developer Toolkit speichern im Verzeichnis WORKAREA temporäre Arbeitsdateien, die vor dem Schließen der Sitzung entfernt werden. Temporäre Ausgaben bleiben jedoch manchmal enthalten. Dies ist beispielsweise der Fall, wenn während der Verarbeitung ein Kommunikationsfehler auftritt. Sie sollten den Inhalt des Verzeichnisses WORKAREA deshalb von Zeit zu Zeit löschen.

z/OS UNIX stellt das Shell-Skript skulker bereit, mit dem Sie Dateien ausgehend von dem Verzeichnis, in dem sie enthalten sind, und ausgehend von ihrem Alter löschen können. In Verbindung mit dem z/OS UNIX-Dämon cron, der Befehle an angegebenen Tagen und zu vorgegebenen Zeiten ausführt, können Sie ein automatisiertes Tool konfigurieren, das das Verzeichnis WORKAREA regelmäßig bereinigt. Weitere Informationen zum Skript skulker und zum Dämon cron enthält die Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802).

Anmerkung: Das Verzeichnis WORKAREA/ ist in /var/rdz/ enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Kapitel 7. Installationsprüfung

Nach der vollständigen Produkthanpassung können Sie die in diesem Kapitel beschriebenen IVPs (Installation Verification Programs) verwenden, um die erfolgreiche Konfiguration der zentralen Produktkomponenten zu überprüfen.

Gestartete Tasks prüfen

JMON, JES Job Monitor

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

```
JM200I Server initialization complete.
```

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

Anmerkung: Starten Sie JES Job Monitor, bevor Sie mit weiteren Installationsprüftests (IVP, Installation Verification Program) fortfahren.

LOCKD, Sperrdämon

Starten Sie die gestartete Task (bzw. den Benutzerjob) LOCKD. Der Sperrdämon gibt nach einem erfolgreichen Start die folgende Konsolnachricht aus:

```
FEK501I Lock daemon started, port=4036, cleanup interval=1440,  
log level=1
```

RSED, RSE-Dämon

Starten Sie die gestartete Task (bzw. den Benutzerjob) RSED mit dem Parameter IVP=IVP. Bei Verwendung dieses Parameters wird der Server nach Ausführung einiger Installationsprüftests beendet. Die Ausgabe dieser Tests ist in DD STDOUT verfügbar. Bei bestimmten Fehlern sind auch in DD STDERR Daten verfügbar. Die STDOUT-Daten sollten wie im folgenden Beispiel aussehen:

Anmerkung: Starten Sie den RSE-Dämon ohne die IVP-Parameter, bevor Sie mit weiteren IVP-Tests fortfahren. Der RSE-Dämon gibt nach einem erfolgreichen Start die folgende Konsolnachricht aus:

```
FEK002I RseDaemon started. (port=4035)  
  
RSE daemon IVP test  
  
Wed Jul 2 17:11:52 2008 UTC  
uid=8(STCRSE) gid=1(STCGROUP)  
  
RSE daemon port is 4035  
RSE configuration files located in /etc/rdz  
  
-----  
current environment variables  
-----  
@="/usr/lpp/rdz/bin/rsed.sh" @[1]="4035" @[2]="/etc/rdz"  
CGI_DTCONF="/var/rdz/scldmt"  
CGI_DTWORK="/var/rdz"  
CGI_TRANTABLE="FEK.#CUST.LSTRANS.FILE"  
CLASSPATH=".:usr/lpp/rdz/lib:usr/lpp/rdz/lib/dstore_core.jar:usr/lpp/  
ERRNO="0"
```

```

HOME="/tmp"
IFS="
"
JAVA_HOME="/usr/lpp/java/J5.0"
JAVA_PROPAGATE="NO"
LANG="C"
LIBPATH=".:usr/lib:usr/lpp/java/J5.0/bin:usr/lpp/java/J5.0/bin/classi
LINENO="66"
LOGNAME="STCRSE"
MAILCHECK="600"
OLDPWD="/tmp"
OPTIND="1"
PATH=".:usr/lpp/java/J5.0/bin:usr/lpp/rdz/bin:usr/lpp/ispf/bin:/bin:/
PPID="33554711"
PS1="\$ "
PS2="> "
PS3="#? "
PS4="+ "
PWD="/etc/rdz"
RANDOM="27298"
RSE_CFG="/etc/rdz"
RSE_HOME="/usr/lpp/rdz"
RSE_LIB="/usr/lpp/rdz/lib"
SECONDS="0"
SHELL="/bin/sh"
STEPLIB="NONE"
TZ="EST5EDT"
_BPX_SHAREAS="YES"
_BPX_SPAWN_SCRIPT="YES"
_CEE_DMPRTARG="/tmp"
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_CMDSERV_BASE_HOME="/usr/lpp/ispf"
_CMDSERV_CONF_HOME="/etc/rdz"
_CMDSERV_WORK_HOME="/var/rdz"
_RSE_CMDSERV_OPTS="&SESSION=SPAWN"
_RSE_DAEMON_CLASS="com.ibm.etools.zos.server.RseDaemon"
_RSE_DAEMON_IVP_TEST="1"
_RSE_DAEMON_PORT="4035"
_RSE_JAVAOPTS=" -DISPF_OPTS='&SESSION=SPAWN' -DA_PLUGIN_PATH=/usr/lpp/rd
_RSE_POOL_SERVER_CLASS="com.ibm.etools.zos.server.ThreadPoolProcess"
_RSE_PWD="/tmp"
_RSE_SERVER_CLASS="org.eclipse.dstore.core.server.Server"
_RSE_SERVER_TIMEOUT="120000"
_SCLMDT_BASE_HOME="/usr/lpp/rdz"
_SCLMDT_CONF_HOME="/var/rdz/sclmdt"
_SCLMDT_TRANTABLE="FEK.#CUST.LSTRANS.FILE"
_SCLMDT_WORK_HOME="/var/rdz"
_SCLM_BASE="/var/rdz/WORKAREA"
_SCLM_BWBCALL="/usr/lpp/rdz/bin/BWBCALL"
_SCLM_DWGET="/var/rdz/WORKAREA"
_SCLM_DWTRANSFER="/var/rdz/WORKAREA"
_SCLM_J2EPUT="/var/rdz/WORKAREA"

-----
java startup test...
-----
java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition (build pmz31dev-2008031
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmz3123-2008
J9VM - 20080314_17962_bHdSMr
JIT - 20080130_0718ifx2_r8
GC - 200802_08)
JCL - 20080314

-----
TCP/IP IVP test...
-----

```

```
Wed Jul 2 13:11:54 EDT 2008
uid=8(STCRSE) gid=1(STCGROUP)
using /etc/rdz/rsed.envvars
```

```
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
```

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```
res_init Resolver values:
```

```
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table      = Default
UserId/JobName         = STCRSE
Caller API             = LE C Sockets
Caller Mode            = EBCDIC
(L) DataSetPrefix     = TCPIP
(L) HostName          = CDFMVS08
(L) TcpIpJobName      = TCPIP
(L) DomainOrigin      = RALEIGH.IBM.COM
(L) NameServer        = 9.42.206.2
                      = 9.42.206.3
(L) NsPortAddr        = 53
(L) ResolveVia        = UDP
(*) Options NDots     = 1
(*) SockNoTestStor    = NO
(*) AlwaysWto         = NO
(*) LookUp            = DNS LOCAL
(L) ResolverTimeout   = 10
(L) ResolverUdpRetries = 1
(L) MessageCase       = MIXED
```

```
res_init Succeeded
```

```
res_init Started: 2008/07/02 13:11:54.755363
```

```
res_init Ended: 2008/07/02 13:11:54.755371
```

```
*****
```

```
MVS TCP/IP NETSTAT CS V1R9 TCPIP Name: TCPIP 13:11:54
```

```
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled
```

```
-----
host IP address:
-----
```

```
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75
```

```
Success, addresses match
```

```
-----
PassTicket IVP test...
-----
```

```
Success, PassTicket IVP finished normally
```

```
-----
RSE daemon IVP ended
```

Services prüfen

Die Installation von Developer for System z stellt mehrere Installationsprüfprogramme (IVP, Installation Verification Programs) für die Basisservices und die optionalen Services bereit. Die IVP-Skripts befinden sich im Installationsverzeichnis (standardmäßig /usr/lpp/rdz/bin/).

Tabelle 17. Installationsprüfprogramme für Services

fekfivpa	„Verbindung zu TSO Commands Service mit APPC (optional)“ auf Seite 118
fekfivpd	„RSE-Dämonverbindung“ auf Seite 115
fekfivpi	„Verbindung mit TSO/ISPF-Client-Gateway von ISPF“ auf Seite 116
fekfivpj	„JES Job Monitor-Verbindung“ auf Seite 115
fekfivpl	„Sperrdämonverbindung“ auf Seite 116
fekfivpr	„REXEC-Verbindung (optional)“ auf Seite 120
fekfivps	„SCLMDT-Verbindung (optional)“ auf Seite 118
fekfivpt	„TCP/IP konfigurieren“ auf Seite 114
fekfivpz	„REXEC/SSH-Shell-Script (optional)“ auf Seite 121

Für die nachfolgenden Tasks wird vorausgesetzt, dass Sie aktivierter z/OS UNIX-Benutzer sind. Zum Aktivieren können Sie den TSO-Befehl **OMVS** absetzen. Mit dem Befehl **exit** können Sie zu TSO zurückkehren.

Für die Benutzer-ID, die die Installationsprüfprogramme (Installation Verification Programs, IVPs) ausführt, ist eine große Regionsgröße erforderlich, weil speicherintensive Funktionen (wie beispielsweise 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 auch andere Fehler auftreten; z. B. schlägt das Starten von Java möglicherweise fehl.)

```
CEE5213S The signal SIGPIPE was received.
%z/OS UNIX command%: command was killed by signal number 13
    %line-number% **%    %REXX command%
    RC(137)
```

Anmerkung: Die gestarteten Tasks von Developer for System z müssen aktiv sein, bevor der IVP-Test gestartet wird.

Installationsprüfprogramm initialisieren

Bei allen Beispielbefehlen in diesem Abschnitt wird vorausgesetzt, dass bestimmte Umgebungsvariablen gesetzt sind. Wenn das der Fall ist, sind die IVP-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 ("\$" ist die z/OS UNIX-Eingabeaufforderung):

```
$ pwd
/u/userid
$ cd /etc/rdz
$ ./ivpinit
RSE configuration files located in /etc/rdz --default
added /usr/lpp/rdz/bin to PATH
```

Der erste Punkt (".") in `./ivpinit` ist ein z/OS UNIX-Befehl zur Ausführung der Shell in der aktuellen Umgebung, damit die in der Shell gesetzten Umgebungsvariablen auch nach dem Beenden der Shell in Kraft bleiben. Der zweite Punkt bezieht sich auf das aktuelle Verzeichnis.

Anmerkung:

- Wenn `./ivpinit` NICHT vor den `fekfivp*`-Scripts ausgeführt wird, muss der Pfad zu diesen Scripts angegeben werden, wenn sie aufgerufen werden. Sehen Sie sich dazu das folgende Beispiel an:

```
/usr/lpp/rdz/bin/fekfivpr 512 USERID
```

Die meisten `fekfivp*`-Scripts fordern außerdem die Position der angepassten Datei `rsed.envvars` an, wenn `./ivpinit` nicht zuerst ausgeführt wird.

- Einige IVP-Tests verwenden die TCP/IP-REXX-Socket-API, die erfordert, dass die TCP/IP-Ladebibliothek (standardmäßig `TCPIP.SEZALOAD`) in der `LINKLIST` oder `STEPLIB` enthalten ist. Für die Ausführung solcher IVP-Tests können die folgenden Befehle erforderlich sein ("\$" ist die z/OS UNIX-Eingabeaufforderung):

```
$ EXPORT STEPLIB=$STEPLIB:TCPIP.SEZALOAD
```

Wenn zu einer vorhandenen `STEPLIB` eine Bibliothek ohne APF-Berechtigung hinzugefügt wird, werden die APF-Berechtigungen der vorhandenen `STEPLIB`-Dateien entfernt.

Beachten Sie auch, dass `TCPIP.SEZALOAD` vor `CEE.SCEELKED` eingefügt werden muss, wenn `CEE.SCEELKED` in `LINKLIST` oder `STEPLIB` verwendet wird. Andernfalls wird für die TCP/IP-REXX-Socketaufrufe ein Systemabbruch 0C1 ausgegeben.

Informationen zur Diagnostizierung von RSE-Verbindungsproblemen können Sie Kapitel 9, „Konfigurationsprobleme lösen“, auf Seite 137 oder den technischen Hinweisen auf der Supportseite für Developer for System z (<http://www-306.ibm.com/software/awdtools/rdz/support/>) entnehmen.

Portverfügbarkeit

Die Portverfügbarkeit für JES Job Monitor, den RSE-Dämon sowie optional für REXEC oder SSH können Sie durch Absetzen des Befehls **netstat** prüfen. Das Ergebnis sollte die von diesen Services verwendeten Ports zeigen. Sehen Sie sich dazu die folgenden Beispiele an (\$ ist die z/OS UNIX-Eingabeaufforderung):

IPV4

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy TCPIP Name: TCPIP 13:57:36
User Id Conn Local Socket Foreign Socket State
-----
RSED 0000004B 0.0.0.0..4035 0.0.0.0..0 Listen
LOCKD 0000004C 0.0.0.0..4036 0.0.0.0..0 Listen
JMON 00000037 0.0.0.0..6715 0.0.0.0..0 Listen
```

IPV6

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy TCPIP Name: TCPIP 14:03:35
User Id Conn State
-----
RSED 0000004B Listen
```

```

Local Socket: 0.0.0.0..4035
Foreign Socket: 0.0.0.0..0
LOCKD 0000004C Listen
Local Socket: 0.0.0.0..4036
Foreign Socket: 0.0.0.0..0
JMON 00000037 Listen
Local Socket: 0.0.0.0..6715
Foreign Socket: 0.0.0.0..0

```

TCP/IP konfigurieren

Wenn Sie APPC für TSO Commands Service verwenden, ist Developer for System z bei der Initialisierung darauf angewiesen, dass TCP/IP mit dem richtigen Hostnamen konfiguriert ist. Dies impliziert, dass die verschiedenen TCP/IP- und Resolverkonfigurationsdateien ordnungsgemäß definiert sein müssen. Weitere Informationen zur TCP/IP- und Resolverkonfiguration enthält Anhang B, „TCP/IP konfigurieren“, auf Seite 313. Führen Sie den folgenden Befehl aus, um die aktuellen Einstellungen zu überprüfen:

```
fekfivpt
```

Anmerkung: Dieses Installationsprüfverfahren setzt den TCP/IP-Befehl **netstat** ab, dessen Ausführung möglicherweise durch Ihre Sicherheitssoftware verhindert wird.

Der Befehl sollte eine Ausgabe wie im folgenden Beispiel zurückgeben (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpt
```

```

Wed Jul 2 13:11:54 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

```

```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
```

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```

res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table = Default
UserId/JobName = USERID
Caller API = LE C Sockets
Caller Mode = EBCDIC
(L) DataSetPrefix = TCPIP
(L) HostName = CDFMVS08
(L) TcpIpJobName = TCPIP
(L) DomainOrigin = RALEIGH.IBM.COM
(L) NameServer = 9.42.206.2
                  9.42.206.3
(L) NsPortAddr = 53 (L) ResolverTimeout = 10
(L) ResolveVia = UDP (L) ResolverUdpRetries = 1
(*) Options NDots = 1
(*) SockNoTestStor
(*) AlwaysWto = NO (L) MessageCase = MIXED
(*) LookUp = DNS LOCAL
res_init Succeeded
res_init Started: 2008/07/02 13:11:54.755363
res_init Ended: 2008/07/02 13:11:54.755371
*****

```



```
MVS TCP/IP NETSTAT CS V1R0          TCPIP Name: TCPIP          13:11:54
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled
```

```
-----
host IP address:
-----
```

```
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75
```

```
Success, addresses match
```

RSE-Dämonverbindung

Führen Sie den folgenden Befehl aus, um die RSE-Dämonverbindung zu überprüfen. Ersetzen Sie 4035 durch den vom RSE-Dämon verwendeten Port und USERID durch eine gültige Benutzer-ID.

```
fekfivpd 4035 USERID
```

Nach einer Aufforderung zur Kennworteingabe sollte der Befehl eine Ausgabe wie im folgenden Beispiel zurückgeben (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpd 4035 USERID
```

```
Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
Password:
SSL is disabled
connected
8108
570655399
Success
```

Anmerkung: Wenn Sie eine SSL-fähige Verbindung testen, überprüfen Sie beim Empfang der folgenden Fehlernachricht, ob Sie den richtigen Port angegeben haben: gsk_secure_socket_init() failed: Socket closed by remote partner

JES Job Monitor-Verbindung

Führen Sie den folgenden Befehl aus, um die JES Job Monitor-Verbindung zu überprüfen. Ersetzen Sie 6715 durch die Portnummer von JES Job Monitor.

```
fekfivpj 6715
```

Der Befehl sollte die Bestätigungsnachricht von JES Job Monitor zurückgeben. Vergleichen Sie hierzu das folgende Beispiel (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpj 6715
```

```
Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
hostName=CDFMVS08
hostAddr=9.42.112.75
```

```
Waiting for JES Job Monitor response...
ACKNOWLEDGE01v03
```

```
Success
```

Sperrdämonverbindung

Führen Sie den folgenden Befehl aus, um die Sperrdämonverbindung zu überprüfen.

```
fekfivpl
```

Der Befehl sollte eine Ausgabe wie im folgenden Beispiel zurückgeben (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpl
```

```
Mon Jun 29 08:00:38 EDT 2009
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
hostName=CDFMVS08
hostAddr=9.42.112.75
```

```
Registering user to Lock Daemon...
Waiting for Lock Daemon response...
```

```
Querying to Lock Daemon...
Waiting for Lock Daemon response...
USERID
```

```
Unregistering user from Lock Daemon...
Waiting for Lock Daemon response...
```

```
Querying to Lock Daemon...
Waiting for Lock Daemon response...
```

```
Success
```

Verbindung mit TSO/ISPF-Client-Gateway von ISPF

Überprüfen Sie die Verbindung mit dem TSO/ISPF-Client-Gateway von ISPF, indem Sie den folgenden Befehl ausführen:

```
fekfivpi
```

Der Befehl sollte die Ergebnisse der auf das TSO/ISPF-Client-Gateway von ISPF bezogenen Überprüfungen zurückgeben (Variablen, HFS-Module, Start und Stopp der TSO/ISPF-Sitzung). Vergleichen Sie hierzu das folgende Beispiel (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpi
```

```
Wed Jul 2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----
/etc/rdz/ISPF.conf content:
```

```
-----  
isplib=ISP.SISPMENU  
isptlib=ISP.SISPTENU  
ispplib=ISP.SISPPENU  
ispslib=ISP.SISPSLIB  
sysproc=ISP.SISPCLIB,FEK.SFEKPROC  
-----
```

```
Host install verification for RSE  
Review IVP log messages from HOST below :  
-----
```

RSE connection and base TSO/ISPF session initialization check only

*** CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

```
Server PATH          =  
/usr/lpp/java/J5.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:  
/bin:/usr/sbin:.
```

```
STEPLIB              = FEK.SFEKAUTH:FEK.SFEKLOAD
```

```
_CMDSERV_BASE_HOME   = /usr/lpp/ispf  
_CMDSERV_CONF_HOME    = /etc/rdz  
_CMDSERV_WORK_HOME    = /var/rdz  
-----
```

```
*** CHECK : USS MODULES  
Checking ISPF Directory : /usr/lpp/ispf  
Checking modules in /usr/lpp/ispf/bin directory  
Checking for ISPF configuration file ISPF.conf  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
*** CHECK : TSO/ISPF INITIALIZATION  
( TSO/ISPF session will be initialized )  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
*** CHECK: Shutting down TSO/ISPF IVP session  
RC=0  
MSG: SUCCESSFUL
```

```
-----  
Host installation verification completed successfully  
-----
```

Anmerkung: Falls eine der ISPF-Überprüfungen fehlschlägt, werden detaillierte Informationen angezeigt.

Der Befehl fekfivpi kann mit den folgenden optionalen, nicht positionsgebundenen Parametern verwendet werden:

-file Der Befehl fekfivpi kann umfangreiche Ausgaben (mit Hunderten von Zeilen) erzeugen. Der Parameter **-file** sendet diese Ausgabe an eine Datei `userlog/.eclipse/RSE$LOGNAME/fekfivpi.log`. Hier steht `userlog` für den Wert der Anweisung `user.log` in `rsed.envvars` und `$LOGNAME` für Ihre Benutzer-ID (in Großbuchstaben). Wenn die Anweisung `user.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird Ihr Ausgangspfad verwendet. Der Ausgangspfad wird in Ihrem OMVS-Sicherheitssegment definiert.

-debug

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

Verbindung zu TSO Commands Service mit APPC (optional)

Überprüfen Sie die Verbindung mit dem TSO-Befehlsserver (bei Verwendung von APPC), indem Sie den folgenden Befehl ausführen. Ersetzen Sie USERID durch eine gültige Benutzer-ID:

```
fekfivpa USERID
```

Nach einer Aufforderung zur Kennworteingabe sollte der Befehl den APPC-Dialog zurückgeben. Sehen Sie sich dazu das folgende Beispiel an (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpa USERID
Enter password:
```

```
Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
20070607 13:57:18.584060 /usr/lpp/rdz/bin/fekfscmd: version=Oct 2003
20070607 13:57:18.584326 Input parms: 1.2.3.4 * NOTRACE USERID *****
20070607 13:57:18.586800 APPC: Allocate succeeded
20070607 13:57:18.587022 Conversation id is 0DDBD3F80000000D
20070607 13:57:18.587380 APPC: Set Send Type succeeded
20070607 13:57:26.736674 APPC: Confirm succeeded
20070607 13:57:26.737027 Req to send recd value is 0
20070607 13:57:26.737546 APPC: SEND_DATA return_code = 0
20070607 13:57:26.737726 request_to_send_received = 0
20070607 13:57:26.737893 Send Data succeeded
20070607 13:57:26.738169 APPC: Set Prepare to Receive type succeeded
20070607 13:57:26.738580 APPC: Prepare to Receive succeeded
20070607 13:57:26.808899 APPC: Receive data
20070607 13:57:26.809122 RCV return_code = 0
20070607 13:57:26.809270 RCV data_received= 2
20070607 13:57:26.809415 RCV received_length= 29
20070607 13:57:26.809556 RCV status_received= 4
20070607 13:57:26.809712 RCV req_to_send= 0
20070607 13:57:26.809868 Receive succeeded
:IP: 0 9.42.112.75 1674 50246
20070607 13:57:26.810533 APPC: CONFIRMED succeeded
```

SCLMDT-Verbindung (optional)

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

```
fekfivps
```

Der Befehl sollte die Ergebnisse der auf SCLM Developer Toolkit bezogenen Überprüfungen zurückgeben (Variablen, HFS-Module, REXX-Laufzeit, Start und Stopp der TSO/ISPF-Sitzung). Vergleichen Sie hierzu das folgende Beispiel (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivps
```

```
Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

/etc/rdz/ISPF.conf content:

isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
sysproc=ISP.SISPCLIB,FEK.SFEKPROC

Host install verification for RSE
Review IVP log messages from HOST below :

*** CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

Server PATH = /usr/lpp/java/J5.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:
/bin:/usr/sbin:.

STEPLIB = FEK.SFEKAUTH:FEK.SFEKLOAD

_CMDSERV_BASE_HOME = /usr/lpp/ispf
_CMDSERV_CONF_HOME = /etc/rdz
_CMDSERV_WORK_HOME = /var/rdz
_SCLMDT_CONF_HOME = /var/rdz/sclmdt
_SCLMDT_WORK_HOME = /var/rdz
_SCLMDT_TRANTABLE = FEK.#CUST.LSTRANS.FILE

*** CHECK : JAVA PATH SETUP VERIFICATION

RC=0
MSG: SUCCESSFUL

*** CHECK : USS MODULES

Checking ISPF Directory : /usr/lpp/ispf
Checking modules in /usr/lpp/ispf/bin directory
Checking for ISPF configuration file ISPF.conf
Checking install bin Directory : /usr/lpp/rdz/bin
RC=0
MSG: SUCCESSFUL

*** CHECK : REXX RUNTIME ENVIRONMENT

RC=0
MSG: SUCCESSFUL

*** CHECK : TSO/ISPF INITIALIZATION

(TSO/ISPF session will be initialized)
RC=0
MSG: SUCCESSFUL

*** CHECK: Shutting down TSO/ISPF IVP session

RC=0
MSG: SUCCESSFUL

Host installation verification completed successfully

Anmerkung: Falls eine der SCLMDT-Überprüfungen fehlschlägt, werden detaillier-
te Informationen angezeigt.

Der Befehl fekfivps kann mit den folgenden optionalen, nicht positionsgebundenen Parametern verwendet werden:

- file** Der Befehl fekfivps kann umfangreiche Ausgaben (mit Hunderten von Zeilen) erzeugen. Der Parameter -file sendet diese Ausgabe an eine Datei userlog/.eclipse/RSE/\$LOGNAME/fekfivps.log. Hier steht userlog für den Wert der Anweisung user.log in rsed.envvars und \$LOGNAME für Ihre Benutzer-ID (in Großbuchstaben). Wenn die Anweisung user.log in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird Ihr Ausgangspfad verwendet. Der Ausgangspfad wird in Ihrem OMVS-Sicherheitssegment definiert.
- debug** Der Parameter -debug erstellt eine detaillierte Testausgabe. Verwenden Sie diese Option nur auf Anweisung des IBM Support Center.

REXEC-Verbindung (optional)

Führen Sie den folgenden Befehl aus, um die REXEC-Verbindung zu überprüfen. Ersetzen Sie 512 durch den von REXEC verwendeten Port und USERID durch eine gültige Benutzer-ID.

```
fekfivpr 512 USERID
```

Nachdem der Befehl Sie zur Eingabe eines Kennworts aufgefordert hat, sollte er den REXEC-Trace, eine Warnung zum Zeitlimit, die Java-Version und die RSE-Servernachricht zurückgeben. Sehen Sie sich hierzu das folgende Beispiel an (" \$" ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpr 512 USERID
Enter password:
```

```
Wed Jul 2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
$ EZYRC01I Calling function rexec_af with the following:
EZYRC02I Host: CDFMVS08, user USERID, cmd cd /etc/rdz;export RSE_USER_ID=USERI
D;./server.zseries -ivp, port 512
EZYRC19I Data socket = 4, Control socket = 6.
```

```
RSE server IVP test
```

```
CDFMVS08 -- Wed Jul 2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
```

```
RSE configuration files located in /etc/rdz --default
```

```
RSE userid is USERID --default
```

```
-----
Address Space size limits
```

```
-----
current address space size limit is 2147483647 (2048.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----
service history
```

```
-----
Fri Jun 19 00:01:00 2009 -- COPY -- HHOP760 v7600 created 18 Jun 2009
-----
```

expect to see time out messages after a successful IVP test

```
-----
starting RSE server in background -- Fri Jun 19 15:59:05 EDT 2009
-----
java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition (build pmz31dev-20070201 (SR4))
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmmz3123-20070201 (JI
T enabled)
J9VM - 20070131_11312_bHdSMr
JIT - 20070109_1805ifx1_r8
GC - 200701_09)
JCL - 20070126

DStore Server Starting...
Server Started Successfully
8108
Server running on: CDFMVS08
```

Anmerkung:

- Falls Sie keine Java- und RSE-Serverausgabe empfangen, ist wahrscheinlich die INETD-Region zu klein. (Beim Start von einer TSO/OMVS-Shellsitzung aus muss die Regionsgröße bei mindestens 2096128 liegen und beim Start mit BPXBATCH bei 0.)
- Das von REXEC verwendete Shell-Script können Sie gesondert testen. Eine diesbezügliche Beschreibung enthält der nächste IVP-Test im Abschnitt „REXEC/SSH-Shell-Script (optional)“.
- Der Server wird gestartet, ohne dass ein Client versucht, eine Verbindung herzustellen, sodass das Zeitlimit (von fünf Sekunden) überschritten wird. Diese Überschreitung führt zu einer Verbindungsfehlernachricht, die so ähnlich wie das folgende Beispiel aussieht:

```
Connection error
Server: error initializing socket: java.net.SocketTimeoutException:
Accept timed out
```

REXEC/SSH-Shell-Script (optional)

Wenn Sie den vorherigen IVP-Test aus dem Abschnitt „REXEC-Verbindung (optional)“ auf Seite 120 erfolgreich beendet haben, können Sie diesen Test überspringen.

Führen Sie den folgenden Befehl aus, um das von der REXEC- und SSH-Verbindung verwendete Shell-Script zu überprüfen:

```
fekfivpz
```

Der Befehl sollte eine Warnung zum Zeitlimit, die Java-Version und die RSE-Servernachricht zurückgeben. Sehen Sie sich hierzu das folgende Beispiel an (" \$" ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpz
```

```
Wed Jul 2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
```

```
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```


RSE server IVP test

CDFMVS08 -- Wed Jul 2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)

RSE configuration files located in /etc/rdz --default
RSE userid is USERID --default

Address Space size limits

current address space size limit is 2147483647 (2048.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

service history

Fri Jun 19 00:01:00 2009 -- COPY -- HHOP760 v7600 created 18 Jun 2009

expect to see time out messages after a successful IVP test

starting RSE server in background -- Fri Jun 19 15:59:05 EDT 2009

java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition (build pmz31dev-20070201 (SR4))
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmmz3123-20070201 (JIT enabled))
J9VM - 20070131_11312_bHdSMr
JIT - 20070109_1805ifx1_r8
GC - 200701_09)
JCL - 20070126

DStore Server Starting...
Server Started Successfully
8108
Server running on: CDFMVS08

Anmerkung:

- Falls keine Ausgabe angezeigt wird, ist Ihre (TSO-)Region wahrscheinlich zu klein. (Die Regionsgröße muss bei 2.096.128 liegen.)
- Der Server wird gestartet, ohne dass ein Client versucht, eine Verbindung herzustellen, sodass das Zeitlimit (von fünf Sekunden) überschritten wird. Diese Überschreitung führt zu einer Verbindungsfehlernachricht, die so ähnlich wie das folgende Beispiel aussieht:

Connection error
Server: error initializing socket: java.net.SocketTimeoutException:
Accept timed out

Teil 2. Informationen zu Developer for System z

Kapitel 8. Bedienerbefehle

In diesem Kapitel erhalten Sie einen Überblick über die für Developer for System z verfügbaren Bedienerbefehle (oder Konsolbefehle). Wenn Sie sich nicht mit den Syntaxdiagrammen auskennen, die zur Erläuterung des Befehlsformats verwendet werden, lesen Sie die Informationen im Abschnitt „Hinweise zum Lesen eines Syntaxdiagramms“ auf Seite 134.

Start (S)

Mit dem Befehl **START** können Sie eine gestartete Task (STC) dynamisch starten. Die abgekürzte Fassung des Befehls ist der Buchstabe S.

JES Job Monitor

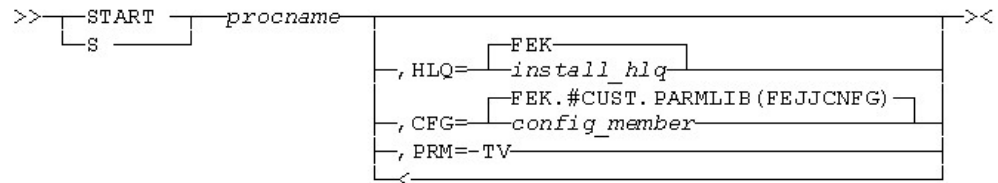


Abbildung 33. Bedienerbefehl 'START JMON'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der während der Hostkonfiguration verwendete Standardname ist JMON.

HLQ=HLQ für Installation

Für die Installation von Developer for System z verwendetes übergeordnetes Qualifikationsmerkmal. Die Standardeinstellung ist FEK.

CFG=Konfigurationsmember

Absoluter Name der Datei und des Members für die Konfigurationsdatei von JES Job Monitor. Die Standardeinstellung ist FEK.#CUST.PARMLIB(FEJJCNFG).

PRM=-TV

Aktivieren des ausführlichen Modus (Trace-Modus). Der Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSE-Dämon

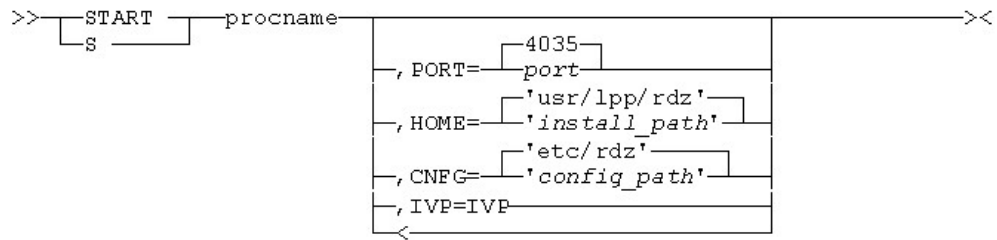


Abbildung 34. Bedienerbefehl 'START RSED'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der während der Hostkonfiguration verwendete Standardname lautet RSED.

PORT=Port

Der Port des RSE-Dämons, zu dem Clients eine Verbindung herstellen. Die Standardeinstellung ist 4035.

HOME='Installationspfad'

Pfadpräfix und der obligatorische Pfad /usr/lpp/rdz für die Installation von Developer for System z. Die Standardeinstellung ist '/usr/lpp/rdz'. Beachten Sie, dass beim z/OS UNIX-Pfad die Groß-/Kleinschreibung beachtet werden muss und dass der Pfad in Hochkommata (') eingeschlossen werden muss, damit Kleinbuchstaben erhalten bleiben.

CNFG='Konfigurationspfad'

Absolute Position der unter z/OS UNIX gespeicherten Konfigurationsdateien. Die Standardeinstellung ist '/etc/rdz'. Beachten Sie, dass beim z/OS UNIX-Pfad die Groß-/Kleinschreibung beachtet werden muss und dass der Pfad in Hochkommata (') eingeschlossen werden muss, damit Kleinbuchstaben erhalten bleiben.

IVP=IVP

Server nicht starten, sondern das Installationsprüfprogramm für den RSE-Dämon ausführen

Sperrdämon

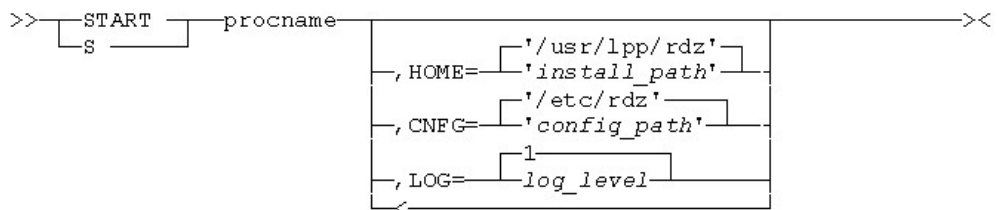


Abbildung 35. Bedienerbefehl 'START LOCKD'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der während der Hostkonfiguration verwendete Standardname lautet LOCKD.

HOME='Installationspfad'

Pfadpräfix und der obligatorische Pfad /usr/lpp/rdz für die Installation von Developer for System z. Die Standardeinstellung ist '/usr/lpp/rdz'. Beachten Sie, dass beim z/OS UNIX-Pfad die Groß-/Kleinschreibung beachtet werden muss und dass der Pfad in Hochkommata (') eingeschlossen werden muss, damit Kleinbuchstaben erhalten bleiben.

CNFG='Konfigurationspfad'

Absolute Position der unter z/OS UNIX gespeicherten Konfigurationsdateien. Die Standardeinstellung ist '/etc/rdz'. Beachten Sie, dass beim z/OS UNIX-Pfad die Groß-/Kleinschreibung beachtet werden muss und dass der Pfad in Hochkommata (') eingeschlossen werden muss, damit Kleinbuchstaben erhalten bleiben.

LOG=Protokollstufe

Der Detaillierungsgrad für die Ausgabe in DD STDOUT.

- 0 : Nur Fehlernachrichten protokollieren
- 1 : Fehlernachrichten und Warnungen protokollieren (Standardeinstellung)
- 2 : Fehlernachrichten, Warnungen und Informationsnachrichten protokollieren

Modify (F)

Mit dem Befehl **MODIFY** können Sie Kenndaten einer aktiven Task dynamisch abfragen und ändern. Die abgekürzte Fassung des Befehls ist der Buchstabe F.

JES Job Monitor

```
>> [MODIFY] [procname] [ , APPL=-TV ] ><
      [F] [ ] [ , APPL=-TN ]
```

Abbildung 36. Bedienerbefehl 'MODIFY JMON'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der während der Hostkonfiguration verwendete Standardname ist JMON.

- TV Aktivieren des ausführlichen Modus (Trace-Modus). Der Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.
- TN Inaktivieren des ausführlichen Modus (Trace-Modus).

RSE-Dämon

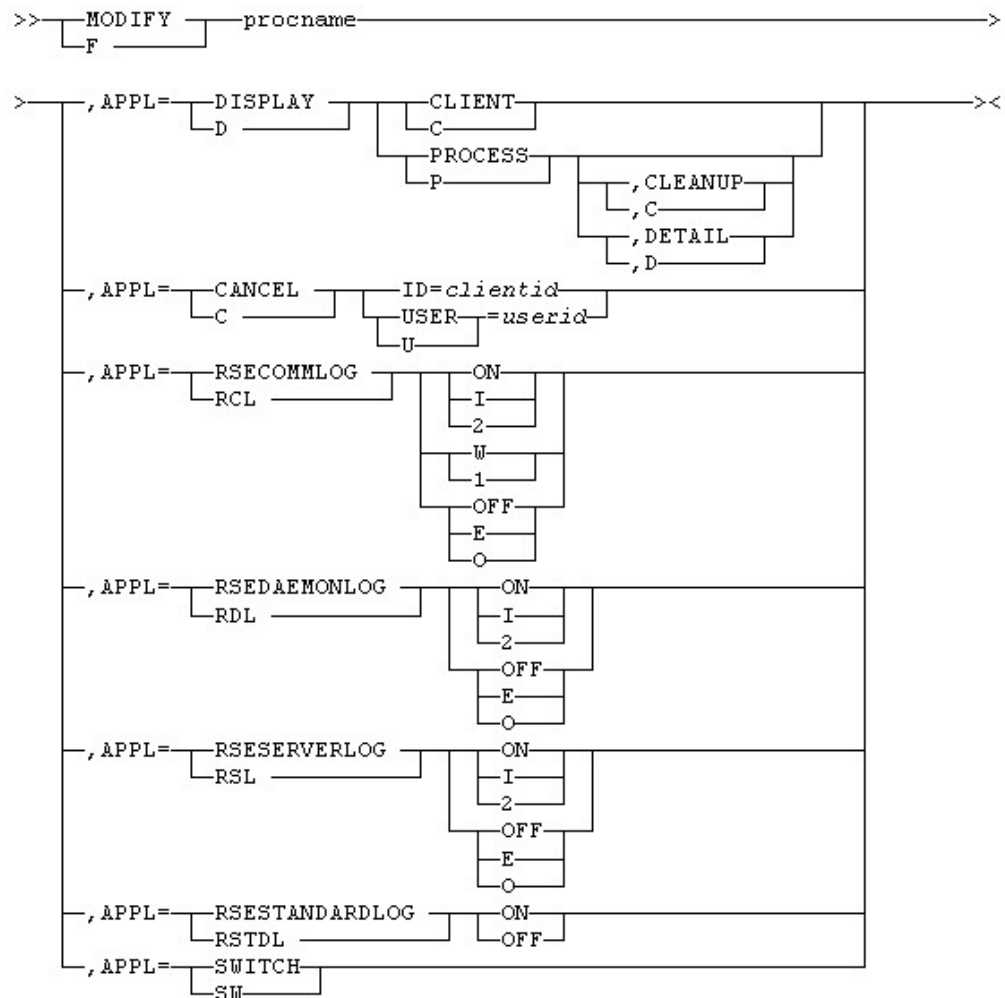


Abbildung 37. Bedienerbefehl 'MODIFY RSED'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der während der Hostkonfiguration verwendete Standardname lautet RSED.

DISPLAY CLIENT

Anzeigen der aktiven Clients

<Client-ID> : <Benutzer-ID> : <verbunden seit>

DISPLAY PROCESS[,CLEANUP,DETAIL]

Anzeigen der RSE-Thread-Pool-Prozesse. Für den Lastausgleich der verbundenen Benutzer kann es mehrere aktive Prozesse geben.

ProcessId(<Prozess-ID>) Memory Usage(<Belegung des Java-Heapspeichers>%)
Clients(<Anzahl der Clients>) Order(<Startreihenfolge>) <Fehlerstatus>

Anmerkung:

- <Prozess-ID> kann in prozessspezifischen z/OS UNIX-Bedienerbefehlen verwendet werden.

- Jeder Prozess hat seinen eigenen Java-Heapspeicher, dessen Größe in `rsed.envvars` festgelegt werden kann.
- `<Startreihenfolge>` ist eine fortlaufende Zahl, die die Reihenfolge angibt, in der die Thread-Pools gestartet wurden. Die Zahl entspricht der Zahl im Dateinamen der Dateien `stderr.*.log` und `stdout.*.log`.

In normalen Situationen ist `<Fehlerstatus>` leer. In Tabelle 18 sind die möglichen, nicht leeren Werte für `<Fehlerstatus>` dokumentiert.

Tabelle 18. Fehlerstatus des Thread-Pools

Status	Beschreibung
severe error	Der Thread-Pool-Prozess hat einen nicht behebbaren Fehler festgestellt und die Operationen angehalten. In den anderen Statusfeldern werden die letzten bekannten Werte angezeigt. Verwenden Sie die Option <code>CLEANUP</code> des Änderungsbefehls DISPLAY PROCESS , um diesen Eintrag aus der Tabelle zu entfernen.
killed process	Der Thread-Pool-Prozess wurde durch Java, z/OS UNIX oder einen Bedienerbefehl abgebrochen. In den anderen Statusfeldern werden die letzten bekannten Werte angezeigt. Verwenden Sie die Option <code>CLEANUP</code> des Änderungsbefehls DISPLAY PROCESS , um diesen Eintrag aus der Tabelle zu entfernen.
timeout	Der Thread-Pool-Prozess hat dem RSE-Dämon während einer Clientverbindungsanforderung nicht zeitnah geantwortet. In den anderen Statusfeldern werden die aktuellen Werte angezeigt. Der Thread-Pool wird in zukünftigen Clientverbindungsanforderungen ausgeschlossen. Der Status <code>*timeout*</code> wird zurückgesetzt, wenn sich ein Client abmeldet, der von diesem Thread-Pool bereitgestellt wurde.

Es werden weitere Informationen bereitgestellt, wenn die Option "DETAIL" des Änderungsbefehls **DISPLAY PROCESS** verwendet wird:

```

ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
PROCESS LIMITS:  CURRENT  HIGHWATER  LIMIT
  JAVA HEAP USAGE(%)    10      56      100
    CLIENTS              0       25       60
  MAXFILEPROC           83      103    64000
  MAXPROCUSER           97       99     200
    MAXTHREADS           9       14     1500
  MAXTHREADTASKS        9       14     1500

```

Das Feld 'ASId' ist die Adressraum-ID in Hexadezimalschreibweise. Die Tabelle zum Verarbeitungslimit zeigt die aktuelle Ressourcennutzung, die obere Grenze für die Ressourcennutzung und die Ressourcengrenze an.

Beachten Sie, dass die definierte Grenze aufgrund von anderen Begrenzungsfaktoren möglicherweise nie erreicht wird.

CANCEL ID=Client-ID

Abbrechen der Clientverbindung auf der Basis der Client-ID, die im Modifizierungsbefehl **DISPLAY CLIENT** angegeben ist

CANCEL USER=Benutzer-ID

Abbrechen der Clientverbindung auf der Basis der Benutzer-ID des Clients, die im Modifizierungsbefehl **DISPLAY CLIENT** angegeben ist

RSECOMMLOG {ON,OFF,I,W,E,2,1,0}

Steuert die Tracedetailstufe für den RSE-Server (rsecomm.log) und die MVS-Dateiservices (lock.log und ffs*.log). Die Standardeinstellung wird beim Start in rsecomm.properties definiert. Drei Detaillierungsgrade sind verfügbar:

E oder 0 oder OFF	Nur Fehlnachrichten
W oder 1	Fehlnachrichten und Warnungen. Dies ist die Standardeinstellung in rsecomm.properties.
I oder 2 oder ON	Fehlnachrichten, Warnungen und Informationsnachrichten

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSEDAEMONLOG {ON,OFF,I,E,2,0}

Steuert die Tracedetailstufe für den RSE-Dämon (rsedaemon.log). Die Standardeinstellung wird beim Start in rsecomm.properties definiert. Zwei Detaillierungsgrade sind verfügbar:

E oder 0 oder OFF	Nur Fehlnachrichten
I oder 2 oder ON	Fehlnachrichten, Warnungen und Informationsnachrichten

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSESERVERLOG {ON,OFF,I,E,2,0}

Steuert die Tracedetailstufe für die RSE-Thread-Pools (rseserver.log). Die Standardeinstellung wird beim Start in rsecomm.properties definiert. Zwei Detaillierungsgrade sind verfügbar:

E oder 0 oder OFF	Nur Fehlnachrichten
I oder 2 oder ON	Fehlnachrichten, Warnungen und Informationsnachrichten

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

RSESTANDARDLOG {ON,OFF}

Inaktiviert (OFF) oder aktiviert (ON) die Aktualisierung der Protokolldateien mit den Datenströmen 'stdout' und 'stderr' der Thread-Pools (stdout*.log und stderr*.log). Die Standardeinstellung wird beim Start durch die Anweisung enable.standard.log in rsed.envvars definiert.

Ein detaillierter Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden.

SWITCH

Wechsel zu einer neuen Prüfprotokolldatei

Anmerkung:

- Weitere Informationen zu den oben genannten Protokolldateien enthält der Abschnitt „Protokolldateien“ auf Seite 138 in Kapitel 9, „Konfigurationsprobleme lösen“, auf Seite 137.
- Weitere Informationen zu Prüffunktionen enthält der Abschnitt „Prüfprotokollierung“ auf Seite 165 in Kapitel 10, „Sicherheitsaspekte“, auf Seite 159.

Sperrdämon

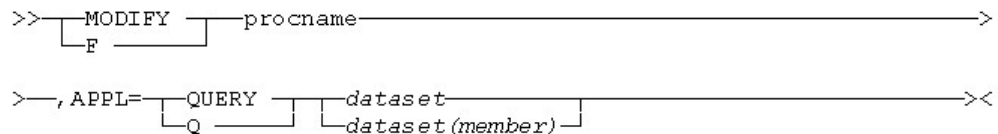


Abbildung 38. Bedienerbefehl 'MODIFY LOCKD'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Der während der Hostkonfiguration verwendete Standardname lautet LOCKD.

QUERY dataset[(Member)]

Fragt den Sperrstatus der aufgelisteten Datei oder des aufgelisteten Members ab. Der Server antwortet mit einer der folgenden Nachrichten:

```
BPXM023I (stclock) Datei[(Member)] NOT LOCKED
BPXM023I (stclock) Datei[(Member)] LOCKED BY Benutzer-ID
```

Anmerkung:

- Der Server berichtet auch Sperren von anderen Produkten, wie z. B. ISPF.
- Für Sperren durch Clients von Developer for System z, die keine Registrierung mit dem Sperrdämon durchführen konnten, wird der Adressraum des Thread-Pool-Servers (RSEDx) als Sperreneigentümer angegeben.

Wenn es dem RSE-Server nicht möglich ist, den Client mit dem Sperrdämon zu registrieren, wird die Konsolnachricht FEK513W generiert. Die in dieser Nachricht aufgeführten ASID- und TCB-Werte können mit der Ausgabe des Bedienerbefehls **D GRS,RES=(*,Datei[(Member))]** verglichen werden, um zu ermitteln, welcher derzeitige Benutzer die Sperre hält.

Stop (P)

Mit dem Befehl **STOP** können Sie eine aktive Task stoppen. Die abgekürzte Fassung des Befehls ist der Buchstabe P.

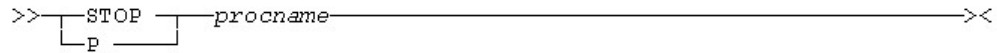


Abbildung 39. Bedienerbefehl 'STOP'

Prozedurname

Der Name des Members in einer Prozedurenbibliothek, mit dem der Server gestartet wird. Während der Hostkonfiguration wird für JES Job Monitor, den RSE-Dämon und den Sperrdämon jeweils der Standardname JMON, RSED und LOCKD verwendet.

Konsolnachrichten

JES Job Monitor

Es gibt keine produktspezifischen Konsolnachrichten für JES Job Monitor. Der Server greift für Aktionen, die von der Clientkomponente von Developer for System z ausgeführt werden, auf die von z/OS und JES generierten Konsolnachrichten zurück.

RSE-Dämon, RSE-Thread-Pool-Server und Sperrdämon

In Tabelle 19 werden die produktspezifischen Konsolnachrichten aufgelistet, die vom RSE-Dämon, vom RSE-Thread-Pool-Server und vom Sperrdämon generiert werden.

Tabelle 19. RSE-Konsolnachrichten

Nachrichten-ID	Nachrichtentext
FEK001I	RseDaemon being initialized in {0} bit mode
FEK002I	RseDaemon started. (port={0})
FEK003I	Stop command being processed
FEK004I	RseDaemon: Max Heap Size={0}MB and private AS Size={1}MB
FEK005I	Server process started. (processId={0})
FEK009I	RseDaemon is waiting for the server process to start.
FEK010I	(rsed.envvars location = {0})
FEK011I	(log directory = {0})
FEK100E	Daemon port/timeout value must be digits
FEK101E	JRE {0} or higher required
FEK102E	Invalid arguments received: {0}
FEK103E	Almost Disk-Full in {0}
FEK104E	Maximum number of processes has been reached
FEK105E	Error in sending audit data (rc={0})
FEK110E	socket() failed. reason=({0})
FEK111E	setsockopt() failed. reason=({0})
FEK112E	bind() failed. reason=({0})
FEK113E	listen() failed. reason=({0})
FEK114E	accept() failed. reason=({0})
FEK115E	write() failed. reason=({0})
FEK116E	pipe() failed. reason=({0})

Tabelle 19. RSE-Konsolnachrichten (Forts.)

Nachrichten-ID	Nachrichtentext
FEK117E	socketpair() failed. reason=({0})
FEK118E	select() failed. reason=({0})
FEK119E	_console() failed. reason=({0})
FEK130E	gsk_environment_open() failed. reason=({0})
FEK131E	gsk_attribute_set_enum(GSK_PROTOCOL_SSLV2) failed. reason=({0})
FEK132E	gsk_attribute_set_enum(GSK_PROTOCOL_SSLV3) failed. reason=({0})
FEK133E	gsk_attribute_set_enum(GSK_PROTOCOL_TLSV1) failed. reason=({0})
FEK134E	gsk_attribute_set_buffer(GSK_KEYRING_FILE) failed. reason=({0})
FEK135E	gsk_attribute_set_buffer(GSK_KEYRING_PW) failed. reason=({0})
FEK136E	gsk_environment_init() failed. reason=({0})
FEK137E	gsk_secure_socket_open() failed. reason=({0})
FEK138E	gsk_attribute_set_numeric_value(GSK_FD) failed. reason=({0})
FEK139E	gsk_attribute_set_buffer(GSK_KEYRING_LABEL) failed. reason=({0})
FEK140E	gsk_attribute_set_enum(GSK_SESSION_TYPE) failed. reason=({0})
FEK141E	gsk_attribute_set_callback(GSK_IO_CALLBACK) failed. reason=({0})
FEK142E	gsk_secure_socket_init() failed. reason=({0})
FEK143E	gsk_attribute_set_enum(GSK_CLIENT_AUTH_TYPE) failed. reason=({0})
FEK144E	gsk_get_cert_info failed. reason=({0})
FEK145E	gsk_secure_socket_read() failed. reason=({0})
FEK146E	gsk_secure_socket_write() failed. reason=({0})
FEK150E	RseDaemon abnormally terminated; {0}
FEK201I	{0} Command has been processed
FEK202E	Invalid Command Entered
FEK203E	Invalid Display Command: Display Process Client
FEK204E	Invalid Cancel Command: Cancel ID= User=
FEK205E	Command was not processed owing to consecutive SWITCHs
FEK206E	Audit Log facility is not active
FEK207I	No Client to be displayed
FEK208I	{0} canceled
FEK209I	No Process to be displayed
FEK210I	{0} canceled owing to duplicate logon
FEK501I	Lock daemon started, port={0}, cleanup interval={1}, log level={2}
FEK502I	Lock daemon terminating
FEK510E	Lock daemon, missing port
FEK511E	Lock daemon, wrong port, port={0}
FEK512E	Lock daemon, socket error, port={0}
FEK513W	Lock daemon, registration failed, ASID={0}, TCB={1}, USER={2}
FEK514W	Lock daemon, wrong log level, log level={0}
BPXM023I	(stclock) Datei[(Member)] NOT LOCKED

Tabelle 19. RSE-Konsolnachrichten (Forts.)

Nachrichten-ID	Nachrichtentext
BPXM023I	(stclock) Datei[(Member)] LOCKED BY Benutzer-ID
BPXM023I	(stclock) Befehl, WRONG COMMAND
BPXM023I	(stclock) Befehl, MISSING ARGUMENT
BPXM023I	(stclock) Argument, WRONG ARGUMENT

Hinweise zum Lesen eines Syntaxdiagramms

Das Syntaxdiagramm zeigt Ihnen, wie ein Befehl angegeben werden muss, damit das Betriebssystem Ihre Eingabe ordnungsgemäß interpretieren kann. Das Syntaxdiagramm wird von links nach rechts und von oben nach unten gelesen. Folgen Sie dabei der horizontalen Linie (dem Hauptpfad).

Symbole

In Syntaxdiagrammen werden die folgenden Symbole verwendet:

Symbol	Beschreibung
>>	Markiert den Anfang des Syntaxdiagramms
>	Zeigt an, dass das Syntaxdiagramm fortgesetzt wird
	Markiert Anfang und Ende eines Fragments oder Abschnitts des Syntaxdiagramms
><	Markiert das Ende des Syntaxdiagramms

Operanden

In Syntaxdiagrammen werden die folgenden Arten von Operanden verwendet:

- Erforderliche Operanden werden auf der Linie des Hauptpfads angezeigt:
 >>—ERFORDERLICHER_OPERAND—><
- Optionale Operanden werden unterhalb der Linie des Hauptpfads angezeigt:
 >>
 |
 |—OPTIONALER_OPERAND—|
 ><
- Standardoperanden werden oberhalb der Linie des Hauptpfads angezeigt:
 >>
 |
 |—STANDARDOPERAND—|
 ><

Operanden werden als Schlüsselwörter oder Variablen klassifiziert:

- Schlüsselwörter sind Konstanten, die angegeben werden müssen. Erscheint das Schlüsselwort im Syntaxdiagramm in gemischter Groß- und Kleinschreibung, gibt der Abschnitt in Großschreibung die Abkürzung für das Schlüsselwort an (z. B. KEYword). Bei Schlüsselwörtern wird die Groß-/Kleinschreibung nicht unterschieden. Sie können Sie in Großbuchstaben oder in Kleinbuchstaben angeben.
- Variablen sind kursiv angegeben. Sie erscheinen in Kleinbuchstaben und repräsentieren Namen oder Werte, die Sie angeben müssen. Ein Dateiname ist beispielsweise eine Variable. Bei Variablen muss unter Umständen die Groß-/Kleinschreibung beachtet werden.

Syntaxbeispiel

Im folgenden Beispiel ist der Befehl USER ein Schlüsselwort. Der erforderliche variable Parameter ist Benutzer-ID, und der optionale variable Parameter ist Kennwort. Ersetzen Sie die variablen Parameter durch Ihre eigenen Werte:

>>—USER—*Benutzer-ID*—Kennwort—><

Nicht alphanumerische Zeichen und Leerzeichen

Wenn ein Diagramm ein Zeichen enthält, das kein alphanumerisches Zeichen ist (z. B. Klammern, Punkte, Kommata, Gleichheitszeichen und Leerzeichen), müssen Sie das Zeichen als Teil der Syntax eingeben. In diesem Beispiel muss die Eingabe OPERAND=(001 0.001) lauten:

>>—OPERAND—=(—001— —0.001—)—><

Mehrere Operanden auswählen

Ein nach links weisender Pfeil in einer Gruppe von Operanden bedeutet, dass mehr als ein Operand ausgewählt oder ein einzelner Operand wiederholt werden kann:

>>—

←

→

WIEDERHOLBARER_OPERAND_1

WIEDERHOLBARER_OPERAND_2

—><

Mehrere Zeilen

Wenn ein Diagramm mehr als eine Zeile umfasst, endet die erste Zeile mit einer einzelnen Pfeilspitze und die zweite Zeile beginnt mit einer einzelnen Pfeilspitze:

>>—| Die erste Zeile des Syntaxdiagramms, das mehr als eine Zeile umfasst |—>
>—| Die Fortsetzung der Unterbefehle und/oder Parameter |——><

Syntaxfragmente

Einige Diagramme können Syntaxfragmente enthalten, die zur Unterteilung zu langer oder zu komplexer Diagramme bzw. von Diagrammen mit zu vielen Wiederholungen dienen. Die Namen von Syntaxfragmenten sind in gemischter Groß-/Kleinschreibung angegeben und erscheinen im Diagramm sowie in der Überschrift des Diagramms. Das Fragment ist unterhalb des Hauptdiagramms dargestellt:

>>—| Syntaxfragment |——><

Syntaxfragment:

|—ERSTER_OPERAND—,—ZWEITER_OPERAND—,—DRITTER_OPERAND—|

Kapitel 9. Konfigurationsprobleme lösen

Dieses Kapitel soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von Developer for System z auftreten könnten. Es enthält die folgenden Abschnitte:

- „Protokoll- und Konfigurationsanalyse mit FEKLOGS“
- „Protokolldateien“ auf Seite 138
- „Speicherauszugsdateien“ auf Seite 144
- „Traceerstellung“ auf Seite 146
- „z/OS UNIX-Berechtigungsbits“ auf Seite 149
- „Reservierte TCP/IP-Ports“ auf Seite 152
- „Größe des Adressraums“ auf Seite 153
- „APPC-Transaktion und TSO Commands Service“ auf Seite 155
- „Weitere Informationen“ auf Seite 156

Weitere Informationen sind im Bereich 'Support' der Website zu Developer for System z (<http://www-306.ibm.com/software/awdtools/rdz/support/>) verfügbar. In diesem Bereich finden Sie die aktuellsten technischen Hinweise des Unterstützungsteams.

Die aktuellste Version der Dokumentation zu Developer for System z, einschließlich White Papers und anderer hilfreicher Informationen, finden Sie im Abschnitt 'Library' der Website.

Im Information Center zu Developer for System z (<http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp>) ist die Clientkomponente von Developer for System z und ihre Interaktion mit dem Host (aus der Sicht des Clients) dokumentiert.

Wertvolle Informationen enthält auch die z/OS-Internetbibliothek mit der Adresse <http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/>.

Benachrichtigen Sie uns, wenn Sie denken, dass bestimmte Funktionen in Developer for System z fehlen. Unter der folgenden Adresse können Sie eine Erweiterungsanfrage (Request For Enhancement, RFE) öffnen:

<https://www.ibm.com/developerworks/support/rational/rfe/>

Protokoll- und Konfigurationsanalyse mit FEKLOGS

Developer for System z stellt einen Beispieljob, FEKLOGS, bereit, der alle z/OS UNIX-Protokolldateien sowie Installations- und Konfigurationsdaten für Developer for System z zusammenstellt.

Der Beispieljob FEKLOGS ist in FEK.#CUST.JCL enthalten, sofern Sie während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Die Anpassung von FEKLOGS wird in der JCL beschrieben. Die Anpassung schließt die Bereitstellung einiger Schlüsselvariablen ein.

Anmerkung: SDSF-Kunden können den Zeilenbefehl **XDC** in SDSF verwenden, um die Jobausgabe in einer Datei zu speichern, die an das IBM Support Center übergeben werden kann.

Protokolldateien

Developer for System z erstellt Protokolldateien, die Sie und das IBM Support Center bei der Feststellung und Lösung von Problemen unterstützen können. Nachfolgend sind die Protokolldateien, die auf Ihrem z/OS-Hostsystem erstellt werden können, übersichtlich aufgelistet. Überprüfen Sie neben diesen produktspezifischen Protokollen stets, ob das SYSLOG zugehörige Nachrichten enthält.

Nach MVS-basierten Protokollen kann über die entsprechende DD-Anweisung gesucht werden. z/OS UNIX-basierte Protokolldateien befinden sich in den folgenden Verzeichnissen:

- `userlog/$LOGNAME/`

Benutzerspezifische Protokolldateien werden in `userlog/$LOGNAME` gespeichert. Dabei ist `userlog` der kombinierte Wert der Anweisungen `user.log` und `DSTORE_LOG_DIRECTORY` in `rsed.envvars` und `$LOGNAME` ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung `user.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung `DSTORE_LOG_DIRECTORY` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird `.eclipse/RSE/` an den Wert von `user.log` angehängt.

- `.dstoreMemLogging` - Protokollierung der DataStore-Speicherbelegung
- `.dstoreTrace` - Protokollierung der DataStore-Aktionen
- `fa.log` - Protokoll der Fault Analyzer-Integration
- `fekfivpi.log` - Protokoll des IVP-Tests `fekfivpi`
- `fekfivps.log` - Protokoll des IVP-Tests `fekfivps`
- `ffs.log` - Protokoll des FFS-Servers (Foreign File System), der native MVS-Funktionen ausführt
- `ffsget.log` - Protokoll des Datei-Reader, der eine sequenzielle Datei oder ein PDS-Member liest
- `ffsput.log` - Protokoll des Datei-Writer, der eine sequenzielle Datei oder ein PDS-Member schreibt
- `lock.log` - Protokoll des Sperrenmanagers, der eine sequenzielle Datei oder ein PDS-Member sperrt bzw. freigibt
- `rmt_class_loader.cache.jar` - Cache der vom fernen RSE-Klassenlader geladenen Klassen
- `rsecomm.log` - Protokoll des RSE-Servers, der Befehle vom Client verarbeitet, sowie die Protokollierung der Kommunikation zwischen allen Services unter Beteiligung von RSE (Dieses Protokoll kann den Java-Stack-Trace für Ausnahmen enthalten.)
- `stderr.log` - Umgeleitete Daten von der Standardfehlerausgabe `stderr`
- `stdout.log` - Umgeleitete Daten von der Standardausgabe `stdout`

Anmerkung: Das Verzeichnis `.eclipse` und die Protokolldateien `.dstore*` beginnen mit einem Punkt (.) und sind dadurch verdeckt. Mit dem z/OS UNIX-Befehl **ls -lA** können Sie verdeckte Dateien und Verzeichnisse auflisten.

Wenn Sie mit der Clientkomponente von Developer for System z arbeiten, wählen Sie **Fenster > Benutzervorgaben... > Ferne Systeme > Dateien** aus und aktivieren Sie die Option "Verdeckte Dateien anzeigen".

- **Dämonenausgangsverzeichnis**

Die Protokolldateien des RSE-Dämons und RSE-Thread-Pools befinden sich im Dämonenausgangsverzeichnis. Das Dämonenausgangsverzeichnis steht hierbei für den Wert der Anweisung `daemon.log` in `rsed.envvars`. Wenn die Anweisung `daemon.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad der Benutzer-ID verwendet, die der gestarteten Task RSED zugeordnet ist. Das Ausgangsverzeichnis ist im Segment für die OMVS-Sicherheit der Benutzer-ID definiert.

- `rsedaemon.log` - Protokoll des RSE-Dämons
- `rseserver.log` - Protokoll des RSE-Thread-Pools
- `audit.log` - RSE-Prüfprotokoll
- `serverlogs.count` - Zähler zum Protokollieren von RSE-Thread-Pool-Datenströmen
- `stderr.*.log` - Standardfehlerdatenstrom des RSE-Thread-Pools
- `stdout.*.log` - Standardausgabedatenstrom des RSE-Thread-Pools

Anmerkung: Es gibt einige Bedienerbefehle, mit denen das in einige der erwähnten Protokolldateien geschriebene Datenvolumen gesteuert werden kann. Weitere Informationen hierzu enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.

Protokollierung von JES Job Monitor

- **SYSOUT DD**

Es werden normale Operationen protokolliert. Der Standardwert in der Beispiel-JCL `FEK.#CUST.PROCLIB(JMON)` ist `SYSOUT=*`.

- **SYSPRINT DD**

Trace-Protokollierung. Der Standardwert in der Beispiel-JCL `FEK.#CUST.PROCLIB(JMON)` ist `SYSOUT=*`. Der Trace wird mit dem Parameter `-TV` aktiviert. Weitere Details hierzu enthält der Abschnitt „Traceerstellung für JES Job Monitor“ auf Seite 146.

Protokollierung des Sperrdämons

- **STDOUT DD**

Umgeleitete Daten von der Java-Standardausgabe `stdout`. Der Standardwert in der Beispiel-JCL `FEK.#CUST.PROCLIB(LOCKD)` ist `SYSOUT=*`.

- **STDERR DD**

Umgeleitete Daten von der Java-Standardfehlerausgabe `stderr`. Der Standardwert in der Beispiel-JCL `FEK.#CUST.PROCLIB(LOCKD)` ist `SYSOUT=*`.

Protokollierung des RSE-Dämons und des Thread-Pools

- **STDOUT DD**

Umgeleitete Daten von der Java-Standardausgabe `stdout` des RSE-Dämons. Der Standardwert in der Beispiel-JCL `FEK.#CUST.PROCLIB(RSED)` ist `SYSOUT=*`.

- **STDERR DD**

Umgeleitete Daten von der Java-Standardfehlerausgabe `stderr` des RSE-Dämons. Der Standardwert in der Beispiel-JCL `FEK.#CUST.PROCLIB(RSED)` ist `SYSOUT=*`.

- **Dämonausgangsverzeichnis**

Die Protokolldateien des RSE-Dämons und des RSE-Thread-Pools befinden sich in `daemon-home`. Dabei steht `daemon-home` für den Wert der Anweisung `daemon.log` in `rsed.envvars`. Wenn die Anweisung `daemon.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad der Benutzer-ID verwendet, die der gestarteten Task RSED zugeordnet ist. Das Ausgangsverzeichnis ist im Segment für die OMVS-Sicherheit der Benutzer-ID definiert.

- `rsedaemon.log` - Protokoll des RSE-Dämons
- `rseserver.log` - Protokoll des RSE-Thread-Pools
- `audit.log` - RSE-Prüfprotokoll
- `serverlogs.count` - Zähler zum Protokollieren von RSE-Thread-Pool-Datenströmen
- `stderr.*.log` - Standardfehlerdatenstrom des RSE-Thread-Pools
- `stdout.*.log` - Standardausgabedatenstrom des RSE-Thread-Pools

Anmerkung:

- Die Dateien `serverlogs.count`, `stderr.*.log` und `stdout.*.log` werden nur erstellt, wenn die Anweisung `enable.standard.log` in `rsed.envvars` aktiv ist oder wenn die Funktion mit dem Bedienerbefehl **modify rsestandardlog on** dynamisch aktiviert wurde.
- * in `stderr.*.log` und `stdout.*.log` ist standardmäßig 1. Es kann allerdings mehrere RSE-Thread-Pools geben. In diesem Fall wird die Nummer für jeden neuen RSE-Thread-Pool erhöht, um eindeutige Dateinamen zu gewährleisten.
- Wenn die Anweisung `enable.standard.log` aktiv ist, gibt es keine benutzerspezifischen Protokolldateien `stdout.log` und `stderr.log`. Die benutzerspezifischen Daten werden jetzt in den entsprechenden RSE-Thread-Pool-Datenstrom geschrieben.
- Es gibt einige Bedienerbefehle, mit denen das in einige der erwähnten Protokolldateien geschriebene Datenvolumen gesteuert werden kann. Weitere Informationen hierzu enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.

Protokollierung des RSE-Benutzers

- **userlog/\$LOGNAME/**

Die RSE-bezogenen Komponenten erstellen diverse Protokolldateien. Alle Dateien werden in `userlog/$LOGNAME` gespeichert. Dabei ist `userlog` der kombinierte Wert der Anweisungen `user.log` und `DSTORE_LOG_DIRECTORY` in `rsed.envvars` und `$LOGNAME` ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung `user.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung `DSTORE_LOG_DIRECTORY` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird `.eclipse/RSE/` an den Wert von `user.log` angehängt.

- `.dstoreMemLogging` - Protokollierung der DataStore-Speicherbelegung
- `.dstoreTrace` - Protokollierung der DataStore-Aktionen
- `ffs.log` - Protokoll des FFS-Servers (Foreign File System), der native MVS-Funktionen ausführt
- `ffsget.log` - Protokoll des Datei-Reader, der eine sequenzielle Datei oder ein PDS-Member liest

- `ffsput.log` - Protokoll des Datei-Writer, der eine sequenzielle Datei oder ein PDS-Member schreibt
- `lock.log` - Protokoll des Sperrenmanagers, der eine sequenzielle Datei oder ein PDS-Member sperrt bzw. freigibt
- `rmt_class_loader.cache.jar` - Cache der vom fernen RSE-Klassenlader geladenen Klassen
- `rsecomm.log` - Protokoll des RSE-Servers, der Befehle vom Client verarbeitet, sowie die Protokollierung der Kommunikation zwischen allen Services unter Beteiligung von RSE (Dieses Protokoll kann den Java-Stack-Trace für Ausnahmen enthalten.)
- `stderr.log` - Umgeleitete Daten von der Standardfehlerausgabe `stderr`
- `stdout.log` - Umgeleitete Daten von der Standardausgabe `stdout`

Anmerkung:

- Das Verzeichnis `.eclipse` und die Protokolldateien `.dstore*` beginnen mit einem Punkt (.) und sind dadurch verdeckt. Mit dem z/OS UNIX-Befehl **ls -lA** können Sie verdeckte Dateien und Verzeichnisse auflisten. Wenn Sie mit der Clientkomponente von Developer for System z arbeiten, wählen Sie **Fenster > Benutzervorgaben... > Ferne Systeme > Dateien** aus und aktivieren Sie die Option "Verdeckte Dateien anzeigen".
- Die Erstellung der Protokolldateien `.dstore*` wird von den Java-Startoptionen `-DDSTORE_*` gesteuert. Lesen Sie hierzu die Informationen unter „Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren“ auf Seite 42.
- Die Protokolldateien `.dstore*` werden im ASCII-Format erstellt. Verwenden Sie den z/OS UNIX-Befehl **iconv -f ISO8859-1 -t IBM-1047 .dstore***, wenn Sie sie in EBCDIC (Codepage IBM-1047) anzeigen möchten.
- Wenn die Anweisung `enable.standard.log` aktiv ist, gibt es keine benutzerspezifischen Protokolldateien `stdout.log` und `stderr.log`. Die benutzerspezifischen Daten werden jetzt in den entsprechenden RSE-Thread-Pool-Datenstrom geschrieben.
- Es gibt einige Bedienerbefehle, mit denen das in einige der erwähnten Protokolldateien geschriebene Datenvolumen gesteuert werden kann. Weitere Informationen hierzu enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.

Protokollierung der Fault Analyzer-Integration

• `userlog/$LOGNAME/`

Protokollierung der Fault Analyzer-Integration. Dabei ist `userlog` der kombinierte Wert der Anweisungen `user.log` und `DSTORE_LOG_DIRECTORY` in `rsd.envvars` und `$LOGNAME` ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung `user.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung `DSTORE_LOG_DIRECTORY` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird `.eclipse/RSE/` an den Wert von `user.log` angehängt.

- `fa.log` - Protokoll der Fault Analyzer-Integration
- `rsecomm.log` - Kommunikationsprotokoll der Fault Analyzer-Integration

Protokollierung der File Manager-Integration

- **userlog/\$LOGNAME/rsecomm.log**

Protokollierung der Kommunikation für die File Manager-Integration. Dabei ist userlog der kombinierte Wert der Anweisungen user.log und DSTORE_LOG_DIRECTORY in rsed.envvars und \$LOGNAME ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung user.log in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung DSTORE_LOG_DIRECTORY in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird .eclipse/RSE/ an den Wert von user.log angehängt.

Protokollierung von SCLM Developer Toolkit

- **userlog/\$LOGNAME/rsecomm.log**

Protokollierung der Kommunikation für SCLM Developer Toolkit. Dabei ist userlog der kombinierte Wert der Anweisungen user.log und DSTORE_LOG_DIRECTORY in rsed.envvars und \$LOGNAME ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung user.log in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung DSTORE_LOG_DIRECTORY in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird .eclipse/RSE/ an den Wert von user.log angehängt.

CARMA-Protokollierung

- **CARMA-Server-Job**

Wenn Sie über die Batchschnittstelle eine Verbindung zu CARMA öffnen, startet FEK.#CUST.SYSPROC(CRASUBMT) einen Server-Job CRAport (mit der Benutzer-ID als Eigner). Die Angabe port im Namen steht hier für den verwendeten TCP/IP-Port.

- **CARMALOG DD**

Wenn in der ausgewählten CARMA-Startmethode die DD-Anweisung CARMALOG angegeben ist, wird die CARMA-Protokollierung an diese DD-Anweisung im Server-Job umgeleitet. Andernfalls ist sie auf der SYSPRINT-Karte enthalten.

- **SYSPRINT DD**

Die SYSPRINT-Karte des Server-Jobs enthält die CARMA-Protokollierung, sofern nicht die DD-Anweisung CARMALOG definiert ist.

- **userlog/\$LOGNAME/rsecomm.log**

Protokollierung der CARMA-Kommunikation. Dabei ist userlog der kombinierte Wert der Anweisungen user.log und DSTORE_LOG_DIRECTORY in rsed.envvars und \$LOGNAME ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung user.log in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung DSTORE_LOG_DIRECTORY in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird .eclipse/RSE/ an den Wert von user.log angehängt.

APPC-Transaktionsprotokollierung (TSO Commands Service)

- **SYSPRINT DD**

Wenn das APPC-Administrationsdienstprogramm ein Profil für ein Transaktionsprogramm hinzufügt und modifiziert, wird sowohl das Profil als auch die JCL auf Syntaxfehler überprüft. Die Ausgaben dieser Phase umfassen Nachrichten zu Syntaxfehlern im TP-Profil, Verarbeitungsnachrichten des Dienstprogramms und JCL-Konvertierungsanweisungen. Die Protokollierung von Nachrichten dieser Phase wird von der Anweisung SYSPRINT DD für das Dienstprogramm ATBSDFMU gesteuert. Der Standardwert in der Beispiel-JCL FEK.SFEKSAMP(FEKAPPCC) ist SYSOUT=*. Weitere Details hierzu enthält die Veröffentlichung *MVS Planning: APPC/MVS Management* (IBM Form SA22-7599).

- **&SYSUID.FEKFRSRV.&TPDATE.&TPTIME.LOG**

Wenn ein TP ausgeführt wird, werden die TP-Laufzeitnachrichten, z. B. Zuordnungs- und Beendigungsnachrichten, in das Protokoll geschrieben, das vom Schlüsselwort MESSAGE_DATA_SET im TP-Profil genannt wird. Der Standardwert in der Beispiel-JCL FEK.SFEKSAMP(FEKAPPCC) ist &SYSUID.FEKFRSRV.&TPDATE.&TPTIME.LOG. Weitere Details hierzu enthält die Veröffentlichung *MVS Planning: APPC/MVS Management* (IBM Form SA22-7599).

Anmerkung: Diese Protokolldatei erscheint möglicherweise erst, wenn Sie das Schlüsselwort KEEP_MESSAGE_LOG(ALWAYS) zu den Transaktionsdefinitionen hinzugefügt haben. Dies ist von Ihren APPC-Transaktionsdefinitionen und den Standards an Ihrem Standort abhängig. Weitere Details hierzu enthält die Veröffentlichung *MVS Planning: APPC/MVS Management* (IBM Form SA22-7599).

Protokollierung des IVP-Tests fekfivpi

- **userlog/\$LOGNAME/fekfivpi.log**

Ausgabe des Befehls fekfivpi -file (IVP-Test für das TSO/ISPF-Client-Gateway). Dabei ist userlog der kombinierte Wert der Anweisungen user.log und DSTORE_LOG_DIRECTORY in rsed.envvars und \$LOGNAME ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung user.log in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung DSTORE_LOG_DIRECTORY in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird .eclipse/RSE/ an den Wert von user.log angehängt.

Protokollierung des IVP-Tests fekfivps

- **userlog/\$LOGNAME/fekfivps.log**

Ausgabe des Befehls fekfivps -file (IVP-Test für SCLMDT). Dabei ist userlog der kombinierte Wert der Anweisungen user.log und DSTORE_LOG_DIRECTORY in rsed.envvars und \$LOGNAME ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung user.log in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung DSTORE_LOG_DIRECTORY in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird .eclipse/RSE/ an den Wert von user.log angehängt.

Speicherauszugsdateien

Wenn ein Produkt anormal beendet wird, wird ein Speicherauszug zur Unterstützung der Fehlerbestimmung erstellt. Verfügbarkeit und Position dieser Speicherauszüge hängen in hohem Maße von standortspezifischen Einstellungen ab. Es ist möglich, dass sie gar nicht oder an anderen Positionen als unten angegeben erstellt werden.

MVS-Speicherauszüge

Wenn das Programm unter MVS ausgeführt wird, überprüfen Sie die Systemspeicherauszugsdateien und Ihre JCL (je nach Produkt) auf die folgenden DD-Anweisungen:

- SYSABEND
- SYSMDUMP
- SYSUDUMP
- CEEDUMP
- SYSPRINT
- SYSOUT

Weitere Informationen zu diesen DD-Anweisungen sind in den Veröffentlichungen *MVS JCL Reference* (IBM Form SA22-7597) und *Language Environment Debugging Guide* (IBM Form GA22-7560) enthalten.

Java-Speicherauszüge

Unter z/OS UNIX werden die meisten Speicherauszüge von Developer for System z durch die Java Virtual Machine (JVM) gesteuert.

Die JVM erstellt während ihrer Initialisierung eine Gruppe von Speicherauszugsagenten (SYSTDUMP und JAVADUMP). Sie können diese Speicherauszugsagenten mit der Umgebungsvariablen `JAVA_DUMP_OPTS` sowie in der Befehlszeile mit `-Xdump` außer Kraft setzen. JVM-Befehlszeilenoptionen sind in der Anweisung `_RSE_JAVA_OPTS` der Datei `rsed.envvars` definiert. Ändern Sie die Speicherauszeugs-einstellungen nur auf Anweisung des IBM Support Center.

Anmerkung: Mit der Option `-Xdump:what` in der Befehlszeile können Sie feststellen, welche Speicherauszugsagenten nach Beendigung des Systemstarts vorhanden sind.

Folgende Arten von Speicherauszügen können erzeugt werden:

SYSTDUMP

Java-Transaktionsspeicherauszug. Dies ist ein nicht formatierter, von z/OS generierter Speicherauszug.

Der Speicherauszug wird in eine sequenzielle MVS-Datei geschrieben, deren Name standardmäßig die Form `%uid.JVM.TDUMP.%job.D%ym%d.T%H%M%S` hat oder von der Umgebungsvariablen `JAVA_DUMP_TDUMP_PATTERN` bestimmt wird. Falls Sie keine Transaktionsspeicherauszüge erstellen möchten, fügen Sie die Umgebungsvariable `IBM_JAVA_ZOS_TDUMP=NO` zur Datei `rsed.envvars` hinzu.

Anmerkung: Mit `JAVA_DUMP_TDUMP_PATTERN` können Variablen verwendet werden, die zum Zeitpunkt der Erstellung des Transaktionsspeicherauszugs in einen tatsächlichen Wert umgesetzt werden.

Tabelle 20. Variablen für JAVA_DUMP_TDUMP_PATTERN

Variable	Verwendung
%uid	Benutzer-ID
%job	Jobname
%y	Jahr (2-stellig)
%m	Monat (2-stellig)
%d	Tag (2-stellig)
%H	Stunde (2-stellig)
%M	Minute (2-stellig)
%S	Sekunde (2-stellig)

CEEDUMP

LE-Speicherauszug (Language Environment). Dies ist ein Systemspeicherauszug in einer formatierten Zusammenfassung, die die Stack-Traces für jeden Thread im JVM-Prozess zusammen mit Registerinformationen und einem Kurzspeicherauszug für jedes Register anzeigt.

Der Speicherauszug wird in eine z/OS UNIX-Datei mit dem Namen CEEDUMP.yyyymmdd.hhmmss.pid geschrieben. Dabei stehen yyyymmdd für das aktuelle Datum, hhmmss für die aktuelle Uhrzeit und pid für die ID des aktuellen Prozesses. Die möglichen Positionen dieser Datei sind im Abschnitt „Positionen für z/OS UNIX-Speicherauszüge“ auf Seite 146 beschrieben.

HEAPDUMP

Dies ist ein formatierter Speicherauszug (Liste) der Objekte im Java-Heapspeicher.

Der Speicherauszug wird in eine z/OS UNIX-Datei mit dem Namen HEAPDUMP.yyyymmdd.hhmmss.pid.TXT geschrieben. Dabei stehen yyyymmdd für das aktuelle Datum, hhmmss für die aktuelle Uhrzeit und pid für die ID des aktuellen Prozesses. Die möglichen Positionen dieser Datei sind im Abschnitt „Positionen für z/OS UNIX-Speicherauszüge“ auf Seite 146 beschrieben.

JAVADUMP

Dies ist eine formatierte Analyse der JVM. Sie enthält Diagnoseinformationen zur JVM und zur Java-Anwendung, z. B. Angaben zur Anwendungsumgebung, zu Threads, zum nativen Stack, zu Sperren und zum Hauptspeicher.

Der Speicherauszug wird in eine z/OS UNIX-Datei mit dem Namen JAVADUMP.yyyymmdd.hhmmss.pid.TXT geschrieben. Dabei stehen yyyymmdd für das aktuelle Datum, hhmmss für die aktuelle Uhrzeit und pid für die ID des aktuellen Prozesses. Die möglichen Positionen dieser Datei sind im Abschnitt „Positionen für z/OS UNIX-Speicherauszüge“ auf Seite 146 beschrieben.

Weitere Informationen zu JVM-Speicherauszügen enthält der *Java Diagnostic Guide* (IBM Form SC34-6358). LE-spezifische Informationen finden Sie im *Language Environment Debugging Guide* (IBM Form GA22-7560).

Positionen für z/OS UNIX-Speicherauszüge

Die JVM überprüft alle nachfolgend angegebenen Positionen auf ihr Vorhandensein und auf die Schreibberechtigungen. An der ersten verfügbaren Position werden die CEEDUMP-, HEAPDUMP- und JAVADUMP-Dateien gespeichert. Denken Sie daran, dass genug freier Plattenspeicherplatz vorhanden sein muss, damit die Speicherauszugsdatei ordnungsgemäß geschrieben werden kann.

1. In der Umgebungsvariablen `_CEE_DMPTARG` angegebenes Verzeichnis, sofern ein Wert gefunden wird. Diese Variable ist in `rsed.envvars` auf `/tmp` gesetzt. Sie kann in `/dev/null` geändert werden, wenn keine Speicherauszugsdateien erstellt werden sollen.
2. Das aktuelle Arbeitsverzeichnis, sofern es sich nicht um das Stammverzeichnis (`/`) handelt und in das Verzeichnis geschrieben werden kann
3. In der Umgebungsvariablen `TMPDIR` angegebenes Verzeichnis. (Wenn diese Umgebungsvariable gefunden wird, gibt sie die Position eines temporären Verzeichnisses an, sofern nicht `/tmp` verwendet wird.)
4. Das Verzeichnis `/tmp`
5. Falls der Speicherauszug in keinem der oben genannten Verzeichnisse gespeichert werden kann, wird er an `stderr` gesendet.

Traceerstellung

Traceerstellung für JES Job Monitor

Die Traceerstellung für JES Job Monitor wird, wie in Kapitel 8, „Bedienerbefehle“, auf Seite 125 beschrieben, vom Systembediener gesteuert.

- Wenn Sie die gestartete Task JMON mit dem Parameter `PRM=-TV` starten, wird der ausführliche Trace-Modus aktiviert.
- Mit den Befehlen **modify -TV** und **modify -TN** können Sie die Traceerstellung aktivieren bzw. inaktivieren.

Traceerstellung für RSE

Die RSE-bezogenen Komponenten erstellen diverse Protokolldateien. Die meisten Dateien werden in `userlog/$LOGNAME` gespeichert. Dabei ist `userlog` der kombinierte Wert der Anweisungen `user.log` und `DSTORE_LOG_DIRECTORY` in `rsed.envvars` und `$LOGNAME` ist die Anmeldebenutzer-ID (in Großbuchstaben). Wenn die Anweisung `user.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad des Benutzers verwendet. Der Ausgangspfad ist im OMVS-Sicherheitssegment der Benutzer-ID definiert. Wenn die Anweisung `DSTORE_LOG_DIRECTORY` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird `.eclipse/RSE/` an den Wert von `user.log` angehängt.

Das in `ffs*.log`, `lock.log` und `rsecomm.log` geschriebene Datenvolumen wird durch den Bedienerbefehl **modify rsecommlog** oder von der Einstellung `debug_level` in `rsecomm.properties` gesteuert. Ausführliche Informationen hierzu finden Sie in Kapitel 8, „Bedienerbefehle“, auf Seite 125 und im Abschnitt „RSE-Trace (optional)“ auf Seite 96.

Die Erstellung der Protokolldateien `.dstore*` wird von den Java-Startoptionen `-DDSTORE *` gesteuert. Lesen Sie hierzu die Informationen unter „Zusätzliche Java-Startparameter mit `_RSE_JVAOPTS` definieren“ auf Seite 42.

Anmerkung:

- Das Verzeichnis `.eclipse` und die Protokolldateien `.dstore*` beginnen mit einem Punkt (.) und sind dadurch verdeckt. Mit dem z/OS UNIX-Befehl `ls -lA` können Sie verdeckte Dateien und Verzeichnisse auflisten. Wenn Sie mit der Clientkomponente von Developer for System z arbeiten, wählen Sie **Fenster > Benutzer-vorgaben... > Ferne Systeme > Dateien** aus und aktivieren Sie die Option "Verdeckte Dateien anzeigen".
- Die Protokolldateien `.dstore*` werden im ASCII-Format erstellt. Verwenden Sie den z/OS UNIX-Befehl `iconv -f ISO8859-1 -t IBM-1047 .dstore*`, wenn Sie sie in EBCDIC (Codepage IBM-1047) anzeigen möchten.

Die Protokolldateien des RSE-Dämons und des RSE-Thread-Pools befinden sich in `daemon-home`. Dabei steht `daemon-home` für den Wert der Anweisung `daemon.log` in `rsed.envvars`. Wenn die Anweisung `daemon.log` in Kommentarzeichen gesetzt wurde oder nicht vorhanden ist, wird der Ausgangspfad der Benutzer-ID verwendet, die der gestarteten Task RSED zugeordnet ist. Das Ausgangsverzeichnis ist im Segment für die OMVS-Sicherheit der Benutzer-ID definiert.

Das in `rsedaemon*.log` und `rserver.log` geschriebene Datenvolumen wird durch die Bedienerbefehle **modify rsedaemonlog** und **modify rserverlog** oder von der Einstellung `debug_level` in `rsecomm.properties` gesteuert. Ausführliche Informationen hierzu finden Sie in Kapitel 8, „Bedienerbefehle“, auf Seite 125 und im Abschnitt „RSE-Trace (optional)“ auf Seite 96.

Die Dateien `serverlogs.count`, `stderr*.log` und `stdout*.log` werden nur erstellt, wenn die Anweisung `enable.standard.log` in `rsed.envvars` aktiv ist oder wenn die Funktion mit dem Bedienerbefehl **modify rsestandardlog on** dynamisch aktiviert wurde.

Traceerstellung für den Sperrdämon

Das spezifische Sperrdämonprotokoll befindet sich in der STDOUT DD der gestarteten Task LOCKD. Das in dieses Protokoll geschriebene Datenvolumen wird von dem Startparameter LOG gesteuert. Ausführliche Informationen hierzu finden Sie in Kapitel 8, „Bedienerbefehle“, auf Seite 125 und im Abschnitt „RSE-Trace (optional)“ auf Seite 96.

Traceerstellung für CARMA

Den Umfang der von CARMA generierten Trace-Informationen kann der Benutzer steuern, indem er auf dem Client auf der Eigenschaftenregisterkarte der CARMA-Verbindung die 'Tracestufe' definiert. Folgende Optionen sind für die Tracestufe verfügbar:

- Protokollierung inaktivieren
- Fehler
- Warnung
- Information
- Debug

Standardwert:

Fehler

Weitere Informationen zur Position der Protokolldateien enthält der Abschnitt „Protokolldateien“ auf Seite 138.

Trace für Fehlerrückmeldungen

Mit der folgenden Prozedur können Informationen zusammengestellt werden, die notwendig sind, um Probleme bei Fehlerrückmeldungen für ferne Buildprozeduren zu diagnostizieren. Dieser Trace bringt Leistungseinbußen mit sich und sollte nur auf Anweisung des IBM Support Center durchgeführt werden. Alle in diesem Abschnitt enthaltenen Verweise auf HLQ beziehen sich auf das während der Installation von Developer for System z verwendete übergeordnete Qualifikationsmerkmal. Die Standardeinstellung für die Installation ist FEK, die jedoch nicht für Ihren Standort zutreffen muss.

1. Erstellen Sie eine Sicherungskopie Ihrer aktiven ELAXFC0C-Kompilierungsprozedur. Standardmäßig ist diese Prozedur in der Datei HLQ.SFEKSAMP enthalten. Möglicherweise wurde sie jedoch an eine andere Position kopiert, z. B. nach SYS1.PROCLIB. Lesen Sie hierzu den Abschnitt „ELAXF*-Prozeduren für ferne Builderstellung“ auf Seite 26.
2. Ändern Sie die aktive ELAXFC0C-Prozedur so, dass sie in der Kompilierungsoption EXIT(ADEXIT(ELAXMGUX)) die Zeichenfolge 'MAXTRACE' enthält.

```
//COBOL EXEC PGM=IGYCRCTL,REGION=2048K,
//*      PARM=('EXIT(ADEXIT(ELAXMGUX))'),
//      PARM=('EXIT(ADEXIT('MAXTRACE',ELAXMGUX))'),
//      'ADATA',
//      'LIB',
//      'TEST(NONE,SYM,SEP)',
//      'LIST',
//      'FLAG(I,I)'&CICS &DB2 &COMP)
```

Anmerkung: Sie müssen MAXTRACE in doppelte Hochkommata setzen. Die Option sieht jetzt wie folgt aus:
EXIT(ADEXIT('MAXTRACE',ELAXMGUX))

3. Führen Sie eine ferne Syntaxprüfung für das COBOL-Programm durch, für das ein detaillierter Trace erstellt werden soll.
4. Der Abschnitt SYSOUT der JES-Ausgabe beginnt mit einer Auflistung der Dateinamen für SIDEFILE1, SIDEFILE2, SIDEFILE3 und SIDEFILE4.

```
ABOUT TOO OPEN SIDEFILE1 - NAME = 'uid.DT021207.TT110823.M0000045.C0000000'
SUCCESSFUL OPEN SIDEFILE1 - NAME = 'uid.DT021207.TT110823.M0000045.C0000000'
ABOUT TOO OPEN SIDEFILE2 - NAME = 'uid.DT021207.TT110823.M0000111.C0000001'
SUCCESSFUL OPEN SIDEFILE2 - NAME = 'uid.DT021207.TT110823.M0000111.C0000001'
ABOUT TOO OPEN SIDEFILE3 - NAME = 'uid.DT021207.TT110823.M0000174.C0000002'
SUCCESSFUL OPEN SIDEFILE3 - NAME = 'uid.DT021207.TT110823.M0000174.C0000002'
ABOUT TOO OPEN SIDEFILE4 - NAME = 'uid.DT021207.TT110823.M0000236.C0000003'
SUCCESSFUL OPEN SIDEFILE4 - NAME = 'uid.DT021207.TT110823.M0000236.C0000003'
```

Anmerkung: SIDEFILE1 und SIDEFILE2 können, abhängig von Ihren Einstellungen, auf eine DD-Anweisung zeigen (SUCCESSFUL OPEN SIDEFILE1 - NAME = DD:WSEDSF1). Den tatsächlichen Namen der Datei finden Sie im Abschnitt JESJCL der Ausgabe (der sich vor dem Abschnitt SYSOUT befindet).

```
22 //COBOL.WSEDSF1 DD DISP=MOD,
// DSN=uid.ERRCOB.member.SF.Z682746.XML
23 //COBOL.WSEDSF2 DD DISP=MOD,
// DSN=uid.ERRCOB.member.SF.Z682747.XML
```

5. Kopieren Sie diese vier Dateien auf Ihren PC, indem Sie beispielsweise in Developer for System z ein lokales COBOL-Projekt erstellen und die Dateien SIDEFILE1 bis SIDEFILE4 diesem Projekt hinzufügen.
6. Kopieren Sie das vollständige JES-Jobprotokoll auf Ihren PC, indem Sie beispielsweise die Jobausgabe in Developer for System z öffnen und **Datei > Speichern als...** auswählen, um das Protokoll im lokalen Projekt zu speichern.

7. Stellen Sie den ursprünglichen Zustand der Prozedur ELAXFC0C wieder hier, indem Sie die Änderung rückgängig machen (die Zeichenfolge "MAXTRACE" aus den Kompilierungsoptionen entfernen) oder die Sicherungskopie zurückschreiben.
8. Senden Sie die gesammelten Dateien (SIDEFILE1 bis SIDEFILE4 sowie das Jobprotokoll) an das IBM Support Center.

z/OS UNIX-Berechtigungsbits

Developer for System z erfordert, dass für das z/OS UNIX-Dateisystem und einige z/OS UNIX-Dateien bestimmte Berechtigungsbits gesetzt sind.

Dateisystemattribut SETUID

Remote Systems Explorer (RSE) ist die Komponente von Developer for System z, die Kernservices wie die Verbindung vom Client zum Host bereitstellt. Diese Komponente muss in der Lage sein, Tasks wie die Erstellung der Sicherheitsumgebung für den Benutzer auszuführen.

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

Mit dem TSO-Befehl **ISHELL** können Sie den aktuellen Status des Bits SETUID anzeigen. Wählen Sie in der ISHELL-Anzeige **File_systems > 1. Mount table...** aus, um die angehängten Dateisysteme aufzulisten. Mit dem Zeilenbefehl **a** können Sie die Attribute für das ausgewählte Dateisystem anzeigen. Das Feld "Ignore SETUID" sollte auf 0 gesetzt sein.

Programmsteuerung autorisieren

Remote Systems Explorer (RSE) ist die Komponente von Developer for System z, die Kernservices wie die Verbindung vom Client zum Host bereitstellt. Für die Ausführung von Tasks, z. B. die Umschaltung auf die Benutzer-ID des Clients, muss die Komponente programmgesteuert ausgeführt werden.

Während der SMP/E-Installation wird das z/OS UNIX-Programmsteuerungsbit dort gesetzt, wo es erforderlich ist, außer für die Java-Schnittstelle zu Ihrem Sicherheitsprodukt. Lesen Sie hierzu die Informationen unter Kapitel 10, „Sicherheitsaspekte“, auf Seite 159. Dieses Berechtigungsbit könnte verloren gehen, wenn Sie es nicht in einer manuell erstellten Kopie der Verzeichnisse von Developer for System z gesichert haben.

Folgende Dateien von Developer for System z müssen programmgesteuerte Dateien sein:

- /usr/lpp/rdz/bin/
 - fekfdivp
 - fekfomvs
 - fekfrivp
- /usr/lpp/rdz/lib/
 - fekfdir.dll
 - libfekdcore.so
 - libfekfmain.so

- /usr/lpp/rdz/lib/icuc/
 - libicudata.dll
 - libicudata40.1.dll
 - libicudata40.dll
 - libicudata64.40.1.dll
 - libicudata64.40.dll
 - libicudata64.dll
 - libicuuc.dll
 - libicuuc40.1.dll
 - libicuuc40.dll
 - libicuuc64.40.1.dll
 - libicuuc64.40.dll
 - libicuuc64.dll

Anmerkung: Die Dateien libicu*64.* sind nur vorhanden, wenn Sie die vorläufige Programmkorrektur von Developer for System z angewendet haben, die als Antwort auf APAR AM07305 für die Aktivierung der 64-Bit-Unterstützung entwickelt wurde.

Verwenden Sie den z/OS UNIX-Befehl **ls -E, ph>**, um die erweiterten Attribute aufzulisten, in denen das Programmsteuerungsbit mit dem Buchstaben **p** markiert ist. Sehen Sie sich dazu das folgende Beispiel an (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ cd /usr/lpp/rdz
$ ls -E lib/fekf*
-rwxr-xr-x -ps- 2 user      group      94208 Jul  8 12:31 lib/fekfdir.dll
```

Mit dem z/OS UNIX-Befehl **extattr +p** können Sie das Programmsteuerungsbit manuell setzen. Vergleichen Sie hierzu das folgende Beispiel (\$ und # sind die z/OS UNIX-Eingabeaufforderung):

```
$ cd /usr/lpp/rdz
$ su
# extattr +p lib/fekf*
# exit
$ ls -E lib/fekf*
-rwxr-xr-x -ps- 2 user      group      94208 Jul  8 12:31 lib/fekfdir.dll
```

Anmerkung: Für die Verwendung des Befehls **extattr +p** benötigen Sie mindestens Lesezugriff auf das Profil **BPX.FILEATTR.PROGCTL** in der Klasse **FACILITY** Ihrer Sicherheitssoftware. Wenn dieses Profil nicht definiert ist, müssen Sie ein Superuser (UID 0) sein. Weitere Informationen hierzu enthält die Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

APF-Autorisierung

Remote Systems Explorer (RSE) ist die Komponente von Developer for System z, die Kernservices bereitstellt, wie den Verbindungsaufbau des Clients mit dem Host. Für die Ausführung von Tasks, wie das Anzeigen detaillierter Informationen zur Prozessressourcennutzung, muss die Komponente APF-autorisiert ausgeführt werden.

Das z/OS UNIX-APF-Bit wird während der SMP/E-Installation gesetzt, wo es erforderlich ist. Dieses Berechtigungsbit könnte verloren gehen, wenn Sie es nicht in einer manuell erstellten Kopie der Verzeichnisse von Developer for System z gesichert haben.

Die folgenden Dateien von Developer for System z müssen APF-autorisiert sein:

- /usr/lpp/rdz/bin/
 - fekfomvs
 - fekfrivp

Verwenden Sie den z/OS UNIX-Befehl **ls -E**, um die erweiterten Attribute aufzulisten, in denen das APF-Bit mit dem Buchstaben **a** markiert ist. Sehen Sie sich hierzu das folgende Beispiel an (" \$" ist die z/OS UNIX-Eingabeaufforderung):

```
$ cd /usr/lpp/rdz
$ ls -E bin/fekfrivp
-rwxr-xr-x  aps-  2 user      group      114688 Sep 17 06:41 bin/fekfrivp
```

Verwenden Sie den z/OS UNIX-Befehl **extattr +a**, um das APF-Bit manuell zu setzen. Sehen Sie sich hierzu das folgende Beispiel an (" \$" und "# " sind die z/OS UNIX-Eingabeaufforderungen):

```
$ cd /usr/lpp/rdz
$ su
# extattr +a bin/fekfrivp
# exit
$ ls -E bin/fekfrivp
-rwxr-xr-x  aps-  2 user      group      114688 Sep 17 06:41 bin/fekfrivp
```

Anmerkung: Für die Verwendung des Befehls **extattr +a** benötigen Sie mindestens Lesezugriff auf das Profil BPX.FILEATTR.APF in der Klasse FACILITY Ihrer Sicherheitssoftware. Wenn dieses Profil nicht definiert ist, müssen Sie ein Superuser (UID 0) sein. Weitere Informationen enthält die Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Sticky Bit

Einige der optionalen Services von Developer for System z erfordern, dass MVS-Lademodule für z/OS UNIX zur Verfügung stehen. Deshalb wird in z/OS UNIX ein Stub (eine Pseudodatei) mit aktiviertem Sticky Bit erstellt. Wenn der Stub ausgeführt wird, sucht z/OS UNIX nach einem MVS-Lademodul mit demselben Namen und führt dieses anstelle des Stubs aus.

Das Sticky Bit für z/OS UNIX wird während der SMP/E-Installation gesetzt, wo es erforderlich ist. Derartige Berechtigungsbits können verloren gehen, wenn Sie sie nicht in einer manuell erstellten Kopie der Verzeichnisse von Developer for System z gesichert haben.

Das Sticky Bit muss für die folgenden Dateien von Developer for System z aktiviert sein:

- /usr/lpp/rdz/bin/
 - BWBTSOW
 - CRASTART

Verwenden Sie den z/OS UNIX-Befehl **ls -l**, um die Berechtigungen aufzulisten, in denen das Sticky Bit mit dem Buchstaben **t** markiert ist. Sehen Sie sich dazu das folgende Beispiel an (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ cd /usr/lpp/rdz
$ ls -l bin/CRA*
-rwxr-xr-t  2 user      group          71 Jul  8 12:31 bin/CRASTART
```

Mit dem z/OS UNIX-Befehl **chmod +t** können Sie das Sticky Bit manuell setzen. Vergleichen Sie hierzu das folgende Beispiel (\$ und # sind die z/OS UNIX-Eingabeaufforderung):

```
$ cd /usr/lpp/rdz
$ su
# chmod +t bin/CRA*
# exit
$ ls -l bin/CRA*
-rwxr-xr-t  2 user      group          71 Jul  8 12:31 bin/CRASTART
```

Anmerkung: Für die Verwendung des Befehls **chmod** benötigen Sie mindestens die Zugriffsberechtigung READ für das Profil SUPERUSER.FILESYS.CHANGEPERMS in der Klasse UNIXPRIV Ihrer Sicherheitssoftware. Wenn dieses Profil nicht definiert ist, müssen Sie ein Superuser (UID 0) sein. Weitere Informationen hierzu enthält die Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Reservierte TCP/IP-Ports

Mit dem Befehl **netstat** (TSO oder z/OS UNIX) können Sie eine Übersicht der zurzeit verwendeten Ports aufrufen. Die Ausgabe dieses Befehls sieht in etwa wie das folgende Beispiel aus. Die letzte Zahl in der Spalte "Local Socket" (nach "..") gibt die verwendeten Ports an. Da diese Ports bereits genutzt werden, können sie nicht für die Konfiguration von Developer for System z verwendet werden.

IPV4

```
MVS TCP/IP NETSTAT CS VxRy      TCPIP Name: TCPIP          16:36:42
```

User Id	Conn	Local Socket	Foreign Socket	State
-----	----	-----	-----	-----
BPX0INIT	00000014	0.0.0.0..10007	0.0.0.0..0	Listen
INETD4	0000004D	0.0.0.0..512	0.0.0.0..0	Listen
RSED	0000004B	0.0.0.0..4035	0.0.0.0..0	Listen
JMON	00000038	0.0.0.0..6715	0.0.0.0..0	Listen

IPV6

```
MVS TCP/IP NETSTAT CS VxRy      TCPIP Name: TCPIP          12:46:25
```

User Id	Conn	State
-----	----	-----
BPX0INIT	00000018	Listen
	Local Socket:	0.0.0.0..10007
	Foreign Socket:	0.0.0.0..0
INETD4	00000046	Listen
	Local Socket:	0.0.0.0..512
	Foreign Socket:	0.0.0.0..0
RSED	0000004B	Listen
	Local Socket:	0.0.0.0..4035
	Foreign Socket:	0.0.0.0..0
JMON	00000037	Listen
	Local Socket:	0.0.0.0..6715
	Foreign Socket:	0.0.0.0..0

Eine andere bestehende Einschränkung sind reservierte TCP/IP-Ports. Es gibt die beiden folgenden allgemeinen Bereiche, in denen TCP/IP-Ports reserviert werden:

- **PROFILE.TCPIP**

Auf diese Datei verweist die DD-Anweisung PROFILE der gestarteten TCP/IP-Task, die oft den Namen SYS1.TCPPARMS(TCPPROF) hat.

- PORT: Reserviert einen Port für angegebene Jobnamen
- PORTRANGE: Reserviert einen Portbereich für angegebene Jobnamen

Weitere Informationen zu diesen Anweisungen finden Sie im *Communications Server: IP Configuration Guide* (IBM Form SC31-8775).

• **SYS1.PARMLIB(BPXPRMxx)**

- INADDRANYPORT: Gibt die Nummer des ersten Ports für den Portbereich an, den das System für die Bindung an PORT 0 mit INADDR_ANY reserviert. Dieser Wert wird nur für CINET (mehrere aktive TCP/IP-Stacks auf einem einzelnen Host) benötigt.
- INADDRANYCOUNT: Gibt die Anzahl der vom System reservierten Ports an, einschließlich des mit dem Parameter INADDRANYPORT angegebenen Ports. Dieser Wert wird nur für CINET (mehrere aktive TCP/IP-Stacks auf einem einzelnen Host) benötigt.

Weitere Informationen zu diesen Anweisungen können Sie den Veröffentlichungen *UNIX System Services Planning* (IBM Form GA22-7800) und *MVS Initialization and Tuning Reference* (IBM Form SA22-7592) entnehmen.

Diese reservierten Ports können mit dem Befehl **netstat portl** (TSO oder z/OS UNIX) aufgelistet werden. Die erstellte Ausgabe entspricht in etwa dem folgenden Beispiel:

MVS TCP/IP Port#	NETSTAT Prot	CS User	VxRy Flags	TCP/IP Range	Name: TCP/IP IP Address	17:08:32
00007	TCP	MISCSERV	DA			
00009	TCP	MISCSERV	DA			
00019	TCP	MISCSERV	DA			
00020	TCP	OMVS	D			
00021	TCP	FTPD1	DA			
00025	TCP	SMTP	DA			
00053	TCP	NAMESRV	DA			
00080	TCP	OMVS	DA			
03500	TCP	OMVS	DAR	03500-03519		
03501	TCP	OMVS	DAR	03500-03519		

Weitere Informationen zum Befehl **NETSTAT** enthält die Veröffentlichung *Communications Server: IP System Administrator's Commands* (IBM Form SC31-8781).

Anmerkung: Der Befehl **NETSTAT** zeigt nur die in PROFILE.TCPIP definierten Informationen an, die sich mit den Definitionen in BPXPRMxx überschneiden müssten. Überprüfen Sie im Zweifelsfall, welche Ports im PARMLIB-Member BPXPRMxx reserviert sind.

Größe des Adressraums

Der RSE-Dämon ist ein z/OS UNIX-Java-Prozess und erfordert für die Ausführung seiner Funktionen eine große Regionsgröße. Deshalb ist es wichtig, dass für OMVS-Adressräume große Speichergrenzen festgelegt werden.

Anforderungen an die Start-JCL

Der RSE-Dämon wird über JCL mit BPXBATSL gestartet. Die Regionsgröße von BPXBATSL muss gleich null sein.

In SYS1.PARMLIB(BPXPRMxx) festgelegte Begrenzungen

Setzen Sie MAXASSIZE in SYS1.PARMLIB(BPXPRMxx) (zum Definieren der Standardregionsgröße bzw. -prozessgröße für den OMVS-Adressraum) auf mindestens 2G. Dies ist die zulässige Maximalgröße. Dieser Grenzwert gilt systemweit. Er ist daher für alle z/OS UNIX-Adressräume aktiv. Wenn Sie dies nicht wünschen, können Sie in Ihrer Sicherheitssoftware den Grenzwert auch nur für Developer for System z festlegen.

Dieser Wert kann mit folgenden Konsolbefehlen überprüft und dynamisch (bis zum nächsten IPL) gesetzt werden. Lesen Sie hierzu die Beschreibung in der Veröffentlichung *MVS System Commands* (IBM Form GC28-1781).

1. DISPLAY OMVS,0
2. SETOMVS MAXASSIZE=2G

Im Sicherheitsprofil gespeicherte Begrenzungen

Überprüfen Sie ASSIZEMAX im OMVS-Segment der Dämonbenutzer-ID und setzen Sie das Feld auf 2147483647 oder vorzugsweise auf NONE, damit der Wert SYS1.PARMLIB(BPXPRMxx) verwendet wird.

Dieser Wert kann in RACF mit den folgenden TSO-Befehlen überprüft und gesetzt werden. Lesen Sie hierzu die Beschreibung in der Veröffentlichung *Security Server RACF Command Language Reference* (IBM Form SA22-7687).

1. LISTUSER userid NORACF OMVS
2. ALTUSER userid OMVS(NOASSIZEMAX)

Von Systemexits erzwungene Begrenzungen

Stellen Sie sicher, dass Regionsgrößen des OMVS-Adressraums nicht von den Systemexits IEFUSI oder IEALIMIT gesteuert werden. Eine Möglichkeit, dies zu erreichen, ist die Verwendung des Codes SUBSYS(OMVS,NOEXITS) in SYS1.PARMLIB(SMFPRMxx).

SYS1.PARMLIB(SMFPRMxx)-Werte können mit folgenden Konsolbefehlen überprüft und aktiviert werden. Lesen Sie hierzu die Beschreibung in der Veröffentlichung *MVS System Commands* (IBM Form GC28-1781).

1. DISPLAY SMF,0
2. SET SMF=xx

Begrenzungen für die 64-Bit-Adressierung

Das Schlüsselwort MEMLIMIT in SYS1.PARMLIB(SMFPRMxx) legt fest, wie viel virtuellen Speicher eine 64-Bit-Task oberhalb der Grenze von 2GB zuweisen darf. Im Gegensatz zu dem Parameter REGION in JCL bedeutet MEMLIMIT=0M, dass der Prozess keinen virtuellen Speicher oberhalb der Grenze verwenden darf.

Wenn MEMLIMIT nicht in SMFPRMxx angegeben wird, ist der Standardwert 0M, das heißt, dass Tasks an die 2 GB (31 Bit) unterhalb der Grenze gebunden sind. Der in z/OS 1.10 auf 2G geänderte Standard ermöglicht 64-Bit-Tasks die Verwendung von bis zu 4 GB (2 GB unterhalb der Grenze und 2 GB durch MEMLIMIT).

SYS1.PARMLIB(SMFPRMxx)-Werte können mit folgenden Konsolbefehlen überprüft und aktiviert werden. Lesen Sie hierzu die Beschreibung in der Veröffentlichung *MVS System Commands* (IBM Form GC28-1781).

1. DISPLAY SMF,0

2. SET SMF=xx

MEMLIMIT kann auch als Parameter auf der EXEC-Karte in JCL angegeben werden. Wenn der Parameter MEMLIMIT nicht angegeben ist, ist der Standard der in SMF definierte Wert. Dies gilt nicht, wenn REGION=0M angegeben ist. In diesem Fall ist der Standardwert NOLIMIT.

APPC-Transaktion und TSO Commands Service

Falls Sie die APPC-Version von TSO Commands Service nicht nutzen können, können in zwei Bereichen Probleme auftauchen: beim Starten der APPC-Servertransaktion und beim Herstellen einer Verbindung zu RSE.

- Falls die Nachrichten zum Konfigurieren von APPC nicht angezeigt werden, überprüfen Sie, ob das Systemprotokoll RACF-Nachrichten (mit der Nachrichten-ID ICHxxxxx) oder andere Nachrichten zu dem abgesetzten Befehl enthält oder zu der Benutzer-ID, die den Befehl abgesetzt hat. Einige allgemeine Ursachen für Probleme:
 - Sie haben keine Leseberechtigung für die Datei FEK.SFEKPROC.
 - TCP/IP ist nicht aktiv, TCP/IP wurde ein falscher DNS-Name zugeordnet oder das System ist aufgrund von Netzproblemen, einer falschen IP-Adresse oder aus anderen Gründen nicht (mit Ping) erreichbar.
- Wenn die Nachrichten zum Konfigurieren von APPC angezeigt werden, jedoch keine Bestätigung für den erfolgreichen Abschluss der Konfiguration, konnte wahrscheinlich die APPC-Servertransaktion nicht gestartet werden. Überprüfen Sie das Transaktionsfehlerprotokoll (USERID.FEKFRSRV.&TPDATE.&TPTIME.LOG). Einige wahrscheinliche Problemursachen sind:
 - Der TCP/IP-Stack verwendet nicht den Standardnamen TCPIP und die DD-Karte SYSTCPD wurde nicht gesetzt oder zeigt auf die falsche Datei.
 - Der Server konnte SYSPROC oder SYSTSPRT nicht zuordnen.
 - Die JCL zeigt auf die falsche SYSPROC. (SYSPROC muss FEK.SFEKPROC enthalten.)
 - Der Server konnte die von MESSAGE_DATA_SET referenzierte Nachrichtendatei (das Protokoll) nicht öffnen oder nicht auf die Datei zugreifen.
 - Es sind nicht genug APPC-Scheduler-Initiatoren verfügbar.
 - Der APPC- oder ASCH-Adressraum ist nicht aktiv.
 - Die verwendete Klasse (standardmäßig die Klasse "A") ist nicht für den APPC-Scheduler ASCH definiert.
 - Es ist kein Standard-OMVS-Segment für das System vorhanden und der Benutzer hat kein persönliches OMVS-Segment oder im Segment gibt es einen Definitionsfehler.
 - Die Standardgruppe des Standard-OMVS-Segments oder die Standardgruppe des Benutzers hat keine GID-Nummer.

Die im Abschnitt „APPC-Transaktion für TSO Commands Service (optional)“ auf Seite 104 angegebene REXX kann Ihnen bei der Lösung von APPC-Problemen helfen, denn sie ermöglicht Ihnen, APPC in ISPF-Anzeigen im Dialogbetrieb zu verwalten. Mit diesem Tool können Sie die APPC-Transaktion inaktivieren. Die Transaktion ist unverändert vorhanden, akzeptiert dann jedoch keine Verbindungen mehr.

Nachfolgend sind technische Hinweise aufgelistet, die derzeit auf der Supportwebsite (<http://www-306.ibm.com/software/awdtools/rdz/support/>) verfügbar sind. Zusätzliche Informationen entnehmen Sie bitte direkt der Supportwebsite.

- APPC-Prüfung scheitert mit Rückkehrcode 2016 - EHOSTNOTFOUND
- APPC-Prüfung scheitert mit Rückkehrcode 1004 - EIBMIUCVERR
- APPC-Prüfung scheitert mit Rückkehrcode 9 - TP-Name nicht erkannt
- APPC-Prüfung scheitert mit Rückkehrcode 10 - TP kann nicht erneut gestartet werden
- APPC-Prüfung scheitert mit Rückkehrcode 19 - Parameterfehler
- APPC-Prüfung scheitert mit Rückkehrcode 20 - Produktspezifischer Fehler
- APPC-Prüfung scheitert mit Rückkehrcode 26 - Ressourcenfehler
- CEE3501S: The module IOSTREAM was not found
- Der Server konnte nicht gestartet werden: EDC5129I No such file or directory
- exec/tcp: bind: EDC5111I Permission denied, rsn=744C7246
- Der Server antwortet nicht und es erscheint eine der folgenden Nachrichten:
 - IEA995I SYMPTOM DUMP OUTPUT 473 USER COMPLETION CODE=4093 REASON CODE=0000001C (in SDSF LOG)
 - CEE3512S An HFS load of module libicudata32.0.dll failed. The system return code was 0000000157; the reason code was 0BDF019B. (in CEEDUMP)
 - Get Space failed (in client .log)
- Befehl C_CONNECT nicht verfügbar
- Fehlernachricht "FFS server initialization failed" beim Herstellen der Verbindung mit dem Host
- Nachricht "EDC5139I Operation not permitted" beim Herstellen der Verbindung mit dem Host
- Nachricht "RSEG1056U FFS server initialization failed" beim Öffnen einer MVS-Datei

Anmerkung: Diese Liste ist nicht verbindlich. Überprüfen Sie, ob es auf der Supportwebsite zusätzliche technische Hinweise gibt.

Weitere Informationen

Systemgrenzwerte

SYS1.PARMLIB(BPXPRMxx) definiert viele z/OS UNIX-bezogene Begrenzungen, die erreicht werden können, wenn mehrere Clientkomponenten von Developer for System z aktiv sind. Die meisten BPXPRMxx-Werte können mit den Konsolbefehlen **SETOMVS** und **SET OMVS** dynamisch geändert werden.

Verwenden Sie den Konsolbefehl **SETOMVS LIMMSG=ALL**, damit unter z/OS UNIX Konsolnachrichten (BPXI040I) angezeigt werden, wenn Grenzwerte für BPXPRMxx annähernd überschritten werden.

Verbindung verweigert

Jede RSE-Verbindung startet mehrere Prozesse, die permanent aktiv sind. Neue Verbindungen können durch den in SYS1.PARMLIB(BPXPRMxx) gesetzten Grenzwert für die Anzahl der Prozesse zurückgewiesen werden. Dies gilt insbesondere, wenn Benutzer dieselbe UID gemeinsam benutzen (wie es z. B. bei Verwendung des Standard-OMVS-Segments der Fall ist).

- Der Grenzwert pro UID wird durch das Schlüsselwort MAXPROCUSER festgelegt und liegt standardmäßig bei 25.
- Der systemweite Grenzwert wird durch das Schlüsselwort MAXPROCSYS festgelegt und liegt standardmäßig bei 200.

Eine weitere Ursache für zurückgewiesene Verbindungen ist der Grenzwert für die Menge aktiver z/OS-Adressräume und z/OS UNIX-Benutzer.

- Die maximale Anzahl von Adressraum-IDs (ASID) wird in SYS1.PARMLIB(IEASYSxx) mit dem Schlüsselwort MAXUSER definiert. Der Standardwert liegt bei 255.
- Die maximale Anzahl von z/OS UNIX-Benutzer-IDs (UID) wird in SYS1.PARMLIB(BPXPRMxx) mit dem Schlüsselwort MAXUIDS definiert. Der Standardwert liegt bei 200.

Bekannte Probleme mit den Voraussetzungen

MVS-Dateien können nicht geöffnet werden

Wenn Sie APPC für TSO Commands Service verwenden, muss für das Lesen und Schreiben einer MVS-Datei eine Dateisystemdomäne mit physischen Sockets verwendet werden. Wenn das Dateisystem nicht ordnungsgemäß definiert ist oder nicht genug Sockets hat, verhindert der Sperrenmanager (FFS) Lese-/Schreibanforderungen. Die Dateien ffs*.log enthalten dann Nachrichten wie die folgenden:

- Error 127 getting socket pair - setting port to 0.
- Unable to create socket in the UNIX domain. Error is: "The address family is not supported"

Prüfen Sie, ob das Member SYS1.PARMLIB(BPXPRMxx) die folgenden Anweisungen enthält:

```
FILESYSTYPE TYPE(UDS) ENTRYPPOINT(BPXTUINT)
NETWORK DOMAINNAME(AF_UNIX)
        DOMAINNUMBER(1)
        MAXSOCKETS(2000)
        TYPE(UDS)
```

Wenn Sie APPC für TSO Commands Service verwenden, kommt als weitere Ursache für dieses Problem in Frage, dass der TCP/IP-Resolver die Hostadresse nicht ordnungsgemäß auflösen kann, weil eine Resolverkonfigurationsdatei fehlt oder unvollständig ist. Ein deutlicher Hinweis auf dieses Problem ist die folgende Nachricht in lock.log:

```
clientip(0.0.0.0) <> callerip(<Host-IP-Adresse>)
```

Führen Sie das TCP/IP-Installationsprüfprogramm fekfivpt wie in Kapitel 7, „Installationsprüfung“, auf Seite 109 beschrieben aus. Der Abschnitt der Ausgabe mit der Resolverkonfiguration sieht in etwa wie das folgende Beispiel aus:

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```
res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table = Default
UserId/JobName = USERID
Caller API = LE C Sockets
Caller Mode = EBCDIC
```

Vergewissern Sie sich, dass die Definitionen in der von „Local Tcp/Ip Dataset“ referenzierten Datei stimmen.

Wenn Sie für die IP-Resolver-Datei keinen Standardnamen verwenden, bleibt dieses Feld leer (bei Verwendung der z/OS UNIX-Suchreihenfolge). Fügen Sie in dem

Fall die folgende Anweisung zu `rsed.envvars` hinzu. `<Resolver-Datei>` repräsentiert hier den Namen Ihrer IP-Resolver-Datei.

```
RESOLVER_CONFIG='<Resolver-Datei>'
```

Host-Connect-Emulator

- Der Host-Connect-Emulator verwendet für Verbindungen zum Host TN3270-Telnet und nicht den RSE-Server.
- Wenn Sie mit sicherem Telnet (SSL) arbeiten und Zertifikate verwenden, die nicht von einer anerkannten Zertifizierungsstelle signiert sind, muss jeder Client das Zertifikat der Zertifizierungsstelle zu seiner Host-Connect-Emulator-Liste anerkannter Zertifizierungsstellen hinzufügen.
- Zum Inaktivieren der funktionalen SNA-Erweiterungen benötigen Sie möglicherweise die Option `NOSNAEXT` für die `TELNETPARMS` von TCP/IP. Wenn `NOSNAEXT` angegeben ist, führt der TN3270-Telnet-Server keine Verhandlungen zu einer Konfliktlösung oder zu SNA-Prüffunktionen.

Kapitel 10. Sicherheitsaspekte

Developer for System z ermöglicht Benutzern einer Workstation den Zugriff auf Mainframe-Computer, wenn diese selbst kein Mainframe-Computer ist. Wichtige Aspekte bei der Produktkonfiguration sind deshalb das Prüfen von Verbindungsanforderungen, das Bereitstellen von sicherer Kommunikation zwischen dem Host und der Workstation sowie das Autorisieren und Protokollieren der Aktivitäten.

Die von den Servern und Services von Developer for System z verwendeten Sicherheitsmechanismen sind nur wirksam, wenn das zugrunde liegende Dateisystem geschützt ist. Dies impliziert, dass die Programmbibliotheken und Konfigurationsdateien nur von vertrauenswürdigen Systemadministratoren aktualisiert werden können.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Authentifizierungsmethoden“
- „Verbindungssicherheit“ auf Seite 161
- „TCP/IP-Ports“ auf Seite 162
- „PassTickets verwenden“ auf Seite 164
- „Prüfprotokollierung“ auf Seite 165
- „JES-Sicherheit“ auf Seite 166
- „Mit SSL verschlüsselte Kommunikation“ auf Seite 170
- „Clientauthentifizierung unter Verwendung von X.509-Zertifikaten“ auf Seite 171
- „Eingangsport (POE) überprüfen“ auf Seite 175
- „CICSTS-Sicherheit“ auf Seite 176
- „SCLM-Sicherheit“ auf Seite 176
- „Konfigurationsdateien von Developer for System z“ auf Seite 177
- „Sicherheitsdefinitionen“ auf Seite 178

Anmerkung: Der Remote Systems Explorer (RSE), der Kernservices wie den Verbindungsaufbau vom Client zum Host bereitstellt, besteht aus 2 logischen Einheiten:

- Der RSE-Dämon verwaltet die Konfigurationsdateien und wird als gestartete Task oder als Benutzerjob mit langer Ausführungszeit gestartet.
- Der RSE-Server verarbeitet die einzelnen Clientanforderungen und wird vom RSE-Dämon in einem oder mehreren untergeordneten Prozessen als Thread gestartet.

Lesen Sie Kapitel 11, „Wissenswertes zu Developer for System z“, auf Seite 193, um mehr über grundlegende Designkonzepte von Developer for System z zu erfahren.

Authentifizierungsmethoden

Developer for System z unterstützt mehrere Möglichkeiten für die Authentifizierung einer Benutzer-ID, die von einem Client bei der Herstellung einer Verbindung bereitgestellt wurde.

- Benutzer-ID und Kennwort

- Benutzer-ID und Kennwort für einmaliges Anmelden
- X.509-Zertifikat

Beachten Sie, dass die vom Client bereitgestellten Authentifizierungsdaten nur einmalig, und zwar während der einleitenden Verbindungskonfiguration, verwendet werden. Sobald eine Benutzer-ID authentifiziert ist, werden die Benutzer-ID und die selbsterstellten PassTickets für alle Aktionen verwendet, die eine Authentifizierung erfordern.

Benutzer-ID und Kennwort

Der Client stellt bei der Herstellung einer Verbindung die Benutzer-ID und das entsprechende Kennwort bereit. Die Benutzer-ID und das Kennwort werden verwendet, um den Benutzer mit Ihrem Sicherheitsprodukt zu authentifizieren.

Benutzer-ID und Kennwort für einmaliges Anmelden

Basierend auf einem eindeutigen Token kann ein Kennwort für einmaliges Anmelden durch ein Produkt eines anderen Anbieters generiert werden. Kennwörter für einmaliges Anmelden dienen zur Verbesserung Ihrer Sicherheitseinstellung, da ein eindeutiges Kennwort nicht kopiert und nicht ohne die Zustimmung des Benutzers verwendet werden kann. Darüber hinaus ist es unbrauchbar, wenn es abgefangen wird, da es nur einmalig gültig ist.

Bei der Herstellung einer Verbindung stellt der Client eine Benutzer-ID und ein Kennwort für einmaliges Anmelden zur Verfügung. Dieses wird von einem Fremdanbieter bereitgestellt und dient zur Authentifizierung der Benutzer-ID mit dem Sicherheitsexit. Dieser Sicherheitsexit sollte die PassTickets ignorieren, die während der normalen Verarbeitung die Authentifizierungsanforderungen erfüllen. Die PassTickets müssen von Ihrer Sicherheitssoftware verarbeitet werden.

X.509-Zertifikat

Ein Fremdanbieter kann ein oder mehrere X.509-Zertifikate bereitstellen, die zur Authentifizierung eines Benutzers verwendet werden. Wenn das X.509-Zertifikat auf geschützten Einheiten gespeichert ist, kombiniert es eine sichere Konfiguration mit einem hohen Bedienungskomfort, da weder eine Benutzer-ID noch ein Kennwort erforderlich ist.

Beim Herstellen der Verbindung stellt der Client ein ausgewähltes Zertifikat und optional eine ausgewählte Erweiterung bereit, die zur Authentifizierung der Benutzer-ID mit Ihrem Sicherheitsprodukt dient.

Beachten Sie, dass diese Authentifizierungsmethode nur von der Verbindungsmethode des RSE-Dämons unterstützt wird und SSL aktiviert sein muss.

Authentifizierung durch JES Job Monitor

Die Clientauthentifizierung wird vom RSE-Dämon (oder REXEC/SSH) als Teil der Verbindungsanforderung des Clients vorgenommen. Sobald der Benutzer authentifiziert ist, werden selbsterstellte PassTickets für alle zukünftigen Authentifizierungsanforderungen verwendet, einschließlich des automatischen Anmeldens beim JES Job Monitor.

JES Job Monitor muss für die Überprüfung von PassTickets berechtigt sein, damit eine Überprüfung durch JES Job Monitor für die vom RSE übermittelten Benutzer-IDs und PassTickets möglich ist. Dies impliziert Folgendes:

- Das Lademodul FEJMON, das sich standardmäßig in der Ladebibliothek FEK.SFEKAUTH befindet, muss für APF autorisiert sein.
- RSE und JES Job Monitor müssen dieselbe Anwendungs-ID (APPLID) verwenden. Als Anwendungs-ID wird von beiden Servern standardmäßig FEKAPPL verwendet. Dies kann jedoch durch die Anweisung der Anwendungs-ID in `rsed.envvars` für RSE und in `FEJCNFG` für JES Job Monitor geändert werden.

Anmerkung: Ältere Clientversionen (bis Version 7.0) kommunizieren direkt mit dem JES Job Monitor. Für diese Verbindungen wird ausschließlich die Authentifizierung durch Benutzer-ID und Kennwort unterstützt.

Verbindungssicherheit

Verschiedene Ebenen der Kommunikationssicherheit werden vom RSE unterstützt. Dieser steuert die gesamte Kommunikation zwischen dem Client und den Developer for System z-Services:

- Die externe Kommunikation (Client-Host) kann auf bestimmte Ports beschränkt werden. Dieses Feature ist standardmäßig inaktiviert.
- Die externe Kommunikation (Client-Host) kann mit SSL verschlüsselt werden. Dieses Feature ist standardmäßig inaktiviert.
- Durch die Prüfung des Eingangsports kann erreicht werden, dass nur anerkannten TCP/IP-Adressen der Hostzugriff gewährt wird. Dieses Feature ist standardmäßig inaktiviert.

Externe Kommunikation auf angegebene Ports beschränken

Der Systemprogrammierer kann die Ports angeben, über die der RSE-Server mit dem Client kommunizieren kann. Standardmäßig kann jeder verfügbare Port verwendet werden. Dieser Portbereich steht nicht in Verbindung mit dem Port des RSE-Dämons.

Nachfolgend sehen Sie eine kurze Beschreibung des RSE-Verbindungsprozesses, die Ihnen helfen soll, die Portverwendung zu verstehen.

1. Der Client stellt über den Host-Port 4035 eine Verbindung mit dem RSE-Dämon her.
2. Der RSE-Dämon erstellt einen RSE-Server-Thread.
3. Der RSE-Server öffnet einen Host-Port, zu dem der Client eine Verbindung herstellen kann. Der Benutzer kann die Auswahl dieses Ports auf dem Client auf der Eigenschaftenregisterkarte für das Subsystem (nicht zu empfehlen) oder mit der Definition `_RSE_PORTRANGE` in `rsed.envvars` konfigurieren.
4. Der RSE-Dämon gibt die Portnummer an den Client zurück.
5. Der Client stellt eine Verbindung mit dem Host-Port her.

Anmerkung: Dieser Prozess ist mit der (optionalen) alternativen Verbindungsmethode unter Verwendung von REXEC/SSH vergleichbar. Lesen Sie hierzu die Informationen unter „REXEC (oder SSH) verwenden (optional)“ auf Seite 102.

Kommunikation mit SSL verschlüsseln

Alle externen Datenströme von Developer for System z, die den RSE-Server passieren, können mit SSL (Secure Sockets Layer) verschlüsselt werden. Die Verwendung von SSL wird von den Einstellungen in der Konfigurationsdatei `ssl.properties` gesteuert. Lesen Sie hierzu die Beschreibung im Abschnitt „Mit SSL verschlüsselte Kommunikation“ auf Seite 170.

Der Host-Connect-Emulator auf dem Client stellt eine Verbindung zu einem TN3270-Server auf dem Host her. Die Verwendung von SSL wird von TN3270 gesteuert. Diese Art der Verwendung ist im *Communications Server IP Configuration Guide* (IBM Form SC31-8775) dokumentiert.

Die Clientkomponente von Application Deployment Manager ruft mit dem CICS-TS-Web-Service bzw. der RESTful-Schnittstelle die Hostservices von Application Deployment Manager auf. Die Verwendung von SSL wird von CICS-TS gesteuert. Diese Art der Verwendung ist im *RACF Security Guide for CICS TS* dokumentiert.

Eingangsport überprüfen

Developer for System z unterstützt die Prüfung des Eingangsports, die eine Beschränkung des Hostzugriffs auf anerkannte TCP/IP-Adressen ermöglicht. Die Verwendung des Eingangsports wird von der Definition bestimmter Profile in Ihrer Sicherheitssoftware sowie von der Anweisung `enable.port.of.entry` in `rased.envvars` gesteuert. Eine diesbezügliche Beschreibung finden Sie unter „Eingangsport (POE) überprüfen“ auf Seite 175.

Die Aktivierung des Eingangsports wirkt sich auch auf andere TCP/IP-Anwendungen aus, die die Überprüfung des Eingangsports unterstützen, wie z. B. auf INETD.

TCP/IP-Ports

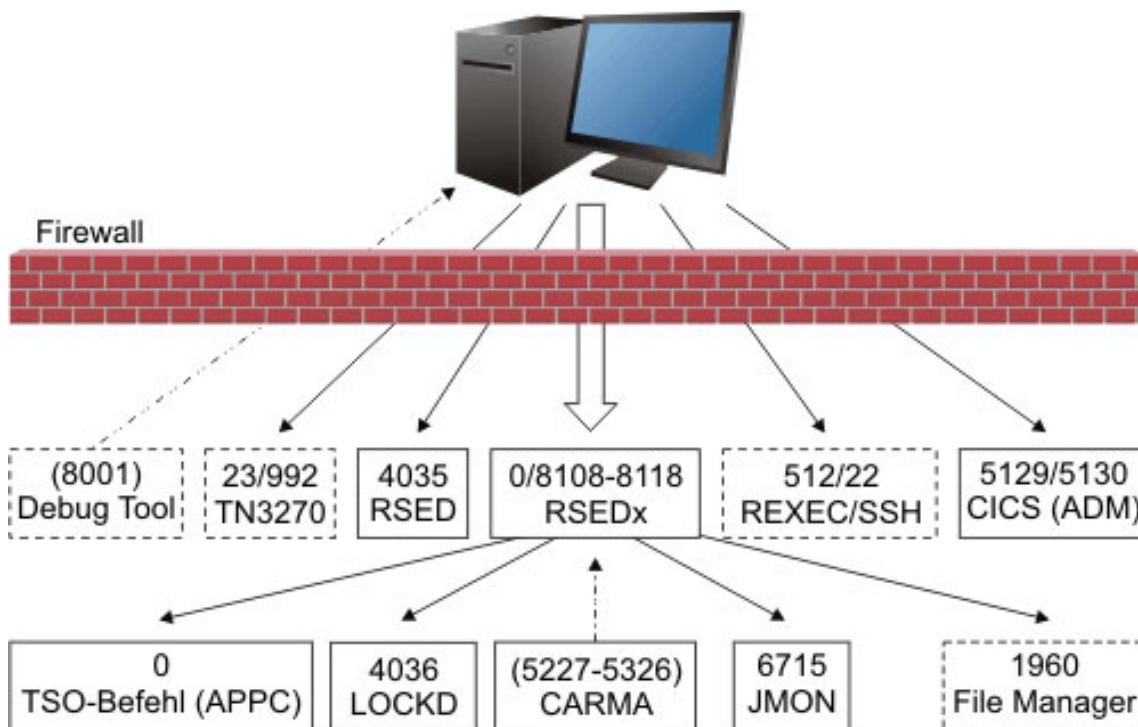


Abbildung 40. TCP/IP-Ports

Abb. 40 auf Seite 162 stellt die TCP/IP-Ports dar, die mit Developer for System z verwendet werden können. Die Pfeilspitzen deuten an, welcher Teilnehmer für die Bindung (Pfeilspitzenseite) verantwortlich ist und welcher die Verbindung herstellt.

Externe Kommunikation

Definieren Sie für die Firewall, die Ihren z/OS-Host schützt, die folgenden Ports für die Client-Host-Kommunikation (unter Verwendung des TCP-Protokolls):

- RSE-Dämon für die Einrichtung der Client-Host-Kommunikation (Standardport 4035). Die Kommunikation über diesen Port kann mit SSL verschlüsselt werden.
- RSE-Server für die Kommunikation zwischen Client und Host. Standardmäßig kann jeder verfügbare Port verwendet werden. Mit der Definition `_RSE_PORTRANGE` in `rsed.envvars` ist jedoch eine Einschränkung auf einen bestimmten Portbereich möglich. Die Kommunikation über diesen Port kann mit SSL verschlüsselt werden.
- Einen der INETD-Services für ferne (hostbasierte) Aktionen in z/OS UNIX-Unterprojekten (optional)
 - REXEC (z/OS UNIX-Version), Standardport 512
 - SSH (z/OS UNIX-Version), Standardport 22. Die Kommunikation über diesen Port ist mit SSL verschlüsselt.
- TN3270-Telnet-Service für den Host-Connect-Emulator (Standardport 23) (optional). Die Kommunikation kann mit SSL verschlüsselt werden. (Der Standardport ist dann 992.) Welcher Standardport dem Telnet-Service TN3270 zugeordnet wird, hängt davon ab, ob der Benutzer sich für oder gegen die Verwendung der Verschlüsselung entscheidet.
- Eine oder beide CICSTS-Anwendungsschnittstellen für Application Deployment Manager (optional):
 - RESTful-Schnittstelle, Standardport 5130
 - Web-Service-Schnittstelle, Standardport 5129. Die Kommunikation über diesen Port kann mit SSL verschlüsselt werden.

Anmerkung:

- Ältere Clientversionen (bis Version 7.0) kommunizieren direkt mit dem JES Job Monitor am Standardport 6715.
- Während einer fernen Debugsitzung für COBOL, PL/I oder Assembler wird IBM Debug Tool für z/OS aufgerufen. Dieses Produkt kommuniziert direkt mit dem Client. Die Kommunikation wird auf dem Host eingeleitet und stellt eine Verbindung mit dem Port 8001 des Clients her.

Interne Kommunikation

Mehrere Hostservices von Developer for System z werden in gesonderten Threads oder Adressräumen ausgeführt und verwenden TCP/IP-Sockets als Kommunikationsmechanismus. Alle diese Services nutzen RSE für die Kommunikation mit dem Client und beschränken ihren Datenstrom nur auf den Host. Für einige Services kann jeder verfügbare Port verwendet werden. Für andere kann der Systemprogrammierer wie folgt auswählen, welcher Port oder Portbereich verwendet werden soll:

- JES Job Monitor für JES-bezogene Services, Standardport 6715. Der Port kann im Konfigurationsmember `FEJJC�FG` festgelegt werden.
- Sperrdämon für Services zum Sperren von Dateien. Standardport ist 4036. Der Port kann im Konfigurationsmember `rsed.envvars` festgelegt werden.

- (Optional) File Manager-Integration für die Interaktion mit IBM File Manager, Standardport 1960
- (Optional) CARMA-Kommunikation (Standardportbereich 5227-5326, 100 Ports). Der Portbereich kann in der Konfigurationsdatei CRASRV.properties festgelegt werden.
- (Optional) Die APPC-Version von TSO Commands Service verwendet für die Kommunikation mit dem Sperrenmanager jedes verfügbare Socket. (Der Sperrenmanager stellt MVS-Dateien für Clients in eine Warteschlange.) Sie können keinen bestimmten Portbereich festlegen.

Anmerkung: Ältere Clientversionen (bis Version 7.0) kommunizieren direkt mit dem JES Job Monitor-Server am Standardport 6715.

CARMA und TCP/IP-Ports

In den meisten Fällen, beispielsweise bei einem RSE-Dämon, bindet ein Server an einen Port und wartet auf Verbindungsanforderungen. CARMA verwendet eine andere Methode, da der CARMA-Server während des Initialisierens der Verbindungsanforderung durch den Client noch nicht aktiv ist.

Wenn der Client eine Verbindungsanforderung sendet, sucht der CARMA-Miner, der als Benutzerthread in einem RSE-Thread-Pool aktiv ist, nach einem freien Port des Bereichs, der in der Konfigurationsdatei CRASRV.properties angegeben ist, und bindet an diesen Port. Der Miner startet anschließend den CARMA-Server und übergibt die Portnummer, sodass der Server eine Verbindung zu dem entsprechenden Port herstellen kann. Sobald der Server mit dem Port verbunden ist, kann der Client Anforderungen an den Server senden und Ergebnisse empfangen.

Aus Sicht des TCP/IP ist demnach RSE (über den CARMA-Miner) der Server, der an einen Port bindet, und der CARMA-Server der Client, der eine Verbindung mit dem Server herstellt.

PassTickets verwenden

Nach der Anmeldung kann mit PassTickets innerhalb des RSE-Servers die Thread-sicherheit gewährleistet werden. Dieses Feature kann nicht inaktiviert werden. PassTickets sind vom System generierte Kennwörter mit einer Lebensdauer von ca. 10 Minuten. Die generierten PassTickets basieren auf den DES-Verschlüsselungsalgorithmen, der Benutzer-ID, der Anwendungs-ID, einer Zeitmarke und einem geheimen Schlüssel. Dieser geheime Schlüssel ist eine 64-Bit-Zahl (16 Hexadezimalzeichen), die für Ihre Sicherheitssoftware definiert werden muss.

Nachfolgend sehen Sie eine kurze Beschreibung des RSE-Sicherheitsprozesses, die Ihnen helfen soll, die PassTicket-Verwendung zu verstehen.

1. Der Client stellt über den Host-Port 4035 eine Verbindung mit dem RSE-Dämon her.
2. Der RSE-Dämon authentifiziert den Client anhand der vom Client angegebenen Berechtigungsnachweise.
3. Der RSE-Dämon erstellt eine eindeutige Client-ID und einen RSE-Server-Thread.
4. Der RSE-Server generiert ein PassTicket und dann eine Sicherheitsumgebung für den Client, in der das PassTicket als Kennwort verwendet wird.
5. Der Client stellt zu dem vom RSE-Dämon zurückgegebenen Host-Port eine Verbindung her.

6. Der RSE-Server überprüft den Client anhand der Client-ID.
7. Für alle künftigen Aktionen, die ein Kennwort erfordern, verwendet der RSE-Server ein neu generiertes PassTicket.

Das eigentliche Kennwort des Clients wird nach der ersten Authentifizierung nicht mehr benötigt, da die mit SAF kompatiblen Sicherheitsprodukte sowohl PassTickets als auch reguläre Kennwörter überprüfen. Der RSE-Server generiert jedes Mal, wenn ein Kennwort erforderlich ist, ein PassTicket und verwendet dieses, so dass das Kennwort für den Client nur temporär gültig ist.

Durch die Verwendung von PassTickets kann RSE eine beliebige benutzerspezifische Sicherheitsumgebung einrichten, ohne dabei alle Benutzer-IDs und Kennwörter in einer Tabelle speichern zu müssen, was zu einer Beeinträchtigung führen könnte. Außerdem werden Clientauthentifizierungen ermöglicht, die keine wiederverwendbaren Kennwörter verwenden, wie X.509-Zertifikate.

Für Sicherheitsprofile in den Klassen APPL und PTKTDATA ist es erforderlich, PassTickets verwenden zu können. Diese Profile sind anwendungsspezifisch und haben daher keine Auswirkung auf Ihre aktuelle Systemkonfiguration.

Als Voraussetzung für anwendungsspezifische PassTickets müssen sowohl RSE als auch JES Job Monitor die gleiche Anwendungs-ID (APPLID) verwenden. Als Anwendungs-ID wird von beiden Servern standardmäßig FEKAPPL verwendet. Dies kann jedoch durch die Anweisung der APPLID in `rsed.envvars` für RSE und in `FEJCNFG` für JES Job Monitor geändert werden.

Sie sollten OMVSAPPL nicht als Anwendungs-ID verwenden, da diese ID den geheimen Schlüssel zu den meisten z/OS UNIX-Anwendungen entschlüsselt. Sie sollten ebenso nicht die standardmäßige MVS-Anwendungs-ID (MVS gefolgt von der SMF-ID des Systems) verwenden, da diese ID den geheimen Schlüssel zu den meisten MVS-Anwendungen (einschließlich Benutzer-Batch-Jobs) entschlüsselt.

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

Prüfprotokollierung

Developer for System z unterstützt die Prüfprotokollierung für Aktionen, die vom RSE-Dämon verwaltet werden. Die Prüfprotokolle werden als Textdateien im CSV-Format (Comma Separated Value) im Dämonprotokollverzeichnis gespeichert.

Steuerung der Prüffunktion

Die Prüffunktion wird von mehreren Optionen in `rsed.envvars` beeinflusst. Lesen Sie hierzu die Informationen unter „Zusätzliche Java-Startparameter mit `_RSE_JAVA_OPTS` definieren“ auf Seite 42.

- Die Aktivierung/Inaktivierung der Prüffunktion erfolgt über die Option `enable.audit.log`.
- Die Standardeinstellungen für die Prüfung werden von den Optionen `audit.*` gesteuert.
- Die Option `daemon.log` steuert die Position der Prüfprotokolldateien.
- Die Codepage, in der das Prüfprotokoll geschrieben wird, wird von der Anweisung `_RSE_HOST_CODEPAGE` gesteuert, wie im Abschnitt „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 33 dokumentiert.

Mit dem Bedienerbefehl **modify switch** kann manuell zu einer neuen Prüfprotokolldatei gewechselt werden (siehe Kapitel 8, „Bedienerbefehle“, auf Seite 125).

Wenn im Dateisystem mit den Prüfprotokolldateien nur noch ein kleiner freier Speicherbereich verfügbar ist, wird eine Warnung an die Konsole gesendet. Diese Konsolnachricht (FEK103E) wird immer wieder angezeigt, bis das Speicherproblem gelöst ist. Eine Liste der von RSE generierten Konsolnachrichten finden Sie im Abschnitt „Konsolnachrichten“ auf Seite 132.

Prüfdaten

Nach einer vordefinierten Zeit oder nach dem Absetzen des Bedienerbefehls **modify switch** wird eine neue Prüfprotokolldatei begonnen. Die alte Protokolldatei wird unter dem Namen `audit.log.jjjjmmdd.hhmmss` gespeichert. Der Abschnitt `jjjjmmdd.hhmmss` steht hier für die Datums-/Zeitmarke der Schließung dieses Protokolls. Die der Datei vom System zugeordnete Datums-/Zeitmarke (Datum und Uhrzeit) zeigt an, dass es sich um eine archivierte Protokolldatei handelt. Aus der Kombination der Zeitmarken zweier aufeinanderfolgender Prüfprotokolldateien können Sie den von der älteren Datei abgedeckten Zeitraum ersehen.

Folgende Aktionen werden protokolliert:

- Systemzugriff (Aufbau und Trennung von Verbindungen)
- JES-Spool-Zugriff (Submit, Display, Hold, Release, Cancel, Purge)
- Dateizugriff (READ, WRITE, CREATE, DELETE, RENAME, COMPRESS, MIGRATION, RECALL)
- Ausführung von TSO-Befehlen

Jede protokollierte Aktion wird (mit einer Datums-/Zeitmarke) im CSV-Format (Comma Separated Value) gespeichert, das von Automatisierungs- oder Datenanalysetools gelesen werden kann.

Prüfprotokolldateien haben die Berechtigungsbitmaske 640 (-rw-r-----). Dies bedeutet, der Eigner (z/OS UNIX-Benutzer-ID des RSE-Dämons) hat Lese- und Schreibzugriff auf die Dateien. Die Standardgruppe des Eigners hat Lesezugriff. Alle anderen Zugriffsversuche werden zurückgewiesen, sofern sie nicht von einem Superuser (UID 0) oder einer Person mit entsprechenden Berechtigungen für das Profil `SUPERUSER.FILESYS` in der Klasse `UNIXPRIV` unternommen werden.

JES-Sicherheit

Developer for System z ermöglicht Clients den Zugriff auf die JES-Spooldatei über JES Job Monitor. Der Server etabliert Basiszugriffsbeschränkungen, die Sie mit den Standardschutzfeatures für die Spooldatei in Ihrem Sicherheitsprodukt erweitern können. Bedieneraktionen für Spooldateien (Hold, Release, Cancel und Purge) werden über die EMCS-Konsole ausgeführt, für die bedingte Berechtigungen konfiguriert werden müssen.

Aktionen für Beschränkungen der Jobziele

JES Job Monitor ermöglicht Benutzern von Developer for System z keinen umfassenden JES-Spoolzugriff. Es stehen nur die Befehle 'Hold', 'Release', 'Cancel' und 'Purge' zur Verfügung und dies standardmäßig nur für Spooldateien, deren Eigner der Benutzer ist. Die Befehle werden durch Auswahl der entsprechenden Option in der Clientmenüstruktur abgesetzt (keine Eingabeaufforderung). Mit Sicherheitsprofilen, die definieren, für welche Jobs die Befehle verfügbar sind, kann der Geltungsbereich der Befehle ausgedehnt werden.

Ähnlich wie das SDSF-Aktionszeichen **SJ** unterstützt auch JES Job Monitor den Befehl 'JCL anzeigen', um die JCL abzurufen, die die ausgewählte Jobausgabe erstellt hat. Diese wird in einem Editierfenster angezeigt. JES Job Monitor ruft die JCL von JES ab. Dies ist eine hilfreiche Funktion in Situationen, in denen das ursprüngliche JCL-Member nicht einfach auffindbar ist.

Tabelle 21. JES Job Monitor, Konsolbefehle

Aktion	JES2	JES3
Hold	\$Hx(jobid) x = {J, S oder T}	*F,J=jobid,H
Release	\$Ax(jobid) x = {J, S oder T}	*F,J=jobid,R
Cancel	\$Cx(jobid) x = {J, S oder T}	*F,J=jobid,C
Purge	\$Cx(jobid),P x = {J, S oder T}	*F,J=jobid,C
JCL anzeigen	Nicht zutreffend	Nicht zutreffend

Die verfügbaren JES-Befehle, die in Tabelle 21 aufgelistet sind, sind standardmäßig auf Jobs beschränkt, deren Eigner der Benutzer ist. Mit der Anweisung **LIMIT_COMMANDS** kann dies geändert werden (siehe Abschnitt „Konfigurationsdatei für JES Job Monitor (FEJJCNFG)“ auf Seite 29).

Tabelle 22. Matrix der Befehlsberechtigungen für LIMIT_COMMANDS

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

Für den Schutz von SYSIN/SYSOUT-Dateien verwendet das JES die Klasse JESSPOOL. Ähnlich wie SDSF, wendet JES Job Monitor die Klasse JESSPOOL auch auf den Schutz von Jobressourcen an.

Wenn **LIMIT_COMMANDS** nicht auf **USERID** gesetzt ist, fordert JES Job Monitor die Berechtigung für das entsprechende Profil in der Klasse JESSPOOL an, wie in der folgenden Tabelle aufgeführt.

Tabelle 23. Erweiterte JESSPOOL-Profile

Befehl	JESSPOOL-Profil	Erforderlicher Zugriff
Hold	nodeid.userid.jobname.jobid	ALTER

Tabelle 23. Erweiterte JESSPOOL-Profil (Forts.)

Befehl	JESSPOOL-Profil	Erforderlicher Zugriff
Release	nodeid.userid.jobname.jobid	ALTER
Cancel	nodeid.userid.jobname.jobid	ALTER
Purge	nodeid.userid.jobname.jobid	ALTER
JCL anzeigen	nodeid.userid.jobname.jobid.JCL	READ

Verwenden Sie in der vorherigen Tabelle die folgenden Ersetzungen:

Knoten-ID	NJE-Knoten-ID des Ziel-JES
Benutzer-ID	Lokale Benutzer-ID des Jobeigners
Jobname	Name des Jobs
Job-ID	JES-Job-ID

Wenn die Klasse JESSPOOL nicht aktiv ist, bewirken die Werte LIMITED und NOLIMIT für LIMIT_COMMANDS ein unterschiedliches Verhalten (siehe Tabelle 9 auf Seite 32). Das Verhalten beider Werte ist identisch, wenn JESSPOOL aktiv ist, denn die Klasse verweigert den Zugriff standardmäßig, wenn ein Profil nicht definiert ist.

Aktionen für Beschränkungen der Jobausführung

Nachdem die zulässigen Ziele angegeben sind, besteht die zweite Phase der Befehlssicherheit für JES-Spoolprogramme aus den Berechtigungen, die für das tatsächliche Ausführen des Bedienerbefehls erforderlich sind. Die Sicherheitsprüfungen für z/OS und JES erzwingen diese Ausführungsberechtigungen.

Beachten Sie, dass der Befehl 'JCL anzeigen' kein Bedienerbefehl wie die anderen JES Job Monitor-Befehle (Hold, Release, Cancel und Purge) ist. Daher finden die unteren Beschränkungen keine Anwendung, denn es findet keine weitere Sicherheitsprüfung statt.

JES Job Monitor setzt alle von einem Benutzer angeforderten JES-Bedienerbefehle über eine erweiterte MCS-Konsole (EMCS) ab, deren Bezeichnung durch die Anweisung `CONSOLE_NAME` gesteuert wird, wie im Abschnitt „Konfigurationsdatei für JES Job Monitor (FEJJCNFNG)“ auf Seite 29 dokumentiert.

Der Sicherheitsadministrator kann mit dieser Konfiguration unter Verwendung der Klassen OPERCMDS und CONSOLE differenzierte Berechtigungen zur Befehlsausführung definieren.

- Um eine EMCS-Konsole verwenden zu können, muss der Benutzer (mindestens) über eine Leseberechtigung für das Profil `MVS.MCSOPER.console-name` in der Klasse OPERCMDS verfügen. Beachten Sie, dass das System die Berechtigung für die Anforderung gewährt, wenn kein Profil definiert ist.
- Damit ein JES-Bedienerbefehl ausgeführt werden kann, muss ein Benutzer über eine ausreichende Berechtigung für das Profil `JES%.**` (oder spezifischer) in der Klasse OPERCMDS verfügen. Beachten Sie, dass JES den Befehl nicht ausführen kann, wenn kein Profil definiert ist bzw. die Klasse OPERCMDS nicht aktiv ist.

- Der Sicherheitsadministrator kann auch festlegen, dass ein Benutzer JES Job Monitor zur Ausführung des Bedienerbefehls verwenden muss. Dazu legt er `WHEN(CONSOLE(JMON))` in der Definition **PERMIT** fest. Damit diese Konfiguration verwendet werden kann, muss die Klasse `CONSOLE` aktiv sein. Hinweis: Es genügt, wenn die Klasse `CONSOLE` aktiv ist. Für die EMCS-Konsolen werden keine weiteren Profile überprüft.

Ihre Sicherheitssoftware verhindert, dass ein Benutzer in einer TSO-Sitzung eine Konsole `JMON` erstellt, weil er sich so als JES Job Monitor-Server ausgeben könnte. Auch wenn die Konsole erstellt werden kann, unterscheidet sich der Eingangsport (JES Job Monitor oder TSO). Von dieser Konsole abgesetzte JES-Befehle werden jedoch nicht die Sicherheitsprüfung bestehen, wenn Ihre Sicherheitssoftware wie in dieser Veröffentlichung beschrieben konfiguriert und der Benutzer nicht autorisiert ist, JES-Befehle über andere Mechanismen zu verwenden.

Beachten Sie, dass JES Job Monitor die Konsole zur Ausführung eines Befehls nicht erstellen kann, wenn der Konsolname bereits verwendet wird. Um dies zu verhindern, kann der Systemprogrammierer in der JES Job Monitor-Konfigurationsdatei die Anweisung `GEN_CONSOLE_NAME=ON` setzen. Alternativ kann der Sicherheitsadministrator Sicherheitsprofile definieren, um zu verhindern, dass TSO-Benutzer eine Konsole erstellen. Die folgenden RACF-Beispielbefehle hindern jeden Benutzer (mit Ausnahme berechtigter Benutzer) am Erstellen einer TSO- oder SDSF-Konsole:

- `RDEFINE TSOAUTH CONSOLE UACC(NONE)`
- `PERMIT CONSOLE CLASS(TSOAUTH) ACCESS(READ) ID(#userid)`
- `RDEFINE SDSF ISFCMD.ODSP.ULOG.* UACC(NONE)`
- `PERMIT ISFCMD.ODSP.ULOG.* CLASS(SDSF) ACCESS(READ) ID(#userid)`

Anmerkung: Benutzer, die nicht berechtigt sind, diese Bedienerbefehle auszuführen, können trotzdem mit JES Job Monitor Jobs übergeben und Jobausgaben lesen, sofern sie über eine ausreichende Berechtigung für eventuelle Profile verfügen, die diese Ressourcen schützen (z. B. diejenigen in den Klassen `JESINPUT`, `JESJOBS` und `JESSPOOL`).

Weitere Informationen zum Bedienerbefehlsschutz finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Zugriff auf Spooldateien

JES Job Monitor erlaubt standardmäßig die Anzeige aller Spooldateien. Mit der Anweisung `LIMIT_VIEW` kann dies geändert werden (siehe Abschnitt „Konfigurationsdatei für JES Job Monitor (FEJJC�FG)“ auf Seite 29).

Tabelle 24. Berechtigungsmatrix zum Durchsuchen für `LIMIT_VIEW`

LIMIT_VIEW	Jobeigner	
	Benutzer	Anderer Eigner
USERID	Zulässig	Nicht zulässig
NOLIMIT (default)	Zulässig	Zulässig, wenn die Sicherheitsprofile die Berechtigung enthalten oder die JESSPOOL-Klasse nicht aktiv ist

Wenn Benutzer nur ihre eigenen JES-Spool-Jobs anzeigen können sollen, definieren Sie in der Konfigurationsdatei von JES Job Monitor (FEJJC�FG) die Anweisung `"LIMIT_VIEW=USERID"`. Falls Benutzer auf weitere Jobs zugreifen müssen, jedoch nicht auf alle Jobs, können Sie die Standardschutzfeatures für Spooldateien verwenden, beispielsweise die Klasse JESSPOOL.

Denken Sie beim Definieren weiterer Schutzmaßnahmen daran, dass JES Job Monitor für den Zugriff auf Spooldateien (die SYSOUT-Anwendungsprogrammierschnittstelle) SAPI verwendet. Damit ist impliziert, dass der Benutzer für Spooldateien (selbst zum Anzeigen) zumindest die Berechtigung `UPDATE` haben muss. Diese Voraussetzung gilt nicht unter z/OS ab Version 1.7 (für JES3 unter z/OS ab Version 1.8). Für Anzeigefunktionen ist die Berechtigung `READ` ausreichend.

Weitere Informationen zum Schutz von JES-Spooldateien finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Mit SSL verschlüsselte Kommunikation

Die externe Kommunikation (Client-Host) kann mit SSL (Secure Socket Layer) verschlüsselt werden. Dieses Feature ist standardmäßig inaktiviert und wird von den Einstellungen in `ssl.properties` gesteuert, wie im Abschnitt „RSE-SSL-Verschlüsselung (optional)“ auf Seite 93 dokumentiert.

Aufgrund unterschiedlicher Architektur unterstützen der RSE-Dämon und der RSE-Server verschiedene Mechanismen zum Speichern von Zertifikaten. Dies impliziert, dass für den RSE-Dämon sowie den RSE-Server SSL-Definitionen und -Zertifikate erforderlich sind. Ein gemeinsam genutztes Zertifikat kann verwendet werden, wenn der RSE-Dämon und der RSE-Server dieselbe Zertifikatsverwaltungsmethode verwenden.

Tabelle 25. Mechanismen für den SSL-Zertifikatsspeicher

Zertifikatsspeicher	Erstellt und verwaltet von	RSE-Dämon	RSE-Server
Schlüsseldatei	SAF-konformes Sicherheitsprodukt	unterstützt	unterstützt
Schlüsseldatenbank	gskkyman von z/OS UNIX	unterstützt	/
Keystore	Java-Keytool	/	unterstützt

Anmerkung: Für die Verwaltung von Zertifikaten sind SAF-kompatible Schlüsseldateien die bevorzugte Methode.

SAF-konforme Schlüsseldateien können den privaten Schlüssel eines Zertifikats entweder in der Sicherheitsdatenbank oder mithilfe von ICSF (Integrated Cryptographic Service Facility) speichern, der Schnittstelle für Verschlüsselungshardware von System z.

ICSF wird für die Speicherung von privaten Schlüsseln für digitale Zertifikate empfohlen, da diese Methode sicherer als andere Lösungen zur Verwaltung von privaten Schlüsseln ohne ICSF ist. Mit ICSF wird sichergestellt, dass private Schlüssel unter dem ICSF-Masterschlüssel verschlüsselt werden und dass der Zugriff auf diese Schlüssel von allgemeinen Ressourcen in den Sicherheitsklassen CSFKEYS und CSFSERV gesteuert wird. Außerdem bietet ICSF eine bessere Betriebsleistung, da bei dieser Methode die Hardware Cryptographic Coprocessor verwendet.

Zum Verwalten von Kommunikation, die mit SSL verschlüsselt ist, verwendet der RSE-Dämon System SSL-Funktionen. Dies impliziert, dass SYS1.SIEALNKE von Ihrer Sicherheitssoftware programmgesteuert und RSE über LINKLIST oder die STEPLIB-Anweisung in rsed.envvars verfügbar sein muss.

Wenn SAF-konforme Schlüsseldateien für den RSE-Dämon oder RSE-Server verwendet werden, ist für die RSE-Benutzer-ID (STCRSE in den unteren Beispielbefehlen) die Genehmigung für den Zugriff auf die Schlüsseldatei und die zugeordneten Zertifikate erforderlich.

- RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
- RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
- PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ACCESS(READ) ID(stcrse)
- PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ACCESS(READ) ID(stcrse)
- SETROPTS RACLIST(FACILITY) REFRESH

Weitere Details zur Aktivierung von SSL für Developer for System z finden Sie in Anhang A, „SSL- und X.509-Authentifizierung konfigurieren“, auf Seite 299.

Clientauthentifizierung unter Verwendung von X.509-Zertifikaten

Mit einem X.509-Zertifikat unterstützt der RSE-Dämon die eigene Authentifizierung der Benutzer. Voraussetzung hierfür ist die Verwendung der mit SSL verschlüsselten Kommunikation, da dies eine Erweiterung der Hostauthentifizierung mit einem in SSL verwendeten Zertifikat ist.

Der RSE-Dämon startet den Prozess zur Clientauthentifizierung mit der Prüfung des Clientzertifikats. Einige wichtige Aspekte, die geprüft werden, sind die Gültigkeitsdaten des Zertifikats und die Vertrauenswürdigkeit der Zertifizierungsstelle (CA), die das Zertifikat unterzeichnet hat. Optional kann auch eine Zertifikatswiderrufsliste (CRL) eines anderen Anbieters zu Rate gezogen werden.

Nachdem der RSE-Dämon das Zertifikat geprüft hat, ist es zur Authentifizierung bereit. Das Zertifikat wird an Ihr Sicherheitsprodukt zur Authentifizierung weitergegeben, sofern die `rsed.envvars`-Anweisung `enable.certificate.mapping` nicht auf `false` gesetzt ist. In diesem Fall führt der RSE-Dämon die Authentifizierung durch.

Bei erfolgreicher Authentifizierung legt der Authentifizierungsprozess die in dieser Sitzung zu verwendende Benutzer-ID fest. Diese wird dann vom RSE-Dämon getestet, um sicherzustellen, dass sie für das Hostsystem verwendbar ist, auf dem der RSE-Dämon aktiv ist.

In der letzten Überprüfung (die nicht nur bei Authentifizierungsverfahren mit X.509-Zertifikaten durchgeführt wird, sondern bei allen Verfahren) wird die Berechtigung der Benutzer-ID für die Verwendung von Developer for System z überprüft.

Wenn Sie mit den von TCP/IP verwendeten SSL-Sicherheitsklassifikationen vertraut sind: Die Kombination dieser Überprüfungsschritte entspricht den Spezifikationen der Stufe 3 der Clientauthentifizierung (höchste verfügbare Stufe).

Prüfung der Zertifizierungsstelle (CA)

Ein Bestandteil des Zertifikatsüberprüfungsprozesses besteht in der Prüfung, ob das Zertifikat von einer vertrauenswürdigen Zertifizierungsstelle (CA) unterschrieben wurde. Um dies ausführen zu können, muss der RSE-Dämon Zugriff auf ein Zertifikat haben, das die CA identifiziert.

Wenn für Ihre SSL-Verbindung die **gskkyman**-Schlüsseldatenbank verwendet wird, muss das CA-Zertifikat der Schlüsseldatenbank hinzugefügt werden.

Wenn eine SAF-Schlüsseldatei verwendet wird (empfohlene Methode), müssen Sie Ihrer Sicherheitsdatenbank das CA-Zertifikat als ein CERTAUTH-Zertifikat mit dem TRUST- oder HIGHTRUST-Attribut hinzufügen, wie in dem folgenden RACF-Beispielbefehl gezeigt wird:

- `RACDCERT CERTAUTH ADD(dsn) HIGHTRUST WITHLABEL('label')`

Beachten Sie, dass in den Datenbanken der meisten Sicherheitsprodukte bereits Zertifikate von bekannten CAs mit dem Status NOTRUST vorhanden sind. Verwenden Sie die folgenden RACF-Beispielbefehle, um die vorhandenen CA-Zertifikate aufzulisten und um ein Zertifikat, basierend auf der zugeordneten Bezeichnung, als vertrauenswürdig zu markieren.

- `RACDCERT CERTAUTH LIST`
- `RACDCERT CERTAUTH ALTER(LABEL('HighTrust CA')) HIGHTRUST`

Anmerkung: Der HIGHTRUST-Status ist erforderlich, wenn Sie eine RACF-Authentifizierung des Benutzers zugrunde legen, die auf der HostId-Mappings-Erweiterung im Zertifikat basiert. Weitere Informationen hierzu enthält der Abschnitt „Authentifizierung durch Ihre Sicherheitssoftware“ auf Seite 173.

Sobald ein CA-Zertifikat Ihrer Sicherheitsdatenbank hinzugefügt ist, muss es mit der RSE-Schlüsseldatei verbunden werden, wie in dem folgenden RACF-Beispielbefehl gezeigt wird:

- `RACDCERT ID(stcrse) CONNECT(CERTAUTH LABEL('HighTrust CA')
RING(rdzssl.racf))`

Weitere Informationen zum **RACDCERT**-Befehl enthält die Veröffentlichung *Security Server RACF Command Language Reference* (IBM Form SA22-7687).

Achtung: Wenn zur Authentifizierung eines Benutzers der RSE-Dämon anstelle Ihrer Sicherheitssoftware zugrunde gelegt wird, müssen Sie darauf achten, in der SAF-Schlüsseldatei bzw. der **gskkyman**-Schlüsseldatenbank die CAs mit den TRUST- und HIGHTRUST-Status nicht zu vermischen. Der RSE-Dämon kann zwischen diesen beiden Status nicht unterscheiden. Daher sind Zertifikate, die von einer CA mit TRUST-Status unterschrieben wurden, zur Authentifizierung der Benutzer-ID gültig.

Zertifikatswiderrufsliste (CRL) abfragen (optional)

Falls gewünscht, können Sie den RSE-Dämon anweisen, eine oder mehrere Zertifikatswiderrufslisten (CRL) zu überprüfen. Dies erweitert den Sicherheitsschutz des Überprüfungsprozesses. Dafür werden der Datei `rsed.envvars` CRL-bezogene Umgebungsvariablen hinzugefügt. Informationen zu diesen Beispielvariablen enthält der Abschnitt „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 33:

- `GSK_CRL_SECURITY_LEVEL`
- `GSK_LDAP_SERVER`
- `GSK_LDAP_PORT`
- `GSK_LDAP_USER`
- `GSK_LDAP_PASSWORD`

Weitere Informationen zu diesen und weiteren Umgebungsvariablen, die von z/OS System SSL verwendet werden, finden Sie in der Veröffentlichung *Cryptographic Services System Secure Sockets Layer Programming* (IBM Form SC24-5901).

Anmerkung: Vorsicht bei der Angabe anderer z/OS System SSL-Umgebungsvariablen (`GSK_*`) in `rsed.envvars`, da dies Auswirkungen darauf haben kann, wie der RSE-Dämon SSL-Verbindungen und die Zertifikatsauthentifizierung ausführt.

Authentifizierung durch Ihre Sicherheitssoftware

RACF führt verschiedene Überprüfungen zum Authentifizieren eines Zertifikats aus und gibt die zugeordnete Benutzer-ID zurück. Beachten Sie, dass andere Sicherheitsprodukte dies anders handhaben können. Weitere Informationen zur `initACEE`-Funktion, die für die Authentifizierung (Abfragemodus) verwendet wird, finden Sie in der Dokumentation Ihres Sicherheitsprodukts.

1. RACF überprüft, ob das Zertifikat in der DIGTCERT-Klasse definiert ist. Falls dies der Fall ist, gibt RACF die Benutzer-ID zurück, die diesem Zertifikat beim Hinzufügen zur RACF-Datenbank zugeordnet wurde.

Zertifikate werden mithilfe des **RACDCERT**-Befehls in RACF definiert, wie das folgende Beispiel zeigt:

```
RACDCERT ID(userid) ADD(dsn) TRUST WITHLABEL('Bezeichnung')
```

2. Wenn das Zertifikat nicht definiert ist, überprüft RACF, ob ein entsprechender Zertifikatsnamensfilter in den Klassen DIGTNMAP oder DIGTCRIT definiert ist. Wenn dies der Fall ist, gibt es die Benutzer-ID zurück, die dem passendsten Filter zugeordnet ist.

Anmerkung: Es wird empfohlen, für Zertifikate, die von Developer for System z verwendet werden, keine Namensfilter zu verwenden, da diese Filter alle Zertifikate einer einzigen Benutzer-ID zuordnen. Dies bedeutet, dass sich alle Benutzer von Developer for System z mit derselben Benutzer-ID anmelden.

3. Wenn kein passender Namensfilter vorhanden ist, sucht RACF die HostIdMappings-Zertifikatserweiterung und extrahiert die eingebettete Benutzer-ID und das Hostnamenspaar. Wenn er jedoch gefunden und überprüft wird, gibt RACF die Benutzer-ID zurück, die in der HostIdMappings-Erweiterung definiert ist. Die Benutzer-ID und das Hostnamenspaar sind gültig, wenn alle folgenden Bedingungen wahr sind:

- Das CA-Zertifikat, das zur Unterzeichnung dieses Zertifikats verwendet wird, ist in der DIGTCERT-Klasse als HIGHTRUST markiert.
- Die in der Erweiterung gespeicherte Benutzer-ID besitzt eine gültige Länge (1 bis 8 Zeichen).
- Die dem RSE-Dämon zugeordnete Benutzer-ID verfügt für das IRR.HOST-
.hostname-Profil in der SERVAUTH-Klasse über (mindestens) LESEBERECHTIGUNG, wobei der hostname dem in der Erweiterung gespeicherten Hostnamen entspricht. Hierbei handelt es sich normalerweise um einen Domännennamen, wie CDFMVS08.RALEIGH.IBM.COM.

Die Definition der HostIdMappings-Erweiterung lautet in ASN.1-Syntax:

```
id-ce-hostIdMappings OBJECT IDENTIFIER ::= { 1 3 18 0 2 18 1 }
HostIdMappings ::= SET OF HostIdMapping
HostIdMapping ::= SEQUENCE {
    hostName      IMPLICIT[1] IA5String,
    subjectId     IMPLICIT[2] IA5String,
    proofOfIdPossession IdProof OPTIONAL
}
IdProof ::= SEQUENCE {
    secret        OCTET STRING,
    encryptionAlgorithm OBJECT IDENTIFIER
}
```

Anmerkung: Eine HostIdMappings-Erweiterung wird nicht berücksichtigt, wenn die Zielbenutzer-ID nach Beginn des Gültigkeitszeitraums des Zertifikats erstellt wurde, das die HostIdMappings-Erweiterung enthält. Stellen Sie daher sicher, dass Sie beim Erstellen von Benutzer-IDs speziell für Zertifikate mit HostIdMappings-Erweiterungen die Benutzer-IDs erstellen, bevor die Zertifikatsanforderungen übergeben werden.

Weitere Informationen zu X.509-Zertifikaten und ihrer Verwaltung in RACF sowie zur Vorgehensweise der Definition von Zertifikatsnamensfiltern finden Sie in der Veröffentlichung *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683). Weitere Informationen zum **RACDCERT**-Befehl enthält die Veröffentlichung *Security Server RACF Command Language Reference* (IBM Form SA22-7687).

Authentifizierung durch den RSE-Dämon

Developer for System z kann eine grundlegende X.509-Zertifikatsauthentifizierung durchführen, ohne dabei auf Ihr Sicherheitsprodukt zurückzugreifen. Für die Au-

thentifizierung durch den RSE-Dämon sind eine Benutzer-ID und ein Hostname erforderlich, die in der Zertifikatserweiterung definiert sein müssen. Die Authentifizierung ist nur dann aktiviert, wenn in der Datei `rsed.envvars` die Anweisung `enable.certificate.mapping` auf `FALSE` gesetzt ist.

Diese Funktion ist vorgesehen, falls Ihr Sicherheitsprodukt keine Benutzerauthentifizierung unterstützt, die auf einem X.509-Zertifikat basiert, oder falls Ihr Zertifikat die von Ihrem Sicherheitsprodukt durchgeführten Tests nicht bestehen würde (z. B. wenn das Zertifikat für die `HostIdMappings`-Erweiterung über eine falsche ID verfügt und kein Namensfilter oder keine Namensdefinition in `DIGTCERT` festgelegt wurde).

Der Client fragt den Benutzer nach der zu verwendenden Objektkennung (OID). Standardmäßig wird die `HostIdMappings`-OID verwendet `{1 3 18 0 2 18 1}`.

Der RSE-Dämon extrahiert davon die Benutzer-ID und den Hostnamen. Dabei wird das Format der `HostIdMappings`-Erweiterung verwendet. Dieses Format ist im Abschnitt „Authentifizierung durch Ihre Sicherheitssoftware“ auf Seite 173 beschrieben.

Die Benutzer-ID und das Hostnamenspaar sind gültig, wenn alle folgenden Bedingungen wahr sind:

- Die in der Erweiterung gespeicherte Benutzer-ID besitzt eine gültige Länge (1 bis 8 Zeichen).
- Die dem RSE-Dämon zugeordnete Benutzer-ID verfügt für das `IRR.HOST.hostname`-Profil in der `SERVAUTH`-Klasse über (mindestens) `LESEBERECHTIGUNG`, wobei der `hostname` dem in der Erweiterung gespeicherten Hostnamen entspricht. Hierbei handelt es sich normalerweise um einen Domännennamen, wie `CDFMVS08.RALEIGH.IBM.COM`.

Achtung: Der Sicherheitsadministrator muss sicherstellen, dass alle dem RSE-Dämon bekannten CAs sehr vertrauenswürdig sind, da der RSE-Dämon nicht überprüfen kann, ob der Unterzeichner des Clientzertifikats sehr vertrauenswürdig oder nur vertrauenswürdig ist. Weitere Informationen zu zugänglichen CA-Zertifikaten enthält der Abschnitt „Prüfung der Zertifizierungsstelle (CA)“ auf Seite 172.

Eingangsport (POE) überprüfen

Developer for System z unterstützt die Prüfung des Eingangsports, die eine Beschränkung des Hostzugriffs auf anerkannte TCP/IP-Adressen ermöglicht. Dieses Feature ist standardmäßig inaktiviert und erfordert die Definition des Sicherheitsprofils `BPX.POE`. Vergleichen Sie hierzu die folgenden RACF-Beispielbefehle:

- `RDEFINE FACILITY BPX.POE UACC(NONE)`
- `PERMIT BPX.POE CLASS(FACILITY) ACCESS(READ) ID(STCRSE)`
- `SETOPTS RACLIST(FACILITY) REFRESH`

Anmerkung:

- RSE muss für die Prüfung des Eingangsports konfiguriert werden. Entfernen Sie dazu in `rsed.envvars` das Kommentarzeichen vor der Option `“enable.port.of.entry=true”` (siehe Abschnitt „Zusätzliche Java-Startparameter mit `_RSE_JVAOPTS` definieren“ auf Seite 42).

- Die RSE-Benutzer-ID STCRSE erfordert UID(0), wenn dieses Profil nicht definiert ist und die Prüfung des Eingangsports in `rsed.envvars` aktiviert ist.
- Das Definieren von BPX.P0E wirkt sich auf andere TC/PIP-Anwendungen aus, die die Prüfung des Eingangsports unterstützen, beispielsweise INETD.
- In der Klasse SERVAUTH sollten Sicherheitszonen (EZB.NETACCESS.**-Profile, die IP-Adressräume sind) konfiguriert werden, um die Möglichkeiten der Eingangsportüberprüfung voll auszuschöpfen.

Weitere Informationen zur Kontrolle des Netzzugriffs durch die Eingangsportüberprüfung enthält die Veröffentlichung *Communications Server IP Configuration Guide* (IBM Form SC31-8775).

CICSTS-Sicherheit

Developer for System z ermöglicht CICS-Administratoren über Application Deployment Manager die für den Entwickler editierbaren CICS-Ressourcendefinitionen, deren Standardwerte sowie die Anzeige einer CICS-Ressourcendefinition mithilfe des CICS Resource Definition-Servers (CRD) zu steuern. Weitere Informationen zu den erforderlichen CICS-TS-Sicherheitsdefinitionen enthält Kapitel 15, „CICSTS-Aspekte“, auf Seite 257.

CRD-Repository

Die VSAM-Datei für das CRD-Server-Repository enthält alle Standardressourcendefinitionen und muss daher vor Aktualisierungen geschützt werden. Entwickler müssen jedoch die Möglichkeit haben, die hier gespeicherten Werte zu lesen.

CICS-Transaktionen

Developer for System z stellt mehrere Transaktionen bereit, die der CRD-Server beim Definieren und Abfragen von CICS-Ressourcen verwendet. Wenn die Transaktion zugeordnet wird, stellt die Sicherheitsprüfung für CICS-Ressourcen (sofern aktiviert) sicher, dass die Benutzer-ID berechtigt ist, die Transaktions-ID zu verwenden.

Mit SSL verschlüsselte Kommunikation

Die Clientkomponente von Application Deployment Manager ruft mit CICS-TS-Web-Services oder der RESTful-Schnittstelle den CRD-Server auf. Die Verwendung von SSL für diese Kommunikation wird von der CICS-TS-Definition TCPIPSERVICE gesteuert. Diese Art der Verwendung ist im *RACF Security Guide for CICS TS* dokumentiert.

SCLM-Sicherheit

SCLM Developer Toolkit stellt optionale Sicherheitsfunktionen für die Builderstellung, die Umstufung und das Deployment bereit.

Wenn ein SCLM-Administrator die Sicherheit für eine Funktion aktiviert hat, wird SAF aufgerufen, um zu überprüfen, ob die geschützte Funktion mit der ID des Aufrufenden oder einer Ersatzbenutzer-ID ausgeführt werden darf.

Weitere Informationen zu den erforderlichen SCLM-Sicherheitsdefinitionen enthält der *SCLM Developer Toolkit Administrator's Guide* (IBM Form SC23-9801).

Konfigurationsdateien von Developer for System z

Es gibt mehrere Konfigurationsdateien für Developer for System z, deren Auswirkungen auf die Sicherheitskonfiguration haben. Auf der Basis der Informationen in diesem Kapitel können der Sicherheitsadministrator und Systemprogrammierer entscheiden, welche Einstellungen für die folgenden Anweisungen verwendet werden sollten.

JES Job Monitor - FEJJCNFG

- `LIMIT_COMMANDS={USERID | LIMITED | NOLIMIT }`

Definieren, welche Jobaktionen durchgeführt werden können (mit Ausnahme von 'Durchsuchen' und 'Übergeben'). Weitere Informationen enthält der Abschnitt „Aktionen für Beschränkungen der Jobziele“ auf Seite 166.

- `LIMIT_VIEW={USERID | NOLIMIT}`

Definieren, welche Spooldateien durchsucht werden können. Weitere Informationen enthält der Abschnitt „Zugriff auf Spooldateien“ auf Seite 170.

- `APPLID={FEKAPPL | *}`

Die Anwendungs-ID, die zum Erstellen/Überprüfen von PassTicket verwendet wird. Weitere Informationen enthält der Abschnitt „PassTickets verwenden“ auf Seite 164.

Anmerkung: Details zu diesen und anderen FEJJCNFG-Anweisungen finden Sie im Abschnitt „Konfigurationsdatei für JES Job Monitor (FEJJCNFG)“ auf Seite 29.

RSE - rsed.envvars

- `(_RSE_JAVAOPTS) -DDENY_PASSWORD_SAVE={true | false}`

Lehnen Sie das Speichern von Hostkennwörtern auf dem Client seitens der Benutzer ab. Weitere Informationen enthält der Abschnitt „Zusätzliche Java-Startparameter mit _RSE_JAVAOPTS definieren“ auf Seite 42.

- `(_RSE_JAVAOPTS) -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=value`

Zeitgeber zum Trennen der Verbindung inaktiver Clients. Weitere Informationen enthält der Abschnitt „Zusätzliche Java-Startparameter mit _RSE_JAVAOPTS definieren“ auf Seite 42.

- `(_RSE_JAVAOPTS) -DAPPLID={FEKAPPL | *}`

Die Anwendungs-ID, die zum Erstellen/Überprüfen von PassTicket verwendet wird. Weitere Informationen enthält der Abschnitt „PassTickets verwenden“ auf Seite 164.

- `(_RSE_JAVAOPTS) -Denable.port.of.entry={true | false}`

Aktivieren der Überprüfung des Eingangsports. Weitere Informationen enthält der Abschnitt „Eingangsport (POE) überprüfen“ auf Seite 175.

- `(_RSE_JAVAOPTS) -Denable.certificate.mapping={true | false}`

Verwenden Sie Ihr Sicherheitsprodukt zum Authentifizieren der Benutzer mit einem X.509-Zertifikat. Weitere Informationen enthält der Abschnitt „Clientauthentifizierung unter Verwendung von X.509-Zertifikaten“ auf Seite 171.

- `(_RSE_JAVAOPTS) -Ddaemon.log={/var/rdz/logs | *}`

Position der Prüfprotokolldateien. Weitere Informationen enthält der Abschnitt „Prüfprotokollierung“ auf Seite 165.

Anmerkung: Details zu diesen und anderen rsed.envvars-Anweisungen finden Sie im Abschnitt „RSE-Konfigurationsdatei rsed.envvars“ auf Seite 33.

RSE - ssl.properties

- `daemon_keydb_file={SAF-Schlüsseldateiname | gskkyman-Schlüsseldatenbankname}`
Position des RSE-Dämonzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL verschlüsselte Kommunikation“ auf Seite 170.
- `daemon_key_label=Zertifikatsbezeichnung`
Name des RSE-Dämonzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL verschlüsselte Kommunikation“ auf Seite 170.
- `server_keystore_file={SAF-Schlüsseldateiname | Java-Schlüsseldateiname}`
Position des RSE-Serverzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL verschlüsselte Kommunikation“ auf Seite 170.
- `server_keystore_label=Zertifikatsbezeichnung`
Name des RSE-Serverzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL verschlüsselte Kommunikation“ auf Seite 170.
- `server_keystore_type={JKS | JCECARACFKS | JCECCARACFKS}`
Verwendeter Keystoretyp (Java-Keystore oder SAF-Schlüsseldatei). Weitere Informationen enthält der Abschnitt „Mit SSL verschlüsselte Kommunikation“ auf Seite 170.

Anmerkung: Details zu diesen und anderen `ssl.properties`-Anweisungen finden Sie im Abschnitt „RSE-SSL-Verschlüsselung (optional)“ auf Seite 93.

Sicherheitsdefinitionen

Passen Sie das Beispielmembere FEKRACF an, das RACF- und z/OS UNIX-Beispielbefehle enthält, und übergeben Sie es, um die Basissicherheitsdefinitionen für Developer for System z zu erstellen.

FEKRACF befindet sich in `FEK.#CUST.JCL`, sofern Sie bei der Anpassung und Übergabe des Jobs `FEK.SFEKSAMP(FEKSETUP)` keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

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

Anmerkung:

- Für Sites, die CA ACF2™ for z/OS verwenden, rufen Sie den folgenden Link auf, um Details zu den Befehlen für die Sicherheitsfunktion zu erhalten, die für eine ordnungsgemäße Konfiguration von Developer for System z erforderlich sind: <https://support.ca.com/irj/portal/kbtech?ipLogNrow=0&docid=492389&searchID=TEC492389>.
- Für Sites, die CA Top Secret® for z/OS verwenden, rufen Sie die Seite Ihres Produkts auf der CA-Unterstützungssite (<https://support.ca.com>) auf und überprüfen Sie das zugehörige Dokument mit Informationen zu Developer for System z. Dieses Dokument enthält Details zu den Befehlen für die Sicherheitsfunktion, die für die ordnungsgemäße Konfiguration von Developer for System z erforderlich sind.

In den folgenden Abschnitten sind die erforderlichen Schritte, die optionale Konfiguration und mögliche Alternativen beschrieben.

Voraussetzungen und Prüfliste

Der Sicherheitsadministrator muss die in Tabelle 26 aufgelisteten Werte kennen, um die Sicherheitskonfiguration abzuschließen. Diese Werte wurden in früheren Schritten der Installation und Anpassung von Developer for System z definiert.

Tabelle 26. Variablen für die Sicherheitskonfiguration

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Übergeordnetes Qualifikationsmerkmal für Developer for System z	<ul style="list-style-type: none"> • FEK • SMP/E-Installation 	
Übergeordnetes Qualifikationsmerkmal für die Anpassung von System z	<ul style="list-style-type: none"> • FEK.#CUST • FEK.SFEKSAMP(FEKSETUP), wie in „Anpassungskonfiguration“ auf Seite 15 beschrieben 	
Name der gestarteten Task von JES Job Monitor	<ul style="list-style-type: none"> • JMON • FEK.#CUST.PROCLIB(JMON), wie in „PROCLIB-Änderungen“ auf Seite 21 beschrieben 	
Name der gestarteten Task des RSE-Dämons	<ul style="list-style-type: none"> • RSED • FEK.#CUST.PROCLIB(RSED), wie in „PROCLIB-Änderungen“ auf Seite 21 beschrieben 	
Name der gestarteten Task des Sperrdämons	<ul style="list-style-type: none"> • LOCKD • FEK.#CUST.PROCLIB(LOCKD), wie in „PROCLIB-Änderungen“ auf Seite 21 beschrieben 	
Anwendungs-ID	<ul style="list-style-type: none"> • FEKAPPL • /etc/rdz/rsed.envvars, wie in „Zusätzliche Java-Startparameter mit _RSE_JVAOPTS definieren“ auf Seite 42 beschrieben 	

Nachfolgend sind die Aktionen, die zur vollständigen Basissicherheitskonfiguration von Developer for System z erforderlich sind, übersichtlich aufgelistet. Um diese Anforderungen zu erfüllen, können je nach angestrebter Sicherheitsstufe verschiedene Methoden angewendet werden, wie in den unteren Abschnitten dokumentiert ist. Weitere Informationen zu den Sicherheitskonfigurationen von optionalen Developer for System z-Services enthalten die vorherigen Abschnitte.

- „Sicherheitseinstellungen und -klassen aktivieren“ auf Seite 180
- „OMVS-Segment für Benutzer von Developer for System z definieren“ auf Seite 181
- „Dateiprofile definieren“ auf Seite 181
- „Gestartete Tasks für Developer for System z definieren“ auf Seite 184
- „JES-Befehlssicherheit definieren“ auf Seite 185

- „RSE-Server als sicheren z/OS UNIX-Server definieren“ auf Seite 187
- „Programmgesteuerte MVS-Bibliotheken für RSE definieren“ auf Seite 187
- „Anwendungsschutz für RSE definieren“ auf Seite 188
- „PassTicket-Unterstützung für RSE definieren“ auf Seite 188
- „Programmgesteuerte z/OS UNIX-Dateien für RSE definieren“ auf Seite 190
- „Sicherheitseinstellungen prüfen“ auf Seite 190

Sicherheitseinstellungen und -klassen aktivieren

Developer for System z verwendet eine Vielzahl von Sicherheitsmechanismen, um für den Client eine geschützte und kontrollierte Hostumgebung bereitzustellen. Zu diesem Zweck müssen mehrere Klassen und Sicherheitseinstellungen aktiv sein. Vergleichen Sie hierzu die folgenden RACF-Beispielbefehle:

- Anzeige der aktuellen Einstellungen
 - SETROPTS LIST
- Aktivieren der Funktionsklasse für z/OS UNIX- und digitale Zertifikatsprofile
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- Aktivieren der Definitionen für gestartete Tasks
 - SETROPTS GENERIC(STARTED)
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- Aktivieren der Konsolsicherheit für JES Job Monitor
 - SETROPTS GENERIC(CONSOLE)
 - SETROPTS CLASSACT(CONSOLE) RACLIST(CONSOLE)
- Aktivieren des Bedienerbefehlsschutzes für JES Job Monitor
 - SETROPTS GENERIC(OPERCMDS)
 - SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)
- Aktivieren des Anwendungsschutzes für RSE
 - SETROPTS GENERIC(APPL)
 - SETROPTS CLASSACT(APPL) RACLIST(APPL)
- Aktivieren der geschützten Anmeldung unter Verwendung von PassTickets für RSE
 - SETROPTS GENERIC(PKTCDATA)
 - SETROPTS CLASSACT(PKTCDATA) RACLIST(PKTCDATA)
- Aktivieren der Programmsteuerung, um sicherzustellen, dass RSE nur gesicherten Code laden kann
 - RDEFINE PROGRAM ** ADDMEM('SYS1.CMDLIB'//NOPADCHK) UACC(READ)
 - SETROPTS WHEN(PROGRAM)

Anmerkung: Wenn die Klasse PROGRAM bereits ein Profil * enthält, sollten Sie das Profil ** nicht erstellen. Dadurch wird der von 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. IBM empfiehlt die Verwendung des Profils **, wie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683) beschrieben.

Achtung: Wenn "WHEN PROGRAM" aktiv ist, müssen einige Produkte (beispielsweise FTP) programmgesteuert sein. Testen Sie eine solche Definition, bevor Sie sie auf einem Produktionssystem aktivieren.

- (Optional) Unterstützung für X.509-HostIdMappings und erweiterten Eingangsport (POE) aktivieren
 - SETROPTS GENERIC(SERVAUTH)
 - SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)

OMVS-Segment für Benutzer von Developer for System z definieren

Für jeden Benutzer von Developer for System z muss ein RACF-OMVS-Segment (oder eine funktionale Entsprechung) definiert werden, das eine gültige z/OS UNIX-Benutzer-ID (UID, ungleich null) angibt. Darüber hinaus müssen für jeden Benutzer ein Ausgangsverzeichnis und ein Shellbefehl definiert werden. Für die Standardgruppe jedes Benutzers ist ebenfalls ein OMVS-Segment mit einer Gruppen-ID erforderlich.

Ersetzen Sie in den folgenden RACF-Beispielbefehlen die Platzhalter #userid, #user-identifizier, #group-name und #group-identifizier durch tatsächliche Werte:

- ALTUSER #userid
OMVS(UID(#user-identifizier) HOME(/u/#userid) PROGRAM(/bin/sh) NOASSIZEMAX)
- ALTGROUP #group-name OMVS(GID(#group-identifizier))

Sie können das im Profil BPX.DEFAULT.USER der Klasse FACILITY definierte, gemeinsam genutzte OMVS-Segment verwenden, um die OMVS-Segmentanforderungen von Developer for System z zu erfüllen. Dies wird jedoch nicht empfohlen.

Dateiprofile definieren

Für die meisten Dateien von Developer for System z reicht das Zugriffsrecht READ für Benutzer und ALTER für Systemprogrammierer aus. Ersetzen Sie den Platzhalter #sysprog durch gültige Benutzer-IDs oder RACF-Gruppennamen. Fragen Sie außerdem den Systemprogrammierer, der das Produkt installiert und konfiguriert hat, nach den korrekten Dateinamen. Der während der Installation verwendete Standard-High-Level-Qualifier ist FEK. Der Standard-High-Level-Qualifier für Dateien, die während des Anpassungsprozesses erstellt werden, ist FEK.#CUST.

- ADDGROUP (FEK) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
- ADDSD 'FEK.*.**' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- PERMIT 'FEK.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- SETROPTS GENERIC(DATASET) REFRESH

Anmerkung:

- Der Schutz von FEK.SFEKAUTH vor Aktualisierungen wird dringend angeraten, da diese Datei APF-Berechtigung hat. Dasselbe gilt für FEK.SFEKLOAD und FEK.SFEKLPA, hier jedoch, weil diese Dateien programmgesteuert sind.
- Bei den Beispielbefehlen in dieser Veröffentlichung und im Job FEKRACF wird vorausgesetzt, dass EGN (Enhanced Generic Naming) aktiv ist. Wenn EGN aktiv ist, kann der Qualifier ** eine beliebige Anzahl von Qualifiern in der Klasse DATASET repräsentieren. Ersetzen Sie ** durch *, wenn EGN auf Ihrem System nicht aktiv ist. Weitere Informationen zu EGN finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Einige der optionalen Komponenten von Developer for System z erfordern zusätzliche Sicherheitsdateiprofile. Ersetzen Sie die Platzhalter #sysprog, #ram-developer und #cicsadmin durch gültige Benutzer-IDs oder RACF-Gruppennamen:

- Wenn Sie die Umsetzung langer/kurzer Namen von SCLM Developer Toolkit verwenden, benötigen Benutzer das Zugriffsrecht UPDATE für die Zuordnungs-VSAM FEK.#CUST.LSTRANS.FILE.
 - ADDSD 'FEK.#CUST.LSTRANS.*.*' UACC(UPDATE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
 - PERMIT 'FEK.#CUST.LSTRANS.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - SETROPTS GENERIC(DATASET) REFRESH
- CARMA-RAM-Entwickler (Repository Access Manager) benötigen das Zugriffsrecht UPDATE für die CARMA-VSAMs (FEK.#CUST.CRA*.*).
 - ADDSD 'FEK.#CUST.CRA*.*' UACC(READ)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
 - PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
 - SETROPTS GENERIC(DATASET) REFRESH
- Wenn der CRD-Server (CICS Resource Definition) von Application Deployment Manager verwendet wird, ist für CICS-Administratoren das Zugriffsrecht UPDATE für die VSAM mit dem CRD-Repository erforderlich.
 - ADDSD 'FEK.#CUST.ADNREP*.*' UACC(READ)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
 - PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
 - SETROPTS GENERIC(DATASET) REFRESH
- Wenn das Manifestrepository von Application Deployment Manager definiert ist, ist für alle Benutzer von CICS Transaction Server das Zugriffsrecht 'UPDATE' für die VSAM mit dem Manifestrepository erforderlich.
 - ADDSD 'FEK.#CUST.ADNMAN*.*' UACC(UPDATE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
 - PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
 - SETROPTS GENERIC(DATASET) REFRESH

Verwenden Sie die folgenden RACF-Beispielbefehle für eine besser geschützte Konfiguration, bei der auch die Zugriffsberechtigung READ kontrolliert wird.

- Dateischutz UACC(NONE)
 - ADDGROUP (FEK)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
 - OWNER(IBMUSER) SUPGROUP(SYS1)"
 - ADDSD 'FEK.*.*' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
 - ADDSD 'FEK.SFEKAUTH' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
 - ADDSD 'FEK.SFEKLOAD' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
 - ADDSD 'FEK.SFEKPROC' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
 - ADDSD 'FEK.#CUST.PARMLIB' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
 - ADDSD 'FEK.#CUST.CNTL' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
 - ADDSD 'FEK.#CUST.LSTRANS.*.*' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
 - ADDSD 'FEK.#CUST.CRA*.*' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
 - ADDSD 'FEK.#CUST.ADNREP*.*' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
 - ADDSD 'FEK.#CUST.ADNMAN*.*' UACC(NONE)
 - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
- Berechtigung für den Systemprogrammierer zur Verwaltung aller Bibliotheken
 - PERMIT 'FEK.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)

- PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.LSTRANS.*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- Berechtigung für Clients zum Zugriff auf die Ladebibliotheken und Exec-Bibliotheken
 - PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(*)
 - PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(*)
 - PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(READ) ID(*)
 - PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(READ) ID(*)

Anmerkung: Für FEK.SFEKLPA sind keine Berechtigungen erforderlich, weil der im LPA befindliche Code für alle zugänglich ist.

- Berechtigung für JES Job Monitor zum Zugriff auf die Lade- und Parameterbibliotheken
 - PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
 - PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
- Berechtigung für Clients zum Aktualisieren der VSAM für die Umsetzung langer/kurzer Namen für SCLMDT (optional)
 - PERMIT 'FEK.#CUST.LSTRANS.*.*' CLASS(DATASET) ACCESS(UPDATE) ID(*)
- Berechtigung für RAM-Entwickler zur Aktualisierung der CARMA-VSAMs für CARMA (optional)
 - PERMIT 'FEK.#CUST.CRA*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
- Berechtigung für CICS-Benutzer, die VSAM mit dem CRD-Repository für Application Deployment Manager zu lesen (optional)
 - PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(READ) ID(*)
- Berechtigung für CICS-Administratoren, die VSAM mit dem CRD-Repository für Application Deployment Manager zu aktualisieren (optional)
 - PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
- Berechtigung für CICS-Benutzer, die VSAM mit dem Manifestrepository für Application Deployment Manager zu aktualisieren (optional)
 - PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(UPDATE) ID(*)
- Berechtigung für den CICS TS-Server zum Zugriff auf die Ladebibliothek für BIDI und Application Deployment Manager (optional)
 - PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
- Berechtigung für den DB2-Server zum Zugriff auf die Exec-Bibliothek für den DB2 Stored Procedure Builder (optional)
 - PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(READ) ID(#db2)
- Aktivieren von Sicherheitsprofilen
 - SETROPTS GENERIC(DATASET) REFRESH

Wenn Sie das Zugriffsrecht READ für Systemdateien kontrollieren möchten, müssen Sie Servern und Benutzern von Developer for System z die Berechtigung READ für folgende Dateien einräumen:

- CEE.SCEERUN
- CEE.SCEERUN2

- CBC.SCLBDLL
- ISP.SISPLoad
- ISP.SISPLPA
- SYS1.LINKLIB
- SYS1.SIEALNKE
- REXX.V1R4M0.SEAGLPA

Anmerkung: Wenn Sie die Alternativbibliothek für das REXX-Produktpaket verwenden, ist der Standardname der REXX-Laufzeitbibliothek REXX.*.SEAGALT anstelle von REXX.*.SEAGLPA, wie im Beispiel oben verwendet.

Gestartete Tasks für Developer for System z definieren

Die folgenden RACF-Beispielbefehle erstellen die gestarteten Tasks JMON, RSED und LOCKD mit der ihnen jeweils zugeordneten geschützten Benutzer-ID (STCJMON, STCRSE beziehungsweise STCLOCK) und der Gruppe STCGROUP. Ersetzen Sie die Platzhalter #group-id und #user-id-* durch gültige OMVS-IDs.

- ADDGROUP STCGROUP OMVS(GID(#group-id))
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
- ADDUSER STCJMON DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - JES JOBMONITOR')
OMVS(UID(#user-id-jmon) HOME(/tmp) PROGRAM(/bin/sh) NOASSIZEMAX
NOTHEADSMAX)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCRSE DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - RSE DAEMON')
OMVS(UID(#user-id-rse) HOME(/tmp) PROGRAM(/bin/sh) ASSIZEMAX(2147483647)
NOTHEADSMAX)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCLOCK DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - LOCK DAEMON')
OMVS(UID(#user-id-lock) HOME(/tmp) PROGRAM(/bin/sh) NOASSIZEMAX)
NOTHEADSMAX)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE STARTED JMON.* DATA('RDZ - JES JOBMONITOR')
STDATA(USER(STCJMON) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED RSED.* DATA('RDZ - RSE DAEMON')
STDATA(USER(STCRSE) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED LOCKD.* DATA('RDZ - LOCK DAEMON')
STDATA(USER(STCLOCK) GROUP(STCGROUP) TRUSTED(NO))
- SETROPTS RACLIST(STARTED) REFRESH

Anmerkungen:

1. Stellen Sie sicher, dass die Benutzer-IDs der gestarteten Tasks durch Angabe des Schlüsselworts NOPASSWORD geschützt sind.
2. Stellen Sie sicher, dass der RSE-Server eine eindeutige OMVS-Benutzer-ID besitzt, denn dieser Benutzer-ID werden Zugriffsrechte für z/OS UNIX gewährt.
3. Der RSE-Dämon benötigt für den ordnungsgemäßen Betrieb einen großen Adressraum (2 GB). Sie sollten diesen Wert in der Variable ASSIZEMAX des OMVS-Segments für die Benutzer-ID STCRSE festlegen. Dies stellt sicher, dass der RSE-Dämon unabhängig von Änderungen an MAXASSIZE in SYS1.PARMLIB(BPXPRMxx) die erforderliche Regionsgröße erhält.
4. Für den ordnungsgemäßen Betrieb von RSE ist außerdem eine große Anzahl von Threads erforderlich. Sie können einen Grenzwert in der Variable THREADSMAX des OMVS-Segments für die Benutzer-ID STCRSE festlegen. Dies stellt sicher, dass für RSE unabhängig von Änderungen an MAXTHREADS oder MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) der erforderliche Grenzwert für Threads festgelegt ist. Lesen Sie Kapitel 13, „Optimierungsaspekte“, auf Seite 215, um den korrekten Grenzwert für Threads zu ermitteln.

5. Für die Benutzer-ID STCJMON ist es ebenfalls sinnvoll, THREADSMAX im OMVS-Segment festzulegen, da JES Job Monitor für jede Clientverbindung einen Thread verwendet.

Sie sollten überlegen, ob für die Benutzer-ID STCRSE Einschränkungen definiert werden müssen. Benutzer mit dem Attribut RESTRICTED können nicht auf geschützte Ressourcen (MVS) zugreifen, solange sie nicht ausdrücklich für den Zugriff berechtigt wurden.

ALTUSER STCRSE RESTRICTED

Um sicherzustellen, dass eingeschränkte Benutzer nicht über die 'anderen' Zugriffsbits Zugriff auf z/OS UNIX-Dateisystemressourcen erlangen, müssen Sie das Profil RESTRICTED.FILESYS.ACCESS in der Klasse UNIXPRIV mit UACC(NONE) definieren. Weitere Informationen zur Einschränkung von Benutzer-IDs finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Achtung: Wenn Sie Einschränkungen für Benutzer-IDs festgelegt haben, müssen Sie die Berechtigung für den Zugriff auf eine Ressource explizit mit dem TSO-Befehl **PERMIT** oder dem z/OS UNIX-Befehl **setfacl** hinzufügen. Dies ist unter anderem für Ressourcen erforderlich, für die die Dokumentation von Developer for System z UACC verwendet (wie das Profil ** in der Klasse PROGRAM) oder für die allgemeine z/OS UNIX-Konventionen gelten (beispielsweise, dass jeder Lese- und Ausführungsberechtigung für Java-Bibliotheken besitzt). Testen Sie eine solche Definition, bevor Sie sie auf einem Produktionssystem aktivieren.

JES-Befehlssicherheit definieren

JES Job Monitor setzt alle von einem Benutzer angeforderten JES-Bedienerbefehle über eine erweiterte MCS-Konsole (EMCS) ab, deren Bezeichnung durch die Anweisung `CONSOLE_NAME` gesteuert wird, wie im Abschnitt „Konfigurationsdatei für JES Job Monitor (FEJJCNFG)“ auf Seite 29 dokumentiert.

Der folgende RACF-Beispielbefehl gewährt Benutzern von Developer for System z einen bedingten Zugriff auf eine eingeschränkte Gruppe von JES-Befehlen (Hold, Release, Cancel und Purge). Benutzer haben nur Ausführungsrechte, wenn sie die Befehle über JES Job Monitor absetzen. Ersetzen Sie den Platzhalter `#console` durch den aktuellen Konsolennamen.

- `RDEFINE OPERCMDS MVS.MCSOPER.#console UACC(READ)`
`DATA('RATIONAL DEVELOPER FOR SYSTEM Z')`
- `RDEFINE OPERCMDS JES%.** UACC(NONE)`
- `PERMIT JES%.** CLASS(OPERCMDS) ACCESS(UPDATE) WHEN(CONSOLE(JMON)) ID(*)`
- `SETOPTS RACLIST(OPERCMDS) REFRESH`

Anmerkung:

- Wenn kein Profil `MVS.MCSOPER.#console` definiert ist, kann die Konsole verwendet werden.
- Damit `WHEN(CONSOLE(JMON))` funktioniert, muss die Klasse `CONSOLE` aktiviert sein. In der Klasse `CONSOLE` für EMCS-Konsolen ist jedoch keine aktuelle Profilüberprüfung vorhanden.
- Ersetzen Sie nicht `JMON` mit dem aktuellen Konsolennamen in der Klausel `WHEN(CONSOLE(JMON))`. Das `JMON`-Schlüsselwort repräsentiert die Eingangsportanwendung und nicht den Konsolennamen.

Achtung: Wenn Sie in Ihrer Sicherheitssoftware die JES-Befehle mit dem universellen Zugriffsrecht NONE definieren, kann sich das negativ auf andere Anwendungen und Operationen auswirken. Testen Sie eine solche Definition, bevor Sie sie auf einem Produktionssystem aktivieren.

In Tabelle 27 und Tabelle 28 sehen Sie die Bedienerbefehle, die für JES2 und JES3 abgesetzt werden, sowie die eigenständigen Sicherheitsprofile zu deren Schutz.

Tabelle 27. Bedienerbefehle von JES2 Job Monitor

Aktion	Befehl	OPERCMDS-Profil	Erforderlicher Zugriff
Hold	\$Hx(jobid) x = {J, S oder T}	jesname.MODIFYHOLD.BAT jesname.MODIFYHOLD.STC jesname.MODIFYHOLD.TSU	UPDATE
Release	\$Ax(jobid) x = {J, S oder T}	jesname.MODIFYRELEASE.BAT jesname.MODIFYRELEASE.STC jesname.MODIFYRELEASE.TSU	UPDATE
Cancel	\$Cx(jobid) x = {J, S oder T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE
Purge	\$Cx(jobid),P x = {J, S oder T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE

Tabelle 28. Bedienerbefehle von JES3 Job Monitor

Aktion	Befehl	OPERCMDS-Profil	Erforderlicher Zugriff
Hold	*F,J=jobid,H	jesname.MODIFY.JOB	UPDATE
Release	*F,J=jobid,R	jesname.MODIFY.JOB	UPDATE
Cancel	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE
Purge	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE

Anmerkung:

- Die JES-Bedienerbefehle 'Hold', 'Release', 'Cancel' und 'Purge' können nur für Spooldateien abgesetzt werden, deren Eigner die Clientbenutzer-ID ist, es sei denn, in der Konfigurationsdatei von JES Job Monitor ist für LIMIT_COMMANDS= der Wert LIMITED oder NOLIMIT angegeben. Weitere Informationen hierzu enthält der Abschnitt „Aktionen für Beschränkungen der Jobziele“ auf Seite 166.
- Benutzer können jede Spooldatei anzeigen, sofern in der Konfigurationsdatei von JES Job Monitor nicht LIMIT_VIEW=USERID definiert ist. Weitere Informationen hierzu enthält der Abschnitt „Zugriff auf Spooldateien“ auf Seite 170.
- Benutzer, die nicht berechtigt sind, diese Bedienerbefehle auszuführen, können trotzdem mit JES Job Monitor Jobs übergeben und Jobausgaben lesen, sofern sie über eine ausreichende Berechtigung für eventuelle Profile verfügen, die diese Ressourcen schützen (z. B. diejenigen in den Klassen JESINPUT, JESJOBS und JESSPOOL).

Ihre Sicherheitssoftware verhindert, dass ein Benutzer in einer TSO-Sitzung eine Konsole JMON erstellt, weil er sich so als JES Job Monitor-Server ausgeben könnte. Auch wenn die Konsole erstellt werden kann, unterscheidet sich der Eingangsport

(JES Job Monitor oder TSO). Von dieser Konsole abgesetzte JES-Befehle werden jedoch nicht die Sicherheitsprüfung bestehen, wenn Ihre Sicherheitssoftware wie in dieser Veröffentlichung beschrieben konfiguriert ist und der Benutzer nicht autorisiert ist, JES-Befehle über andere Mechanismen zu verwenden.

RSE-Server als sicheren z/OS UNIX-Server definieren

RSE benötigt die Zugriffsberechtigung UPDATE für das Profil BPX.SERVER, um die Sicherheitsumgebung für den Client-Thread erstellen/löschen zu können. Wenn dieses Profil nicht definiert ist, muss für RSE UID(0) verwendet werden.

- RDEFINE FACILITY BPX.SERVER UACC(NONE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCRSE)
- 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 dies Auswirkungen auf andere z/OS UNIX-Anwendungen und -Operationen. Testen Sie eine solche Definition, bevor Sie sie auf einem Produktionssystem aktivieren. Weitere Informationen zu den verschiedenen Sicherheitsstufen finden Sie in der Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Programmgesteuerte MVS-Bibliotheken für RSE definieren

Server mit der Berechtigung für BPX.SERVER müssen in einer sauberen, programmgesteuerten Umgebung ausgeführt werden. Dies impliziert, dass alle von RSE aufgerufenen Programme ebenfalls programmgesteuert sein müssen. Die Programmsteuerung von MVS-Ladebibliotheken wird von Ihrer Sicherheitssoftware verwaltet.

RSE verwendet die Systembibliothek (SYS1.LINKLIB), die Bibliothek der Laufzeit von Language Environment (CEE.SCEERUN*) und die Ladebibliothek des TSO/ISPF-Client-Gateways von ISPF (ISP.SISPLOAD).

- RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('ISP.SISPLOAD'//NOPADCHK)
- SETROPTS WHEN(PROGRAM) REFRESH

Anmerkung: Wenn die Klasse PROGRAM bereits ein Profil * enthält, sollten Sie das Profil ** nicht verwenden. Dadurch 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. IBM empfiehlt die Verwendung des Profils "***". Informationen hierzu finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Zur Unterstützung optionaler Services müssen die folgenden (zusätzlich vorausgesetzten) Bibliotheken programmgesteuert sein. Diese Liste enthält keine Dateien, die für ein Produkt spezifisch sind, mit dem Developer for System z interagiert, beispielsweise IBM Debug Tool.

- Alternative REXX-Laufzeitbibliothek (für SCLM Developer Toolkit)
 - REXX.*.SEAGALT
- Systemladebibliothek (für SSL-Verschlüsselung)
 - SYS1.SIEALNKE
- Ladebibliothek des File Manager-Listeners (für die File Manager-Integration)
 - FMN.SFMNMODA

Anmerkung: Bibliotheken, die in den Link-Pack-Bereich (LPA) gestellt werden müssen, erfordern Programmsteuerberechtigungen, wenn für den Zugriff auf diese Bibliotheken LINKLIST oder STEPLIB verwendet wird. In dieser Veröffentlichung ist die Verwendung der folgenden LPA-Bibliotheken dokumentiert:

- ISPF (für das TSO/ISPF-Client-Gateway)
 - ISP.SISPLPA
- REXX-Laufzeitbibliothek (für SCLM Developer Toolkit)
 - REXX.*.SEAGLPA
- Developer for System z (für CARMA)
 - FEK.SFEKLPA

Anwendungsschutz für RSE definieren

Während der Clientanmeldung prüft der RSE-Dämon, ob ein Benutzer die Anwendung verwenden darf.

- RDEFINE APPL FEKAPPL UACC(READ) DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- SETROPTS RACLIST(APPL) REFRESH

Anmerkung:

RSE unterstützt die Verwendung von anderen Anwendungs-IDs als FEKAPPL. Ausführlichere Informationen dazu finden Sie im Abschnitt „PassTicket-Unterstützung für RSE definieren“. Die Klassendefinition APPL muss mit der eigentlichen, von RSE verwendeten Anwendungs-ID übereinstimmen.

Achtung: Die Clientverbindungsanforderung schlägt fehl, wenn das Anwendungsprofil nicht definiert ist oder wenn der Benutzer keinen Lesezugriff auf das Profil hat.

PassTicket-Unterstützung für RSE definieren

Das Kennwort des Clients (oder andere Identifikationsmittel, wie ein X.509-Zertifikat) wird nur verwendet, um die Identität des Clients beim Herstellen der Verbindung zu überprüfen. Danach wird die Threadsicherheit mit PassTickets verwaltet.

PassTickets sind vom System generierte Kennwörter mit einer Lebensdauer von ca. 10 Minuten. Die generierten PassTickets basieren auf einem geheimen Schlüssel. Dieser Schlüssel ist eine 64-Bit-Zahl (16 Hexadezimalzeichen). Ersetzen Sie in den folgenden RACF-Beispielbefehlen den Platzhalter key16 durch eine vom Benutzer angegebene Hexadezimalzeichenfolge mit 16 Zeichen (0-9 und A-F).

- RDEFINE PTKTDATA FEKAPPL UACC(NONE) SSIGNON(KEYMASKED(key16))
APPLDATA('NO REPLAY PROTECTION – DO NOT CHANGE')
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE PTKTDATA IRRPTAUTH.FEKAPPL.* UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

- PERMIT IRRPTAUTH.FEKAPPL.* CLASS(PTKTDATA) ACCESS(UPDATE) ID(STCRSE)
- SETROPTS RACLIST(PTKTDATA) REFRESH

RSE unterstützt die Verwendung von anderen Anwendungs-IDs als FEKAPPL. Entfernen Sie in `rsed.envvars` das Kommentarsymbol vor der Option "APPLID=FEKAPPL" und passen Sie diese an, um die Option zu aktivieren. Lesen Sie hierzu die Informationen unter „Zusätzliche Java-Startparameter mit `_RSE_JA-VAOPTS` definieren" auf Seite 42. Die Klassendefinition PTKTDATA muss mit der eigentlichen, von RSE verwendeten Anwendungs-ID übereinstimmen.

Sie sollten OMVSAPPL nicht als Anwendungs-ID verwenden, da diese ID den geheimen Schlüssel zu den meisten z/OS UNIX-Anwendungen entschlüsselt. Sie sollten ebenso nicht die standardmäßige MVS-Anwendungs-ID (MVS gefolgt von der SMF-ID des Systems) verwenden, da diese ID den geheimen Schlüssel zu den meisten MVS-Anwendungen (einschließlich Benutzer-Batch-Jobs) entschlüsselt.

Anmerkung:

- Wenn die Klasse PTKTDATA bereits definiert ist, überprüfen Sie, ob diese als eine generische Klasse definiert ist, bevor Sie die oben aufgeführten Profile erstellen. Ab z/OS Release 1.7 werden mit der Einführung der Java-Schnittstelle zu PassTickets generische Zeichen in der Klasse PTKTDATA unterstützt.
- Ersetzen Sie den Platzhalter (*) in der Definition IRRPTAUTH.FEKAPPL.* durch eine gültige Maske der Benutzer-ID, um die Benutzer-IDs einzuschränken, für die RSE ein PassTicket generieren kann.
- Abhängig von Ihren RACF-Einstellungen steht der Benutzer, der ein Profil definiert, möglicherweise auch auf der Zugriffsliste des Profils. Es wird empfohlen, dass Sie diese Berechtigungen für die Profile PTKTDATA entfernen.
- Damit JES Job Monitor die vom RSE angegebenen PassTickets überprüfen kann, müssen JES Job Monitor und RSE dieselbe Anwendungs-ID besitzen.
- Wenn Sie auf Ihrem System ein Verschlüsselungsprodukt installiert haben und dieses verfügbar ist, kann der Anwendungsschlüssel zur sicheren Anmeldung für einen zusätzlichen Schutz verschlüsselt werden. Verwenden Sie dazu das Schlüsselwort KEYENCRYPTED anstelle von KEYMASKED. Weitere Informationen hierzu finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

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

Programmgesteuerte z/OS UNIX-Dateien für RSE definieren

Server mit der Berechtigung für BPX.SERVER müssen in einer sauberen, programm-gesteuerten Umgebung ausgeführt werden. Dies impliziert, dass alle von RSE auf-gerufenen Programme ebenfalls programmgesteuert sein müssen. Die Programm-steuerung für z/OS UNIX-Dateien wird mit dem Befehl **extattr** verwaltet. Für die Ausführung dieses Befehls benötigen Sie die Zugriffsberechtigung READ für BPX.FILEATTR.PROGCTL in der Klasse FACILITY oder die UID(0).

Der RSE-Server verwendet die gemeinsam genutzte Java-Bibliothek von RACF (/usr/lib/libIRRRacf.so).

- `extattr +p /usr/lib/libIRRRacf.so`

Anmerkung:

- Ab z/OS 1.9 wird /usr/lib/libIRRRacf.so während der SMP/E-Installation von RACF als programmgesteuerte Datei installiert.
- Ab z/OS 1.10 ist /usr/lib/libIRRRacf.so Teil der System Authori-zation Facility (SAF), die zum Lieferumfang des Basisprodukts z/OS gehört. Damit ist die JAR-Datei auch für Kunden verfügbar, die kein RACF verwenden.
- Wenn Sie ein anderes Sicherheitsprodukt als RACF verwenden, kann eine andere Konfiguration erforderlich sein. Ziehen Sie bei Fragen die Dokumentation zu Ihrem Sicherheitsprodukt zu Rate.
- Bei der SMP/E-Installation von Developer for System z wird das Programmsteuerungsbit für interne RSE-Programme gesetzt.
- Verwenden Sie zum Anzeigen des aktuellen Status des Programm-steuerungsbits den z/OS UNIX-Befehl **ls -Eog** (die Datei ist pro-grammgesteuert, wenn der Buchstabe **p** in der zweiten Zeichenfol-ge angezeigt wird).

```
$ ls -Eog /usr/lib/libIRRRacf.so
-rwxr-xr-x  aps-  2    69632 Oct  5  2007 /usr/lib/libIRRRacf.so
```

Sicherheitseinstellungen prüfen

Verwenden Sie die folgenden Beispielbefehle, um die Ergebnisse Ihrer Anpassun-gen in Bezug auf die Sicherheit anzuzeigen.

- Sicherheitseinstellungen und -klassen
 - `SETOPTS LIST`
- OMVS-Segment für Benutzer
 - `LISTUSER #userid NORACF OMVS`
 - `LISTGRP #group-name NORACF OMVS`
- Dateiprofile
 - `LISTGRP FEK`
 - `LISTDSD PREFIX(FEK) ALL`
- Gestartete Tasks
 - `LISTGRP STCGROUP OMVS`
 - `LISTUSER STCJMON OMVS`
 - `LISTUSER STCRSE OMVS`
 - `LISTUSER STCLOCK OMVS`
 - `RLIST STARTED JMON.* ALL STDATA`
 - `RLIST STARTED RSED.* ALL STDATA`

- RLIST STARTED LOCKD.* ALL STDATA
- JES-Befehlssicherheit
 - RLIST CONSOLE JMON ALL
 - RLIST OPERCMDS MVS.MCSOPER.JMON ALL
 - RLIST OPERCMDS JES%.** ALL
- RSE als sicherer z/OS UNIX-Server
 - RLIST FACILITY BPX.SERVER ALL
- Programmgesteuerte MVS-Bibliotheken für RSE
 - RLIST PROGRAM ** ALL
- Anwendungsschutz für RSE
 - RLIST APPL FEKAPPL ALL
- PassTicket-Unterstützung für RSE
 - RLIST PTKTDATA FEKAPPL ALL SSIGNON
 - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.* ALL
- Programmgesteuerte z/OS UNIX-Dateien für RSE
 - ls -E /usr/lib/libIRRracf.so

Kapitel 11. Wissenswertes zu Developer for System z

Der Host von Developer for System z umfasst einige interagierende Komponenten, damit der Client auf die Host-Services und -daten zugreifen kann. Wenn Sie das Design dieser Komponenten verstehen, können Sie die richtigen Konfigurationsentscheidungen treffen.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Komponentenübersicht“
- „RSE als Java-Anwendung“ auf Seite 195
- „Taskeigner“ auf Seite 196
- „Verbindungsflow“ auf Seite 198
- „Sperrdämon“ auf Seite 200
- „z/OS UNIX-Verzeichnisstruktur“ auf Seite 202

Komponentenübersicht

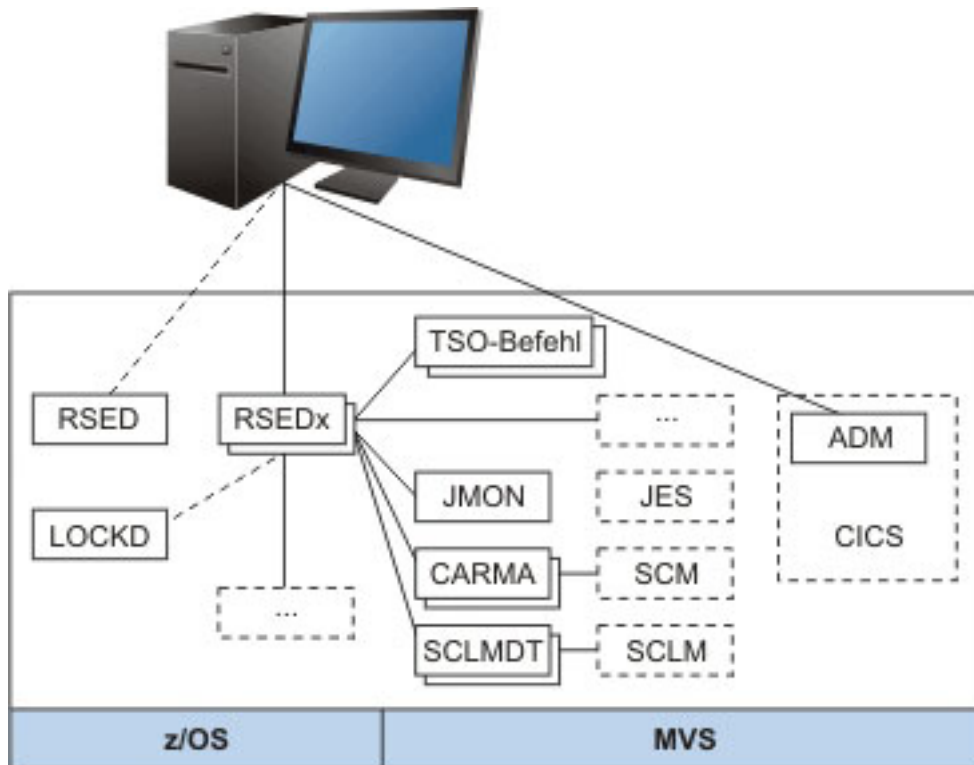


Abbildung 41. Komponentenübersicht

Abb. 41 zeigt eine allgemeine Übersicht des Layouts von Developer for System z auf Ihrem Hostsystem.

- Remote Systems Explorer (RSE) stellt Kernservices wie den Verbindungsaufbau vom Client zum Host und das Starten anderer Server für bestimmte Services bereit. RSE umfasst zwei logische Einheiten:

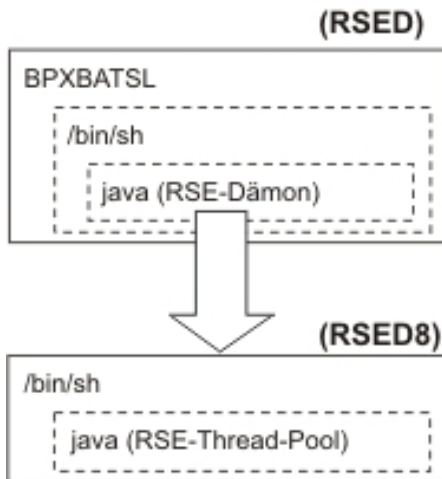
- RSE-Dämon (RSED), der den Verbindungsaufbau verwaltet. Der RSE-Dämon ist auch für die Ausführung im Einzelservermodus verantwortlich. Um dies zu erreichen, erstellt der RSE-Dämon mindestens einen untergeordneten Prozess, auch als RSE-Thread-Pool(s) (RSEDx) bekannt.
- RSE-Server für die einzelnen Clientanforderungen. Ein RSE-Server ist innerhalb eines RSE-Thread-Pools als Thread aktiv.
- Der Sperrdämon (LOCKD) stellt Überwachungsservices für Dateisperren bereit.
- TSO Commands Service (TSO cmd) stellt eine batchähnliche Schnittstelle für TSO- und ISPF-Befehle bereit.
- JMON (JES Job Monitor) stellt alle Services mit Bezug zum JES bereit.
- Common Access Repository Manager (CARMA) bietet eine Schnittstelle für die Interaktion mit Software Configuration Managers (SCMs), beispielsweise CA Endevor.
- SCLM Developer Toolkit (SCLMDT) stellt eine Schnittstelle zur Verfügung, um SCLM zu erweitern und mit SCLM zu interagieren.
- Application Deployment Manager (ADM) stellt verschiedene CICS-bezogene Services bereit.
- Weitere verfügbare Services können von Developer for System z selbst oder von zusätzlich erforderlichen Softwareprogrammen bereitgestellt werden.

Die Beschreibung im vorherigen Abschnitt und in der Liste verdeutlichen die zentrale Rolle von RSE. Mit ein paar wenigen Ausnahmen läuft jede Clientkommunikation über RSE ab. Dies ermöglicht eine sicherheitsbezogene Netzkonfiguration, da nur eine eingeschränkte Menge an Ports für die Kommunikation zwischen Client und Host verwendet wird.

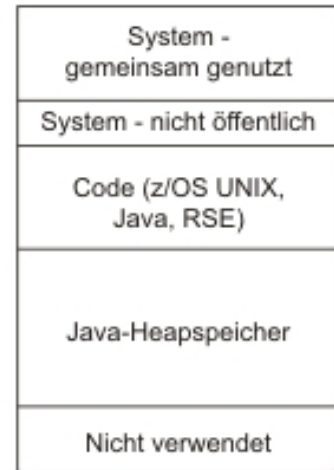
RSE besteht aus einem Dämonadressbereich, der Thread-Pooling und Adressräume steuert, um die Verbindungen und die Arbeitslast der Clients zu verwalten. Der Dämon wird als Sammelpunkt für Verbindungs- und Verwaltungszwecke eingesetzt, während die Thread-Pools die Clientarbeitslast verarbeiten. Auf Basis der in der Konfigurationsdatei `rsed.envvars` definierten Werte und der Summe aller Clientverbindungen können mehrere Adressräume von Thread-Pools durch den Dämon gestartet werden.

RSE als Java-Anwendung

z/OS UNIX-Prozesse



Java-Speicherbelegung



JOBNAME	Status	PID	PPID	Befehl
RSED	FILE SYS KERNEL WAIT	50331904	1	BPXBATSL
RSED	WAITING FOR CHILD	67109114	50331904	/bin/sh...
RSED	FILE SYS KERNEL WAIT	50331949	67109114	java...
RSED8	WAITING FOR CHILD	307	50331949	/bin/sh...
RSED8	FILE SYS KERNAL WAIT	308	307	java...

Abbildung 42. RSE als Java-Anwendung

Abb. 42 zeigt eine grundlegende Sicht auf die Ressourcennutzung (Prozesse und Speicher) von RSE.

RSE ist eine Java-Anwendung, das heißt, sie ist in der z/OS UNIX-Umgebung aktiv. Dies ermöglicht eine einfache Portierung auf verschiedene Hostplattformen und direkte Kommunikation mit dem Client von Developer for System z, der ebenfalls eine Java-Anwendung ist (auf dem Eclipse-Framework basierend). Deshalb ist grundlegendes Wissen zur Arbeitsweise von z/OS UNIX und Java sehr hilfreich, um Developer for System z zu verstehen.

In z/OS UNIX wird ein Programm in einem Prozess ausgeführt, der mithilfe einer Prozess-ID (PID) identifiziert wird. Da jedes Programm in seinem eigenen Prozess aktiv ist, wird beim Aufrufen eines anderen Programms ein neuer Prozess erstellt. Auf den Prozess, der für das Starten eines anderen Prozesses verantwortlich ist, wird mithilfe einer übergeordneten Prozess-ID (Parent PID, PPID) verwiesen. Der neue Prozess wird als untergeordneter Prozess bezeichnet. Der untergeordnete Prozess kann in demselben Adressraum ausgeführt oder in einem neuen Adressraum gestartet (erstellt) werden. Ein neuer Prozess, der in demselben Adressraum ausgeführt wird, kann mit der Ausführung eines Befehls in TSO verglichen werden. Der in einem neuen Adressraum gestartete Prozess ähnelt dem Übergeben eines Batch-Jobs.

Beachten Sie, dass ein Prozess ein Einzelthread- oder ein Multithreadprozess sein kann. In einer Multithreadanwendung (wie RSE) konkurrieren die verschiedenen Threads um die Netzressourcen, als wären sie separate Adressräume (mit weniger Aufwand).

Wenn diese Prozessinformationen dem RSE-Beispiel in Abb. 42 auf Seite 195 zugeordnet werden, ergibt sich der folgende Flow:

1. Beim Starten der RSED-Task wird 'BPXBATSL' ausgeführt. Dies ruft z/OS UNIX auf und erstellt eine Shellumgebung – PID 50331904.
2. In diesem Prozess wird das Shell-Script `rsed.sh` in einem separaten Prozess (`/bin/sh`) ausgeführt – PID 67109114.
3. Das Shell-Script legt die in `rsed.envvars` definierten Umgebungsvariablen fest und führt Java mit den erforderlichen Parametern aus, um den RSE-Dämon zu starten – PID 50331949.
4. Der RSE-Dämon wird in einer neuen Shell in einem untergeordneten Prozess (RSED8) gestartet – PID 307.
5. In dieser Shell werden die in `rsed.envvars` definierten Umgebungsvariablen festgelegt und Java wird mit den erforderlichen Parametern ausgeführt, um den RSE-Thread-Pool zu starten – PID 308.

Java-Anwendungen, wie RSE, ordnen Speicher nicht direkt zu, sondern mithilfe von Java-Speicherverwaltungsservices. Diese Services umfassen Funktionen wie das Zuordnen und Freigeben von Speicher sowie eine Garbage-Collection und werden innerhalb der Grenzwerte des Java-Heapspeichers ausgeführt. Die minimale und maximale Größe des Heapspeichers wird während des Systemstarts von Java (implizit oder explizit) definiert.

Um eine optimale Nutzung der verfügbaren Adressraumgröße zu erreichen, sollte die Größe des Heapspeichers umfangreich sein, um z/OS ausreichend Platz für die Speicherung einer variablen Menge von Systemsteuerungsblöcken (abhängig von der Anzahl aktiver Threads) zu lassen.

Taskeigner

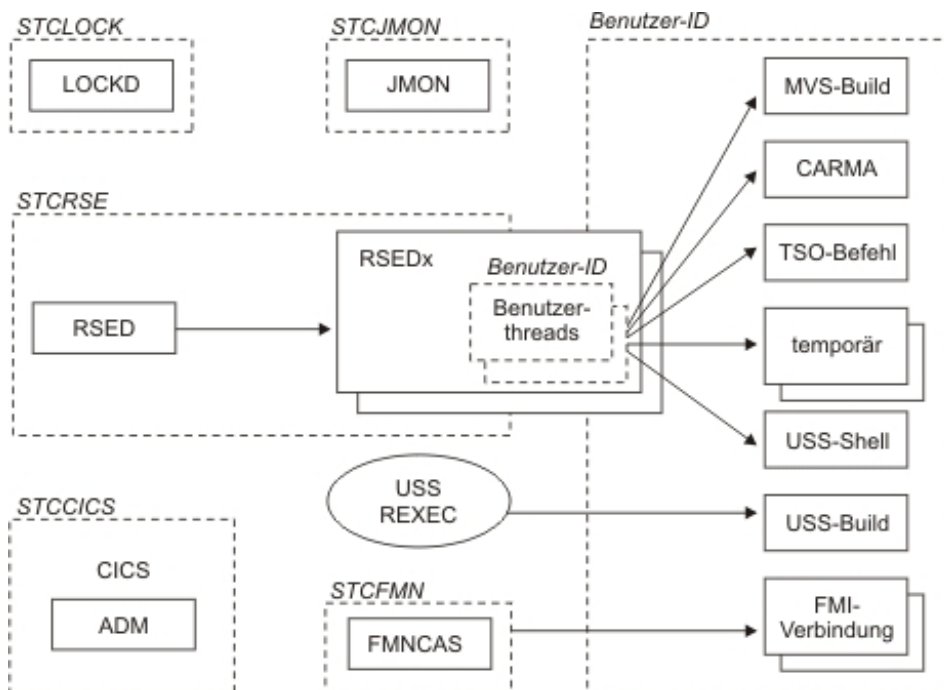


Abbildung 43. Taskeigner

Abb. 43 auf Seite 196 zeigt eine Basisübersicht über die Eigner der Sicherheitsberechtigungs-nachweise, die für verschiedene Tasks in Developer for System z verwendet werden.

Das Eigentumsrecht an einer Task kann in zwei Abschnitte unterteilt werden. Gestartete Tasks gehören der Benutzer-ID, die der gestarteten Task in Ihrer Sicherheitssoftware zugewiesen wird. Alle anderen Tasks, mit Ausnahme der RSE-Thread-Pools (RSEDx), gehören der Client-Benutzer-ID.

Abb. 43 auf Seite 196 zeigt gestartete Tasks in Developer for System z (LOCKD, JMON und RSED) sowie gestartete Beispieltasks und Beispielsystemservices, mit denen Developer for System z kommuniziert. Application Deployment Manager (ADM) ist innerhalb einer CICS-Region aktiv. FMNCAS ist die gestartete Task von File Manager. Das USS REXEC-Tag stellt den z/OS UNIX-REXEC-Service (oder SSH-Service) dar.

Der RSE-Dämon erstellt für die Verarbeitung von Prozessclientanforderungen mindestens einen Adressraum der RSE-Thread-Pools (RSEDx). Jeder RSE-Thread-Pool unterstützt mehrere Clients und gehört demselben Benutzer wie der RSE-Dämon. Jeder Client verfügt über eigene Threads innerhalb eines Thread-Pools. Diese Threads gehören der Client-Benutzer-ID.

Abhängig von den vom Client ausgeführten Aktionen können für die Ausführung der angeforderten Aktion zusätzliche Adressräume gestartet werden. Diese gehören alle der Client-Benutzer-ID. Diese Adressräume können ein MVS-Batch-Job, eine APPC-Transaktion oder ein untergeordneter z/OS UNIX-Prozess sein. Beachten Sie, dass ein untergeordneter z/OS UNIX-Prozess in einem z/OS UNIX-Initiator (BPXAS) aktiv ist und als gestartete Task in JES angezeigt wird.

Die Erstellung dieser Adressräume wird in den meisten Fällen von einem Benutzerthread in einem Thread-Pool entweder direkt oder mithilfe von Systemservices wie ISPF ausgelöst. Der Adressraum kann aber auch von einem Fremdanbieter erstellt werden. File Manager startet beispielsweise einen neuen Adressraum für jede Datei (oder jedes Member), die (das) es für Developer for System z verarbeitet. Der z/OS UNIX-REXEC-Service oder der SSH-Service sind beim Starten von Builds in z/OS UNIX beteiligt.

Die benutzerspezifischen Adressräume werden bei Abschluss der Tasks oder bei Ablauf eines Inaktivitätszeitgebers beendet. Die gestarteten Tasks bleiben aktiv. Die in Abb. 43 auf Seite 196 aufgeführten Adressräume bleiben für einen längeren Zeitraum im System sichtbar. z/OS UNIX wurde jedoch so entwickelt, dass es auch kurz andauernde, temporäre Adressräume gibt.

Verbindungsflow

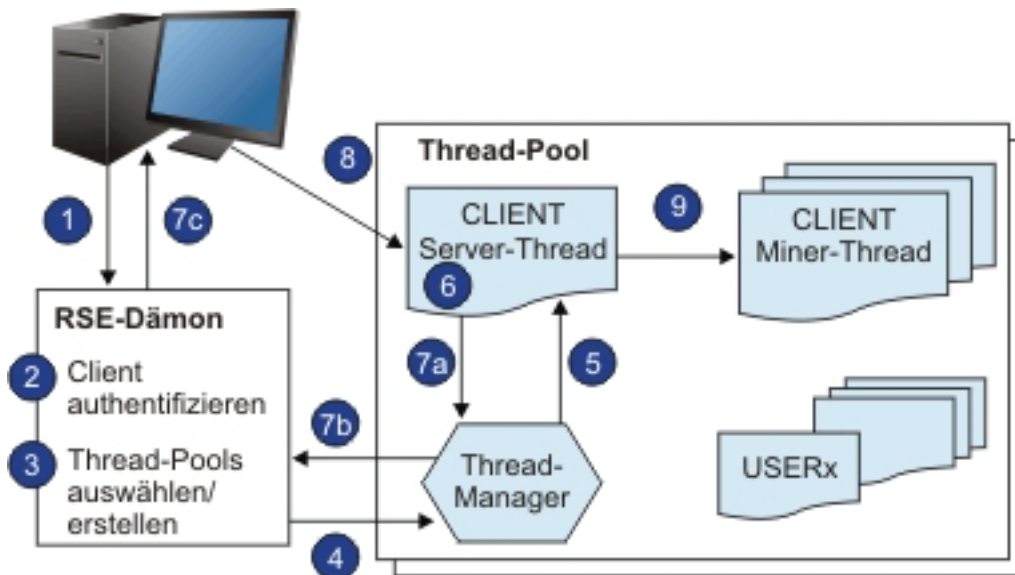


Abbildung 44. Verbindungsflow

Abb. 44 zeigt eine schematische Übersicht des Verbindungsaufbaus eines Clients mit einem Host mithilfe von Developer for System z. Es wird außerdem eine kurze Erklärung zur Verwendung von PassTickets bereitgestellt.

1. Der Client meldet sich bei dem Dämon an (Port 4035).
2. Der RSE-Dämon authentifiziert den Client anhand der vom Client angegebenen Berechtigungsnachweise.
3. Der RSE-Dämon wählt einen vorhandenen Thread-Pool aus oder startet einen neuen Thread-Pool, wenn alle anderen voll sind.
4. Der RSE-Dämon übergibt dem Thread-Pool die Benutzer-ID des Clients.
5. Der Thread-Pool erstellt mithilfe der Client-Benutzer-ID und einem PassTicket für die Authentifizierung einen clientspezifischen RSE-Server-Thread.
6. Der Client-Server-Thread bindet für die zukünftige Clientkommunikation an einen Port.
7. Der Client-Server-Thread gibt die Portnummer für den Client zurück, mit dem eine Verbindung hergestellt werden soll.
8. Der Client trennt die Verbindung mit dem RSE-Dämon und stellt eine Verbindung mit der angegebenen Portnummer her.
9. Der Client-Server-Thread startet mithilfe der Client-Benutzer-ID und einem PassTicket für die Authentifizierung andere benutzerspezifische Threads (Miners). Diese Threads stellen die vom Client angeforderten benutzerspezifischen Services bereit.

Die obige Beschreibung zeigt das threadorientierte Design von RSE. Anstelle des Startens eines Adressraums für jeden Benutzer nutzen mehrere Benutzer einen Adressraum mit Einzel-Thread-Pool. Innerhalb des Thread-Pools ist jeder Miner (benutzerspezifischer Service) in seinem eigenen Thread aktiv, dem der Sicherheitskontext des Benutzers zugeordnet ist, um eine sichere Konfiguration zu gewährleisten. Das Design ist für eine große Anzahl von Benutzern mit eingeschränkter Ressourcennutzung bestimmt, deren Clients jedoch jeweils mehrere Threads verwenden (abhängig von den ausgeführten Tasks sind es 16 oder mehr).

Aus der Netzperspektive arbeitet Developer for System z ähnlich wie ein FTP im passiven Modus. Der Client stellt eine Verbindung mit einem Sammelpunkt (RSE-Dämon) her, trennt die Verbindung anschließend und stellt erneut eine Verbindung mit einer vom Sammelpunkt angegebenen Portnummer her. Die folgende Logik steuert die Auswahl des Ports, der für die zweite Verbindung verwendet wird:

1. Wenn der Client eine Portnummer (ungleich null) in der Registerkarte für die Subsystemeigenschaften angegeben hat, bindet der RSE-Server an diesen Port. Wenn dieser Port nicht verfügbar ist, schlägt die Verbindung fehl.
2. Wenn `_RSE_PORTRANGE` in `rsed.envvars` angegeben ist, bindet der RS-Server an einen Port aus diesem Bereich. Steht kein Port zur Verfügung, schlägt die Verbindung fehl. Beachten Sie, dass der RSE-Server den Port nicht exklusiv für die Dauer der Clientverbindung benötigt. Es kann sich nur während der Serververbindung an den Port und des Verbindungsaufbaus des Clients kein anderer RSE-Server an den Port binden.
3. Wenn keine Einschränkungen festgelegt sind, bindet der RSE-Server an Port 0. Das hat zur Folge, dass TCP/IP die Portnummer auswählt.

Die Verwendung von PassTickets für alle z/OS-Services, die eine Authentifizierung erfordern, ermöglicht Developer for System z das beliebige Aufrufen dieser Services, ohne ein Kennwort speichern oder den Benutzer fortwährend danach fragen zu müssen. Die Verwendung von PassTickets für alle z/OS-Services macht außerdem ein alternatives Authentifizierungsverfahren während der Anmeldung möglich, beispielsweise durch Kennwörter für einmalige Anmeldungen und X.509-Zertifikate.

Sperrdämon

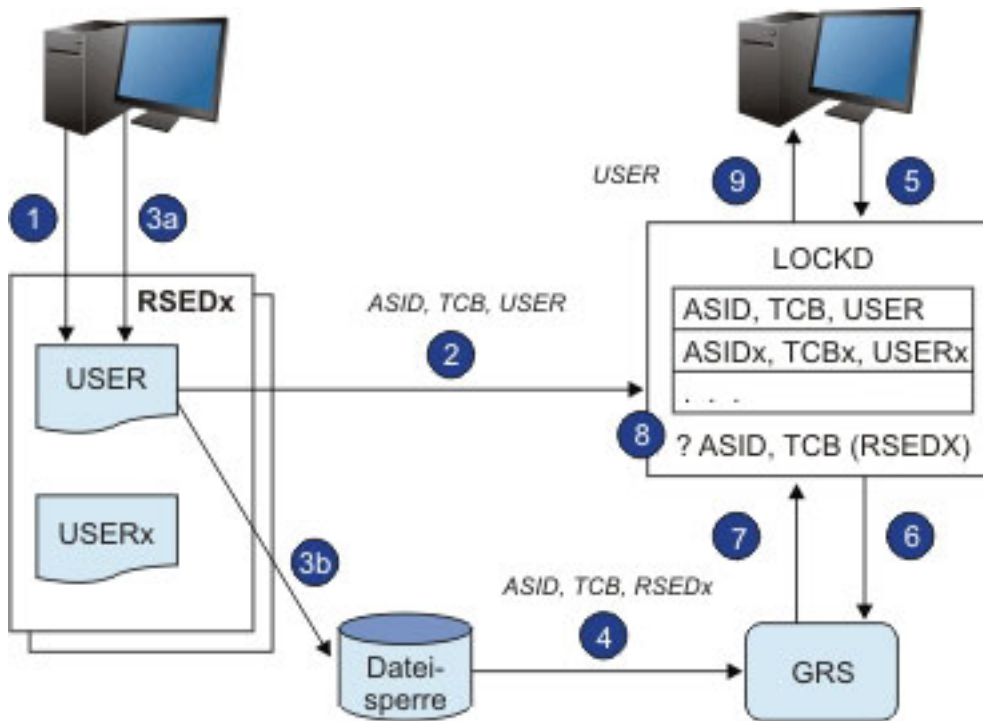


Abbildung 45. Sperrdämonflow

Abb. 45 zeigt eine schematische Übersicht darüber, wie der Sperrdämon festlegt, welcher Developer for System z-Client eine Dateisperre besitzt.

1. Der Client meldet sich an. Dadurch wird ein benutzerspezifischer RSE-Server-Thread (USER) innerhalb eines Thread-Pools (RSEDx) erstellt.
2. Der RSE-Server registriert einen neu verbundenen Benutzer mit dem Sperrdämon. Die Registrierungsinformationen enthalten die Adressraumkennung (ASID des Thread-Pools), die benutzerspezifische ID des Tasksteuerblocks (TCB) und die Benutzer-ID.
3. Der Client öffnet eine Datei zur Bearbeitung. Dies weist den RSE-Server an, eine exklusive Sperre für die Datei festzulegen.
4. Das System registriert die ASID, den TCB und den Tasknamen (RSEDx) des anfordernden Benutzers als Teil des Sperrprozesses. Diese Informationen werden in den GRS-Warteschlangen (Global Resource Serialization) gespeichert.
5. Ein Operator (bzw. der RSE-Server für einen Client) fragt die Sperrstufe der Datei bei dem Sperrdämon ab.
6. Der Sperrdämon durchsucht die GRS-Warteschlangen nach der Information, ob die Datei gesperrt ist.
7. Der Dämon empfängt die ASID, den TCB und den Tasknamen des Sperrereigentümers.
8. Die empfangene ASID und der empfangene TCB werden mit den ASID-TCB-Kombinationen der registrierten Clients verglichen.
9. Wenn eine Übereinstimmung gefunden wird, wird die zugehörige Client-Benutzer-ID an den anfordernden Benutzer zurückgegeben. Ist dies nicht der Fall, wird der von der GRS-Warteschlange empfangene Taskname zurückgegeben.

Innerhalb der EinzelsERVERkonfiguration von Developer for System z, bei der mehrere Benutzer einem Adressraum mit Einzel-Thread-Pool zugeordnet werden, verliert z/OS die Möglichkeit, zu verfolgen, wer die Sperre einer Datei oder eines Members besitzt. Systembefehle stoppen auf der Adressraumebene, die dem RSE-Thread-Pool entspricht.

Um dieses Problem zu umgehen, stellt Developer for System z den Sperrdämon zur Verfügung. Der Sperrdämon verfolgt alle Sperrungen für Dateien oder Member durch RSE-Benutzer sowie alle Sperrungen durch andere Produkte, beispielsweise ISPF.

Der RSE-Server registriert einen neu verbundenen Benutzer mit dem Sperrdämon. Die Registrierungsinformationen enthalten die Adressraumkennung (ASID des Thread-Pools), die benutzerspezifische ID des Tasksteuerblocks (TCB) und die Benutzer-ID.

Beachten Sie, dass die Registrierung nur während des Verbindungsaufbaus erfolgt, das heißt, dass keine RSE-Benutzer registriert werden, die bereits vor dem Start (oder Neustart) des Sperrdämons aktiv waren.

Wenn der Sperrdämon die Anfrage einer Datei empfängt (entweder mithilfe eines Abfragenoperatorbefehls oder vom Client über den RSE-Server), durchsucht der Dämon die GRS-Warteschlangen (Global Resource Serialization) des Systems. Wenn die ASID und der TCB mit der entsprechenden Kombination des registrierten Benutzers übereinstimmen, wird die Benutzer-ID als Sperreneigentümer zurückgegeben. Ist dies nicht der Fall, wird der Jobname bzw. die Benutzer-ID, der/die mit der ASID verknüpft ist, als Sperreneigentümer zurückgegeben.

Wenn die Registrierung fehlschlägt, wird eine Konsolnachricht (FEK513W) mit den Registrierungsinformationen angezeigt. Dies gibt einem Operator die Möglichkeit, die Werte mit der Ausgabe eines **DISPLAY GRS,RES=(*,dataset*)**-Operatorbefehls zu vergleichen, um den Sperreneigentümer zu finden.

Anmerkung: Erfolgreiche Registrierungen werden in 'DD STDOUT' des Servers aufgelistet, sofern log_level auf 2 festgelegt wurde. Dies ist für das manuelle Zuordnen bei erfolgreichen Registrierungen hilfreich, die nach dem Neustart des Sperrdämons entfernt wurden.

Sperren aufheben

Normalerweise wird eine Datei oder ein Member gesperrt, sobald sie/es im Editiermodus geöffnet wird, und die Sperre aufgehoben, wenn der Client die Editiersitzung beendet.

Bestimmte Fehlerbedingungen können den ordnungsgemäßen Ablauf dieses Mechanismus beeinträchtigen. In diesem Fall kann der Benutzer, der Eigentümer der Sperren ist, die Sperren mithilfe des RSE-Operatorbefehls **modify cancel** löschen, wie in Kapitel 8, „Bedienerbefehle“, auf Seite 125 beschrieben. Während dieses Prozesses werden alle aktiven Sperren von Dateien aufgehoben, die mit diesem Benutzer verknüpft sind.

z/OS UNIX-Verzeichnisstruktur

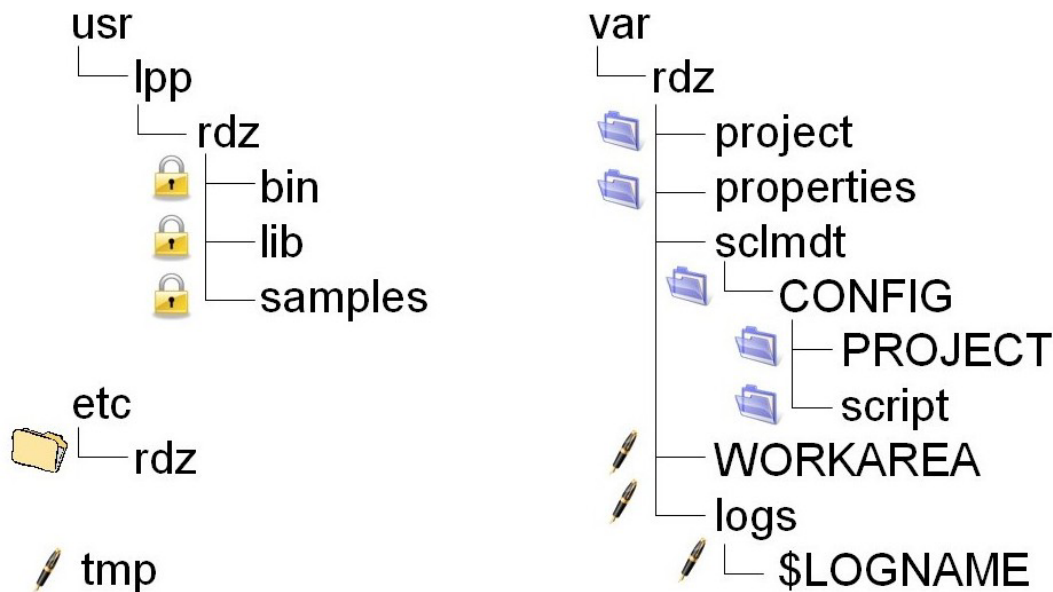


Abbildung 46. z/OS UNIX-Verzeichnisstruktur

Abb. 46 zeigt eine Übersicht über die von Developer for System z verwendeten z/OS UNIX-Verzeichnisse. Die folgende Liste enthält Informationen zu jedem von Developer for System z verwendeten Verzeichnis, darüber, wie die Position geändert werden kann und wer die darin enthaltenen Daten verwaltet.

- `/usr/lpp/rdz/` ist der Stammverzeichnispfad für den Produktcode von Developer for System z. Die eigentliche Position ist in den gestarteten Tasks RSED und LOCKD angegeben (Variable HOME). Die enthaltenen Dateien werden von SMP/E verwaltet.
- `/etc/rdz/` speichert die RSE- und Miner-bezogenen Konfigurationsdateien. Die eigentliche Position ist in den gestarteten Tasks RSED und LOCKD angegeben (Variable CNFG). Die enthaltenen Dateien werden vom Systemprogrammierer verwaltet.
- `/var/rdz/scldmt/CONFIG/` speichert allgemeine SCLMDT-Konfigurationsdateien. Die eigentliche Position ist in `rsed.envvars` angegeben (Variable SCLMDT_CONF_HOME). Die enthaltenen Dateien werden vom SCLM-Administrator verwaltet.
- `/var/rdz/scldmt/CONFIG/PROJECT/` speichert SCLMDT-Projektkonfigurationsdateien. Die eigentliche Position ist in `rsed.envvars` angegeben (Variable SCLMDT_CONF_HOME). Die enthaltenen Dateien werden vom SCLM-Administrator verwaltet.
- `/var/rdz/scldmt/CONFIG/script/` speichert SCLMDT-bezogene Scripts, die von anderen Produkten verwendet werden können. Die eigentliche Position ist nirgendwo angegeben. Die enthaltenen Dateien werden vom SCLM-Administrator verwaltet.
- `/var/rdz/projects/` speichert die hostbasierten Projektdefinitionsdateien. Die eigentliche Position ist in `projectcfg.properties` angegeben (Variable PROJECT_HOME). Die enthaltenen Dateien werden von einem Projektleiter oder einem leitenden Entwickler verwaltet.

- `/var/rdz/properties/` sperrt die hostbasierten Eigenschaftsgruppen. Die eigentliche Position ist in `propertiescfg.properties` angegeben (Variablen `PROPERTY-GROUP` und `DEFAULT-VALUES`). Die enthaltenen Dateien werden von einem Projektleiter oder einem leitenden Entwickler verwaltet.
- `/var/rdz/logs/` sperrt die Protokolle des RSE-Dämons und des RSE-Thread-Pool-Servers. Die eigentliche Position ist in `rsed.envvars` angegeben (Variable `daemon.log`). Die enthaltenen Dateien werden von RSE verwaltet.
- `/var/rdz/logs/$LOGNAME/` sperrt die benutzerspezifischen Protokolle des RSE-Servers und -Miners. Die eigentliche Position ist in `rsed.envvars` angegeben (Variable `user.log` und `DSTORE_LOG_DIRECTORY`). Die enthaltenen Dateien werden von RSE und den Miners verwaltet.

Anmerkung: `/var/rdz/logs/` erfordert die Berechtigungsbitmaske '777', um jedem Client die Erstellung eines `$LOGNAME`-Verzeichnisses und das Speichern von benutzerspezifischen Protokolldateien zu ermöglichen.

- `/var/rdz/WORKAREA/` wird vom TSO/ISPF-Client-Gateway von ISPF und SCLMDT verwendet, um Daten zwischen z/OS UNIX und MVS-basierten Adressräumen zu übertragen. Die eigentliche Position ist in `rsed.envvars` angegeben (Variable `_CMDSERV_WORK_HOME`). Die enthaltenen Dateien werden von ISPF und SCLMDT verwaltet.

Anmerkung: `/var/rdz/WORKAREA/` erfordert die Berechtigungsbitmaske 777, um jedem Client die Erstellung von temporären Dateien zu ermöglichen.

- `/tmp/` wird vom TSO/ISPF-Client-Gateway von ISPF und verschiedenen Miners verwendet, um temporäre Daten zu speichern. Die Position ist nicht anpassbar. Die enthaltenen Dateien werden von ISPF und den Miners verwaltet. Das Verzeichnis ist außerdem die Standardposition für Java-Speicherauszugsdateien, die mit der Variable `_CEE_DUMPTARG` in `rsed.envvars` angepasst werden kann.

Anmerkung: `/tmp/` erfordert die Berechtigungsbitmaske 777, um jedem Client das Erstellen von temporären Daten zu ermöglichen.

Aktualisierungsberechtigungen für Benutzer ohne Systemadministratorrechte

Die in einigen Verzeichnissen (wie in `/var/rdz/projects/`) enthaltenen Daten werden von Benutzern ohne Administratorrechte, beispielsweise Projektmanagern, verwaltet. Diese Benutzer haben möglicherweise kaum Aktualisierungsberechtigungen in z/OS UNIX. Wenn die Dateien von nur einer Benutzer-ID verwaltet werden, können dem Benutzer Aktualisierungsberechtigungen zugewiesen werden, indem die Benutzer-ID als Eigentümer des Verzeichnisses und der darin enthaltenen Daten festgelegt wird.

```
chown -R IBMUSER /var/rdz/projects/
```

Wenn mehrere Benutzer-IDs Aktualisierungsberechtigungen für das Verzeichnis benötigen, können Sie Gruppenberechtigungsbits verwenden.

1. Erstellen Sie eine Gruppe in Ihrer Sicherheitssoftware, die über ein gültiges OMVS-Segment verfügt und stellen Sie eine Verbindung zu allen Benutzer-IDs her, die Aktualisierungszugriff erfordern. Dies ist vorzugsweise die Standardgruppe.

```
ADDGROUP RDZPROJ OMVS(GID(1200))
CONNECT IBMUSER GROUP(RDZPROJ)
ALTUSER IBMUSER DFLTGRP(RDZPROJ)
```

2. Verwenden Sie den z/OS UNIX-Befehl **chgrp**, um die Gruppe dem Verzeichnis und allen enthaltenen Dateien zuzuordnen. Dieser Befehl muss jedes Mal erneut ausgeführt werden, wenn dem Verzeichnis eine Datei hinzugefügt wird und die gewünschte Gruppen-ID nicht der Standardgruppe für die Benutzer-ID entspricht, die die Datei hinzugefügt hat.

```
chgrp -R IBMUSER /var/rdz/projects/
```

3. Verwenden Sie den z/OS UNIX-Befehl **chmod**, um der gesamten Gruppe Aktualisierungsberechtigungen für das Verzeichnis und alle darin enthaltenen Dateien zu erteilen.

```
chmod -R 775 /var/rdz/projects/
```

Kapitel 12. Hinweise zu WLM

Im Gegensatz zu herkömmlichen z/OS-Anwendungen ist Developer for System z keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for System z umfasst mehrere interagierende Komponenten, damit der Client auf die Host-Services und -daten zugreifen kann. Wie in Kapitel 11, „Wissenswertes zu Developer for System z“, auf Seite 193 beschrieben, sind einige dieser Services in verschiedenen Adressräumen aktiv und werden somit verschiedenen WLM-Klassifikationen zugeordnet.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Klassifikation für Verarbeitungsprozesse“
- „Ziele festlegen“ auf Seite 207

Klassifikation für Verarbeitungsprozesse

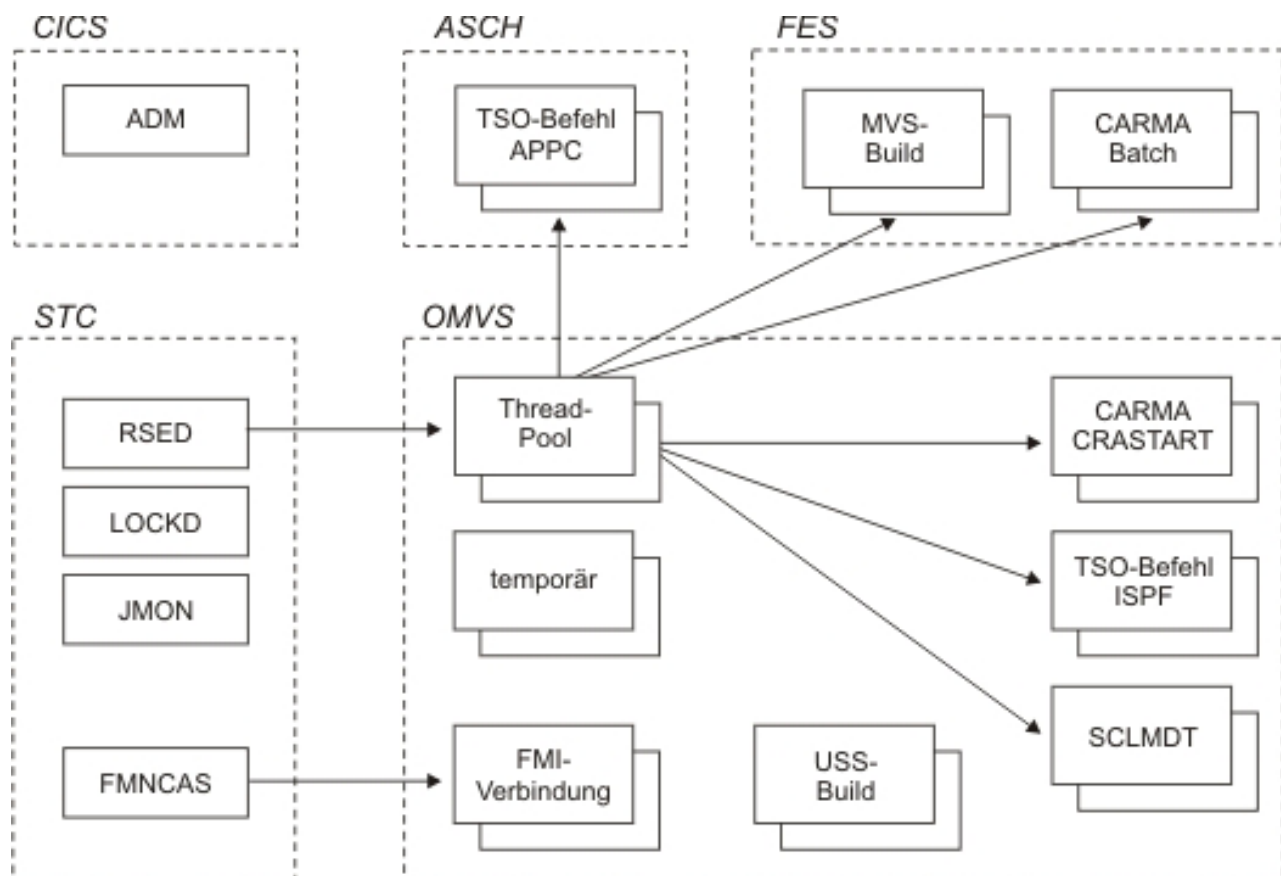


Abbildung 47. WLM-Klassifikation

Abb. 47 zeigt eine Basisübersicht über die Subsysteme, über die die Informationen zu den Verarbeitungsprozessen von Developer for System z an WLM weitergegeben werden.

Application Deployment Manager (ADM) ist innerhalb einer CICS-Region aktiv und befolgt deshalb die CICS-Klassifikationsregeln in WLM.

Der RSE-Dämon (RSED), der Sperrdämon (LOCKD) und JES Job Monitor (JMON) sind gestartete Tasks in Developer for System z (oder lang andauernde Batch-Jobs) mit individuellen Adressräumen.

Wie in „RSE als Java-Anwendung“ auf Seite 195 dokumentiert, startet der RSE-Dämon für jeden RSE-Thread-Pool-Server (der eine variable Anzahl von Clients unterstützt) einen untergeordneten Prozess. Jeder Thread-Pool ist (mithilfe eines z/OS UNIX-Initiators, BPXAS) in einem separaten Adressraum aktiv. Da es sich hierbei um gestartete Prozesse handelt, werden diese nach den WLM-OMVS-Klassifikationsregeln und nicht nach den Klassifikationsregeln für gestartete Tasks klassifiziert.

Abhängig von den Aktionen der Benutzer können die Clients, die in einem Thread-Pool aktiv sind, eine Vielzahl anderer Adressräume erstellen. Abhängig von der Konfiguration von Developer for System z, können einige Verarbeitungsprozesse, wie TSO Commands Service (TSO cmd) oder CARMA, in anderen Subsystemen ausgeführt werden.

Die in Abb. 47 auf Seite 205 aufgeführten Adressräume bleiben für einen längeren Zeitraum im System sichtbar. z/OS UNIX wurde jedoch so entwickelt, dass es auch kurz andauernde, temporäre Adressräume gibt. Diese temporären Adressräume sind im OMVS-Subsystem aktiv.

Während die RSE-Thread-Pools dieselbe Benutzer-ID und einen ähnlichen Jobnamen wie der RSE-Dämon verwenden, gehören alle von einem Thread-Pool gestarteten Adressräume der Client-Benutzer-ID, die die Aktion anfordert. Die Client-Benutzer-ID wird außerdem als Teil des Jobnamens für alle vom Thread-Pool gestarteten OMVS-basierten Adressräume verwendet.

Weitere Adressräume werden von anderen Services erstellt, die Developer for System z verwendet, wie File Manager (FMNCAS) oder z/OS UNIX-REXEC (USS-Build).

Klassifikationsregeln

WLM verwendet Klassifikationsregeln, um im System eingehende Arbeit einer Serviceklasse zuzuordnen. Diese Klassifikation basiert auf Qualitätsmerkmalen für Arbeit. Das erste (verbindliche) Merkmal ist der Subsystemtyp, der die Verarbeitungsanforderung empfängt. In Tabelle 29 werden die Subsystemtypen aufgeführt, die Verarbeitungsanforderungen von Developer for System z empfangen können.

Tabelle 29. WLM-Eingangspunktsysteme

Subsystemtyp	Beschreibung der Arbeit
ASCH	Die Verarbeitungsanforderungen umfassen alle APPC-Transaktionsprogramme, die von dem von IBM gelieferten APPC/MVS-Transaktionscheduler (ASCH) geplant werden.
CICS	Die Verarbeitungsanforderungen umfassen alle Transaktionen, die von CICS verarbeitet werden.
JES	Die Verarbeitungsanforderungen umfassen alle Jobs, die von JES2 oder JES3 initialisiert werden.

Tabelle 29. WLM-Eingangspunktsysteme (Forts.)

Subsystemtyp	Beschreibung der Arbeit
OMVS	Die Verarbeitungsanforderungen umfassen Arbeit, die in verzweigten untergeordneten Adressräumen von z/OS UNIX System Services verarbeitet wird.
STC	Die Verarbeitungsanforderungen umfassen Arbeit, die von den Befehlen 'START' und 'MOUNT' initialisiert wird. STC umfasst außerdem Adressräume der Systemkomponente.

In Tabelle 2 werden zusätzliche Merkmale aufgeführt, die für die Zuordnung von Verarbeitungsprozessen zu einer bestimmten Serviceklasse verwendet werden können. Weitere Details zu den aufgelisteten Merkmalen enthält MVS Planning: Workload Management (IBM Form SA22-7602).

Tabelle 30. WLM-Qualifikationsmerkmale für Arbeitsvorgänge

		ASCH	CICS	JES	OMVS	STC
AI	Accountinformationen	x		x	x	x
LU	LU-Name (*)		x			
PF	Ausführung (*)			x		x
PRI	Priorität			x		
SE	Name der Terminierungsumgebung			x		
SSC	Objektgruppenname des Subsystems			x		
SI	Subsysteminstanz (*)		x	x		
SPM	Subsystemparameter					x
PX	Sysplex-Name	x	x	x	x	x
SY	Systemname (*)	x			x	x
TC	Transaktions-/Jobklasse (*)	x		x		
TN	Transaktions-/Jobname (*)	x	x	x	x	x
UI	Benutzer-ID (*)	x	x	x	x	x

Anmerkung: Für die mit Stern (*) markierten Merkmale können Klassifikationsgruppen angegeben werden, indem der Abkürzung des Typs ein 'G' hinzugefügt wird. Eine Gruppe für den Transaktionsnamen würde beispielsweise 'TNG' lauten.

Ziele festlegen

Wie unter „Klassifikation für Verarbeitungsprozesse“ auf Seite 205 dokumentiert, erstellt Developer for System z unterschiedliche Typen von Verarbeitungsprozessen auf Ihrem System. Diese verschiedenen Tasks kommunizieren miteinander. Dafür ist die eigentliche Antwortzeit wichtig, um Zeitüberschreitungsprobleme bezüglich der Verbindungen zwischen den Tasks zu vermeiden. Deshalb sollten Tasks in Developer for System z in leistungsfähige Serviceklassen oder in Serviceklassen mit mittlerer Leistung mit hoher Priorität eingeordnet werden.

Es wird daher eine Überarbeitung und gegebenenfalls eine Aktualisierung Ihrer aktuellen WLM-Ziele empfohlen. Dies gilt insbesondere für herkömmliche MVS-Unternehmen, für die zeitkritische OMVS-Verarbeitungsprozesse neu sind.

Anmerkung:

- Die Zielinformationen in diesem Abschnitt sind bewusst beschreibend gehalten, da die eigentlichen Leistungsziele sehr vom jeweiligen Standort abhängig sind.
- Um die Auswirkungen einer bestimmten Task auf Ihrem System besser zu verstehen, werden Bezeichnungen wie 'minimale Ressourcennutzung', 'mäßige Ressourcennutzung' und 'erhebliche Ressourcennutzung' verwendet. Diese Angaben sind relativ zur Gesamtressourcennutzung von Developer for System z, nicht vom gesamten System, zu verstehen.

In Tabelle 31 werden die Adressräume aufgelistet, die von Developer for System z verwendet werden. z/OS UNIX ersetzt den Wert "x" in der Spalte "Taskname" durch eine zufällige einstellige Zahl.

Tabelle 31. WLM-Verarbeitungsprozesse

Beschreibung	Taskname	Verarbeitungsprozess
JES Job Monitor	JMON	STC
Sperrdämon	LOCKD	STC
RSE-Dämon	RSED	STC
RSE-Thread-Pool	RSEDx	OMVS
ISPF Client Gateway (TSO Commands Service und SCLMDT)	<Benutzer-ID>x	OMVS
TSO Commands Service (APPC)	FEKFRSRV	ASCH
CARMA (batch)	CRA<port>	JES
CARMA (crastart)	<Benutzer-ID>x	OMVS
CARMA (ISPF-Client-Gateway)	<Benutzer-ID> und <Benutzer-ID>x	OMVS
MVS-Build (Batch-Job)	*	JES
z/OS UNIX-Build (Shellbefehle)	<Benutzer-ID>x	OMVS
z/OS UNIX-Shell	<Benutzer-ID>	OMVS
File Manager-Task	<Benutzer-ID>x	OMVS
Application Deployment Manager	CICSTS	CICS

Hinweise zur Zielauswahl

Die folgenden allgemeinen Hinweise zu WLM unterstützen Sie beim Definieren der Zieldefinitionen für Developer for System z:

- Ihre Zieldefinitionen sollten darauf aufbauen, was tatsächlich erreicht werden kann, und nicht darauf, was Sie gern erreichen möchten. Wenn Sie Ziele höher als notwendig setzen, verschiebt WLM Ressourcen von Arbeitsvorgängen geringerer Wichtigkeit zu Arbeitsvorgängen größerer Wichtigkeit, die die Ressourcen möglicherweise gar nicht benötigen.

- Begrenzen Sie den Arbeitsbetrag, der den Serviceklassen "SYSTEM" und "SYSSTC" zugewiesen wird. Diese Klassen haben eine höhere Zuteilungspriorität als alle anderen von WLM verwalteten Klassen. Verwenden Sie diese Klassen für Arbeitsvorgänge, die sehr wichtig sind, aber eine geringe CPU-Auslastung verursachen.
- Arbeitsvorgänge, die den Klassifikationsregeln nicht entsprechen, werden der Klasse "SYSOTHER" zugeordnet. Diese Klasse verfolgt ein ressourcenabhängiges Ziel. Ein ressourcenabhängiges Ziel bewirkt, dass WLM im Fall freier Ressourcen die Arbeitsvorgänge dieser Klasse berücksichtigt.

Bei der Verwendung von Antwortzeitzielen:

- Damit WLM ein Antwortzeitziel erfolgreich steuern kann, muss eine stetige Taskrate eingehen (mindestens 10 Tasks in 20 Minuten).
- Verwenden Sie durchschnittliche Antwortzeitziele nur bei gut gesteuerten Verarbeitungsprozessen. Eine einzelne lange Transaktion hat eine erhebliche Auswirkung auf die durchschnittliche Antwortzeit und kann eine Überreaktion von WLM hervorrufen.

Bei der Verwendung von Geschwindigkeitszielen:

- Sie erreichen Geschwindigkeitsziele normalerweise nur zu 90 Prozent. Das hat verschiedene Ursachen. Die Adressräume "SYSTEM" und "SYSSTC" haben beispielsweise eine höhere Zuteilungspriorität als Geschwindigkeitsziele.
- WLM basiert seine Geschwindigkeitszielentscheidungen auf einer minimalen Anzahl von Stichproben. Je weniger Arbeit in einer Serviceklasse ausgeführt wird, umso länger dauert es, die erforderliche Anzahl von Stichproben zu sammeln und die Zuteilungsrichtlinie anzupassen.
- Überprüfen Sie Geschwindigkeitsziele erneut, wenn Sie Ihre Hardware ändern. Insbesondere der Einsatz von weniger und schnelleren Prozessoren erfordert Änderungen an den Geschwindigkeitszielen.

STC

Alle gestarteten Tasks von Developer for System z (RSE-Dämon, Sperrdämon und JES Job Monitor) warten Echtzeit-Clientanforderungen.

Tabelle 32. WLM-Verarbeitungsprozesse - STC

Beschreibung	Taskname	Verarbeitungsprozess
JES Job Monitor	JMON	STC
Sperrdämon	LOCKD	STC
RSE-Dämon	RSED	STC

- JES Job Monitor
JES Job Monitor stellt alle Services mit Bezug zum JES bereit. Dazu gehören das Übergeben von Jobs, das Durchsuchen von Spooldateien und das Ausführen von JES-Bedienerbefehlen. Sie sollten ein leistungsfähiges Geschwindigkeitsziel für einen Zeitraum angeben, da die Task WLM keine einzelnen Transaktionen zurückmeldet. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal bis mäßig ist.
- Sperrdämon
Der Sperrdämon fragt die GRS-Serialisierungstabellen auf Client- und Bedieneranforderungen hin ab und vergleicht das Ergebnis mit bekannten Benutzern in Developer for System z. Sie sollten ein leistungsfähiges Geschwindigkeitsziel für

einen Zeitraum angeben, da die Task WLM keine einzelnen Transaktionen zurückmeldet. Es wird eine minimale Ressourcennutzung erwartet.

- RSE-Dämon

Der RSE-Dämon führt die Clientanmeldung und -authentifizierung aus und verwaltet die verschiedenen RSE-Thread-Pools. Sie sollten ein leistungsfähiges Geschwindigkeitsziel für einen Zeitraum angeben, da die Task WLM keine einzelnen Transaktionen zurückmeldet. Es wird eine mäßige Ressourcennutzung mit einem Spitzenwert zu Beginn des Arbeitstages erwartet.

OMVS

Die OMVS-Verarbeitungsprozesse können in zwei Gruppen unterteilt werden: RSE-Thread-Pools und alle anderen Verarbeitungsprozesse. Dies hat zur Ursache, dass alle Verarbeitungsprozesse, außer RSE-Thread-Pools, die Client-Benutzer-ID als Grundlage für den Adressraumnamen verwenden. (z/OS UNIX ersetzt den Wert "x" in der Spalte "Taskname" durch eine zufällige einstellige Zahl.)

Tabelle 33. WLM-Verarbeitungsprozesse - OMVS

Beschreibung	Taskname	Verarbeitungsprozess
RSE-Thread-Pool	RSEDx	OMVS
ISPF Client Gateway (TSO Commands Service und SCLMDT)	<Benutzer-ID>x	OMVS
CARMA (crastart)	<Benutzer-ID>x	OMVS
CARMA (ISPF-Client-Gateway)	<Benutzer-ID> und <Benutzer-ID>x	OMVS
z/OS UNIX-Build (Shellbefehle)	<Benutzer-ID>x	OMVS
z/OS UNIX-Shell	<Benutzer-ID>	OMVS
File Manager-Task	<Benutzer-ID>x	OMVS

- RSE-Thread-Pool

Ein RSE-Thread-Pool ist das Herzstück von Developer for System z. Beinahe alle Daten fließen durch diesen Pool und die Miners (benutzerspezifische Threads) innerhalb des Thread-Pools steuern die Aktionen der meisten anderen Tasks in Bezug auf Developer for System z. Sie sollten ein leistungsfähiges Geschwindigkeitsziel für einen Zeitraum angeben, da die Task WLM keine einzelnen Transaktionen zurückmeldet. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie erheblich ist.

Die restlichen Verarbeitungsprozesse werden alle aufgrund einer allgemeinen Namenskonvention für Adressräume derselben Serviceklasse zugeordnet. Für diese Serviceklasse sollten Sie ein Ziel für mehrere Zeiträume angeben. Für die ersten Zeiträume sollten Sie leistungsfähige Perzentilantwortzeitziele und für den letzten Zeitraum ein Geschwindigkeitsziel mit mittlerer Leistung angeben. Einige Verarbeitungsprozesse, wie das ISPF-Client-Gateway, melden WLM einzelne Transaktionen zurück.

- ISPF Client Gateway

Das ISPF-Client-Gateway ist ein ISPF-Service, der von Developer for System z aufgerufen wird, um nicht interaktive TSO- und ISPF-Befehle auszuführen. Dies schließt sowohl vom Client ausgegebene, explizite Befehle als auch von Developer for System z ausgegebene, implizite Befehle ein, z. B. das Abrufen einer

PDS-Memberliste. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal ist.

- CARMA

CARMA ist ein optionaler Developer for System z-Server, der für die Interaktion mit hostbasierten Software Configuration Managers (SCMs), wie CA Endevor[®] SCM, verwendet wird. Developer for System z lässt verschiedene Startmethoden für einen CARMA-Server zu. Einige davon werden als OMVS-Verarbeitungsprozess gehandhabt. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal ist.

- z/OS UNIX-Build

Wenn ein Client einen Build für ein z/OS UNIX-Projekt initialisiert, startet die z/OS UNIX-REXEC (oder SSH) eine Task, die zur Ausführung des Builds eine Reihe von z/OS UNIX-Shellbefehlen ausführt. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie mäßig bis erheblich ist (abhängig von der Größe des Projekts).

- z/OS UNIX-Shell

Dieser Verarbeitungsprozess verarbeitet vom Client ausgegebene z/OS UNIX-Shellbefehle. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal ist.

- IBM File Manager

Obwohl sie keine Adressräume von Developer for System z sind, sind die gestarteten untergeordneten Prozesse von File Manager hier aufgelistet. Das hat zur Ursache, dass sie auf Anforderung des Developer for System z-Clients gestartet werden können und dieselbe Namenskonvention wie Developer for System z-Tasks verwenden. Diese File Manager-Tasks verarbeiten nicht triviale MVS-Dateiaktionen, wie das formatierte Bearbeiten einer VSAM-Datei. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal bis mäßig ist.

JES

JES-verwaltete Batchprozesse werden von Developer for System z auf verschiedene Weisen verwendet. Die bekannteste Nutzung ist für MVS-Builds, für die ein Job übergeben und überwacht wird, um sein Ende zu bestimmen. Developer for System z kann jedoch auch einen CARMA-Server mit Batchübergabe starten und mit ihm über TCP/IP kommunizieren.

Tabelle 34. WLM-Verarbeitungsprozesse - JES

Beschreibung	Taskname	Verarbeitungsprozess
CARMA (batch)	CRA<port>	JES
MVS-Build (Batch-Job)	*	JES

- CARMA

CARMA ist ein optionaler Developer for System z-Server, der für die Interaktion mit hostbasierten Software Configuration Managers (SCMs), wie CA Endevor[®] SCM, verwendet wird. Developer for System z lässt verschiedene Startmethoden für einen CARMA-Server zu. Einige davon werden als JES-Verarbeitungsprozess gehandhabt. Sie sollten ein leistungsfähiges Geschwindigkeitsziel für einen Zeitraum angeben, da die Task WLM keine einzelnen Transaktionen zurückmeldet.

Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal ist.

- MVS-Build

Wenn ein Client einen Build für ein MVS-Projekt initialisiert, startet Developer for System z zur Ausführung des Builds einen Batch-Job. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie mäßig bis erheblich ist (abhängig von der Größe des Projekts). Abhängig von Ihren lokalen Bedingungen können verschiedene Zielstrategien mit mittlerer Leistung sinnvoll sein.

- Sie können ein Ziel für mehrere Zeiträume angeben: Für den ersten Zeitraum legen Sie ein Perzentilantwortzeitziel und für den zweiten Zeitraum ein Geschwindigkeitsziel fest. In diesem Fall sollten Ihre Entwickler hauptsächlich dieselbe Buildprozedur und Eingabedateien ähnlicher Größe verwenden, um Jobs mit einheitlichen Antwortzeiten zu erstellen. Damit WLM ein Antwortzeitziel erfolgreich steuern kann, muss auch eine stetige Jobrate eingehen (mindestens 10 Jobs in 20 Minuten).
- Ein Geschwindigkeitsziel ist für die meisten Batch-Jobs am besten geeignet, da dieses Ziel stark schwankende Ausführungszeiten und Eingangsdaten handhaben kann.

ASCH

In den aktuellen Versionen von Developer for System z wird das ISPF-Client-Gateway verwendet, um nicht interaktive TSO- und ISPF-Befehle auszuführen. Aus historischen Gründen unterstützt Developer for System z die Ausführung dieser Befehle auch über eine APPC-Transaktion.

Tabelle 35. WLM-Verarbeitungsprozesse - ASCH

Beschreibung	Taskname	Verarbeitungsprozess
TSO Commands Service (APPC)	FEKFRSRV	ASCH

- TSO Commands Service

TSO Commands Service kann von Developer for System z als APPC-Transaktion gestartet werden, um nicht interaktive TSO- und ISPF-Befehle auszuführen. Dies schließt sowohl vom Client ausgegebene, explizite Befehle als auch von Developer for System z ausgegebene, implizite Befehle ein, z. B. das Abrufen einer PDS-Memberliste. Für diese Serviceklasse sollten Sie ein Ziel für mehrere Zeiträume angeben. Für die ersten Zeiträume sollten Sie leistungsfähige Perzentilantwortzeitziele angeben. Für den letzten Zeitraum sollten Sie ein Geschwindigkeitsziel mit mittlerer Leistung angeben. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal ist.

CICS

Application Deployment Manager ist ein optionaler Developer for System z-Server, der innerhalb einer CICS Transaction Server-Region aktiv ist.

Tabelle 36. WLM-Verarbeitungsprozesse - CICS

Beschreibung	Taskname	Verarbeitungsprozess
Application Deployment Manager	CICSTS	CICS

- **Application Deployment Manager**

Der optionale Application Deployment Manager-Server, der innerhalb einer CICSTS-Region aktiv ist, ermöglicht Ihnen das sichere Auslagern ausgewählter CICSTS-Verwaltungstasks an Entwickler. Die Ressourcennutzung hängt im großen Maße von den Benutzeraktionen ab und kann deshalb schwanken. Es ist jedoch zu erwarten, dass sie minimal ist. Der zu verwendende Serviceklassentyp hängt von den anderen in dieser CICS-Region aktiven Transaktionen ab und ist deshalb nicht im Detail beschrieben.

WLM unterstützt mehrere Verwaltungstypen, die Sie für CICS verwenden können:

- **CICS mit einem Regionsziels verwalten**

Das Ziel ist auf eine Serviceklasse festgelegt, die die CICS-Adressräume verwaltet. Für diese Serviceklasse können Sie nur ein Ziel für die Ausführungsgeschwindigkeit festlegen. WLM verwendet für die Adressräume die JES- oder STC-Klassifikationsregeln. Es verwendet jedoch nicht die CICS-Subsystemklassifikationsregeln für Transaktionen.

- **CICS mit einem Antwortzeitziel für Transaktionen verwalten**

Ein Antwortzeitziel kann in einer Serviceklasse festgelegt werden, die einer einzelnen Transaktion oder einer Gruppe von Transaktionen zugewiesen ist. WLM verwendet für die Adressräume die JES- oder STC-Klassifikationsregeln und die CICS-Subsystemklassifikationsregeln für Transaktionen.

Kapitel 13. Optimierungsaspekte

Wie in Kapitel 11, „Wissenswertes zu Developer for System z“, auf Seite 193 erklärt wird, ist RSE (Remote Systems Explorer) der zentrale Bestandteil von Developer for System z. RSE besteht aus einem Dämonadressbereich, der Thread-Pooling und Adressräume steuert, um die Verbindungen und die Arbeitslast der Clients zu verwalten. Der Dämon wird als Sammelpunkt für Verbindungs- und Verwaltungszwecke eingesetzt, während die Thread-Pools die Clientarbeitslast verarbeiten.

Dadurch wird RSE das Hauptziel für die Optimierung der Installation von Developer for System z. Wenn Sie allerdings Hunderte von Benutzern verwalten, die jeweils mindestens 16 Threads, eine bestimmte Speichermenge und mindestens einen Adressraum verwenden, müssen Developer for System z und z/OS richtig konfiguriert sein.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Ressourcennutzung“
- „Speicherbelegung“ auf Seite 226
- „Speicherbelegung im z/OS UNIX-Dateisystem“ auf Seite 232
- „Definitionen von wichtigen Ressourcen“ auf Seite 235
- „Definitionen von verschiedenen Ressourcen“ auf Seite 238
- „Überwachung“ auf Seite 240
- „Beispielkonfiguration“ auf Seite 244

Ressourcennutzung

Verwenden Sie die Informationen in diesem Abschnitt, um die normale und maximale Ressourcennutzung von Developer for System z zu berechnen und Ihre Systemkonfiguration entsprechend planen zu können.

Wenn Sie die in diesem Abschnitt vorhandenen Zahlen und Formeln verwenden, um die Grenzwerte für das System zu definieren, achten Sie darauf, dass Sie mit präzisen Berechnungen arbeiten. Planen Sie beim Festlegen der Begrenzungen für das System ausreichend Spielraum ein, um die Ressourcennutzung für temporäre oder andere Tasks sowie für Benutzer zu ermöglichen, die sich mehrfach gleichzeitig am Host anmelden (beispielsweise über RSE und TN3270).

Anmerkung:

- Diese Informationen gelten nur für Services, auf die über RSE zugegriffen wird und die direkt von Developer for System z bereitgestellt werden. Die Ressourcennutzung von TN3270 ist beispielsweise nicht dokumentiert (der Zugriff erfolgt nicht über RSE). Die Ressourcennutzung von Programmen, die während fernen (hostbasierten) MVS-Builds aufgerufen werden, oder die Ressourcennutzung von z/OS UNIX-Projekten (die nicht direkt von Developer for System z bereitgestellt werden) sind ebenfalls nicht dokumentiert.
- Das Hinzufügen von Erweiterungen anderer Anbieter zu Developer for System z kann die Ressourcennutzung erhöhen.

- Alle Services enthalten kurz andauernde Verwaltungstasks, die bei ihrer Ausführung Ressourcen verwenden und möglicherweise sequenziell oder parallel zueinander ausgeführt werden. Die von diesen Tasks verwendeten Ressourcen sind nicht dokumentiert.
- Die benutzerspezifische Ressourcennutzung von vorausgesetzten Softwareprogrammen, wie ISPF Client Gateway, ist an geeigneter Stelle dokumentiert.
- Die hier dargestellten Zahlen können ohne vorherigen Hinweis geändert werden.

Überblick

Die folgenden Tabellen geben einen Überblick über die Anzahl der Adressräume, Prozesse und Threads, die von Developer for System z verwendet werden. Weitere Details zu den hier dargestellten Zahlen finden Sie in den darauffolgenden Abschnitten:

- „Anzahl der Adressräume“ auf Seite 217
- „Anzahl der Prozesse“ auf Seite 220
- „Anzahl der Threads“ auf Seite 223

Tabelle 37 gibt einen allgemeinen Überblick über Schlüsselressourcen, die von gestarteten Tasks in Developer for System z verwendet werden. Diese Ressourcen werden nur einmal angelegt. Sie werden von allen Developer for System z-Clients gemeinsam genutzt.

Tabelle 37. Allgemeine Ressourcennutzung

Gestartete Task	Adressräume	Prozesse	Threads
JMON	1	1	3
LOCKD	1	3	10
RSED	1	3	11
RSEDx	(a)	2	10

Anmerkung: (a) Mindestens ein Adressraum der RSE-Thread-Pools ist aktiv. Die eigentliche Anzahl der Adressräume der RSE-Thread-Pools ist im Abschnitt „Anzahl der Adressräume“ auf Seite 217 angegeben.

Tabelle 38 gibt einen allgemeinen Überblick über Schlüsselressourcen, die von vorausgesetzten Softwareprogrammen verwendet werden. Diese Ressourcen werden für jeden Developer for System z-Client angelegt, der die entsprechende Funktion aufruft.

Tabelle 38. Benutzerspezifische vorausgesetzte Ressourcennutzung

Vorausgesetzte Software	Adressräume	Prozesse	Threads
ISPF Client Gateway	1	2	4
APPC-Administrator	1	1	2
File Manager	1	1	2

Tabelle 39 gibt einen allgemeinen Überblick über Schlüsselressourcen, die von jedem Developer for System z-Client bei der Ausführung der angegebenen Funktion verwendet werden. Werte, die keine numerischen Werte sind (beispielsweise ISPF), verweisen auf den entsprechenden Wert in Tabelle 38 auf Seite 216.

Tabelle 39. Benutzerspezifische Ressourcennutzung

Benutzeraktion	Adressräume	Prozesse	Threads		
	Benutzer-ID	Benutzer-ID	Benutzer-ID	RSEDx	JMON
Anmelden	-	-	-	16	1
Zeitgeber für Inaktivitätszeit-limit	-	-	-	1	-
Komprimierung für PDS(E) aufheben	ISPF	ISPF	ISPF	-	-
Datei öffnen	ISPF	ISPF	ISPF	-	-
TSO-Befehl	ISPF	ISPF	ISPF	-	-
z/OS UNIX-Shell	1	1	1	6	-
MVS-Build	1	-	-	-	-
z/OS UNIX-Build	3	3	3	-	-
CARMA (batch)	1	1	2	1	-
CARMA (crastart)	1	1	2	4	-
CARMA (ispf)	4	4	7	5	-
SCLMDT	ISPF	ISPF	ISPF	-	-
File Manager-Integration	ISPF + FM	ISPF + FM	ISPF + FM	-	-
Fault Analyzer-Integration	-	-	-	-	-

Anmerkung: ISPF kann durch APPC ersetzt werden, außer bei SCLM Developer Toolkit.

Anzahl der Adressräume

In Tabelle 40 werden die Adressräume aufgelistet, die Developer for System z verwendet, wobei "u" in der Spalte "Anzahl" angibt, dass der Betrag mit der Anzahl der gleichzeitig aktiven Benutzer dieser Funktion multipliziert werden muss. z/OS UNIX ersetzt den Wert "x" in der Spalte "Taskname" durch eine zufällige einstellige Zahl.

Tabelle 40. Anzahl der Adressräume

Anzahl	Beschreibung	Taskname	Gemeinsame Nutzung	Endet nach
1	JES Job Monitor	JMON	Ja	Nie
1	Sperrdämon	LOCKD	Ja	Nie
1	RSE-Dämon	RSED	Ja	Nie
(a)	RSE-Thread-Pool	RSEDx	Ja	Nie

Tabelle 40. Anzahl der Adressräume (Forts.)

Anzahl	Beschreibung	Taskname	Gemeinsame Nutzung	Endet nach
1u	ISPF Client Gateway (TSO Commands Service und SCLMDT)	<Benutzer-ID>x	Nein	15 Minuten oder Abmeldung des Benutzers
1u	TSO Commands Service (APPC)	FEKFRSRV	Nein	60 Minuten oder Abmeldung des Benutzers
1u	CARMA (batch)	CRA<port>	Nein	7 Minuten oder Abmeldung des Benutzers
1u	CARMA (crastart)	<Benutzer-ID>x	Nein	7 Minuten oder Abmeldung des Benutzers
4u	CARMA (ispf)	(1)<Benutzer-ID> oder (3)<Benutzer-ID>x	Nein	7 Minuten oder Abmeldung des Benutzers
(b)	Simultane Verwendung von ISPF Client Gateway von 1 Benutzer	<Benutzer-ID>x	Nein	Fertigstellung der Task
1u	MVS-Build (Batch-Job)	*	Nein	Fertigstellung der Task
3u	z/OS UNIX-Build (Shellbefehle)	<Benutzer-ID>x	Nein	Fertigstellung der Task
1u	z/OS UNIX-Shell	<Benutzer-ID>	Nein	Abmeldung des Benutzers
(c)	File Manager	<Benutzer-ID>x	Nein	Fertigstellung der Task

Anmerkung:

- (a) Mindestens ein Adressraum der RSE-Thread-Pools ist aktiv. Die eigentliche Anzahl hängt ab von:
 - der Anweisung `minimum.threadpool.process` in `rsed.envvars`. Der Standardwert ist 1.
 - der Anzahl der Benutzer, die ein Thread-Pool bedienen kann. Die Standardeinstellungen sind auf 60 Benutzer pro Thread-Pool festgelegt.
 - der oberen Grenze der gleichzeitig aktiven Benutzer, da inaktive Thread-Pools nicht automatisch stoppen.
- (b) In Developer for System z sind mehrere Threads pro Benutzer aktiv. In dem Fall, dass der Adressraum von ISPF Client Gateway das Antworten auf eine Anforderung eines Threads noch nicht beendet hat, während ein anderer Thread eine neue Anforderung sendet, wird für die Verarbeitung der neuen Anforderung ein neues Client-Gateway von ISPF geöffnet. Dieser Adressraum endet mit dem Abschluss der Task.
- (c) Der File Manager-Listener startet einen Adressraum für jedes Objekt, das bearbeitet werden muss, beispielsweise eine VSAM. Dieser Adressraum bleibt solange aktiv, bis Developer for System z das Signal sendet, dass das Objekt nicht länger benötigt wird, beispielsweise indem er die VSAM schließt.
- Für SCLMDT ist ein Adressraum von ISPF Client Gateway erforderlich. SCLMDT nutzt den Adressraum gemeinsam mit TSO Commands Service.
- Die meisten Aktionen, die mit MVS-Dateien verknüpft sind, verwenden TSO Commands Service, das in einer Transaktion mit ISPF Client Gateway beziehungsweise APPC aktiv sein kann.

Verwenden Sie die Formel in Abb. 48, um die maximale Anzahl der Adressräume zu berechnen, die Developer for System z verwendet.

$$3 + A + N \cdot (x + y + z) + (2 + N \cdot 0.01)$$

Abbildung 48. Maximale Anzahl von Adressräumen

Dabei

- entspricht "3" der Anzahl von permanent aktiven Serveradressräumen.
- stellt "A" die Anzahl der Adressräume der RSE-Thread-Pools dar.
- stellt "N" die maximale Anzahl von gleichzeitigen Benutzern dar.
- entspricht "x" einem der folgenden Werte (abhängig von den ausgewählten Konfigurationsoptionen).

X	SCLMDT	TSO über Client-Gateway	TSO über APPC
1	Nein	Nein	Ja
1	Nein	Ja	Nein
1	Ja	Ja	Nein

- "y" entspricht einem der folgenden Werte (abhängig von den ausgewählten Konfigurationsoptionen).

Y	
0	Kein CARMA
1	CARMA (batch)
1	CARMA (crastart)
4	CARMA (ispf)

- "z" ist standardmäßig 0, kann sich jedoch abhängig von den Benutzeraktionen erhöhen.
 - Fügen Sie 1 hinzu, wenn ein MVS-Build ausgeführt wird. Diese Adressräume enden mit dem Abschluss der zugehörigen Build-Task (Batch-Job).
 - Fügen Sie 3 hinzu, wenn ein z/OS UNIX-Build ausgeführt wird. Beachten Sie, dass die eigentliche Anzahl höher sein kann. Das hängt von den Bedürfnissen der aufgerufenen Programme ab. Diese Adressräume enden mit dem Abschluss der zugehörigen Build-Task.
 - Fügen Sie 1 für jede gleichzeitig ablaufende Interaktion mit IBM File Manager hinzu. Diese Adressräume enden, wenn das angeforderte Objekt nicht länger benötigt wird.
- "2 + N*0.01" fügt einen Puffer für temporäre Adressräume hinzu. Die erforderliche Puffergröße kann an Ihrem Standort abweichen.

Verwenden Sie die Formel in Abb. 49, um die maximale Anzahl der Adressräume zu berechnen, die ein Developer for System z-Client verwendet (temporäre Adressräume (nicht dokumentiert) werden nicht berücksichtigt).

$$x + y + z$$

Abbildung 49. Anzahl der Adressräume pro Client

Dabei

- hängt "x" von den ausgewählten Konfigurationsoptionen ab und wird für die Formel dokumentiert, um die maximale Anzahl von Adressräumen zu berechnen (Abb. 48 auf Seite 219).
- hängt "y" von den ausgewählten Konfigurationsoptionen ab und wird für die Formel dokumentiert, um die maximale Anzahl von Adressräumen zu berechnen (Abb. 48 auf Seite 219).
- "z" ist standardmäßig 0, kann sich jedoch abhängig von den Benutzeraktionen erhöhen. "z" wird für die Formel dokumentiert, um die maximale Anzahl von Adressräumen zu berechnen (Abb. 48 auf Seite 219).

Die Definitionen in Tabelle 41 können die eigentliche Anzahl von Adressräumen begrenzen.

Tabelle 41. Begrenzungen für Adressräume

Position	Begrenzung	Beeinträchtigte Ressourcen
rsed.envvars	maximum.threadpool.process	Begrenzt die Anzahl von RSE-Thread-Pools
IEASYMxx	MAXUSER	Begrenzt die Anzahl von Adressräumen
ASCHPMxx	MAX	Begrenzt die Anzahl von APPC-Initiatoren für TSO Commands Service (APPC)

Anzahl der Prozesse

In Tabelle 42 wird die Anzahl der Prozesse angegeben, die Developer for System z verwendet, wobei "u" in der Spalte "Adressräume" angibt, dass der Betrag mit der Anzahl der gleichzeitig aktiven Benutzer dieser Funktion multipliziert werden muss.

Tabelle 42. Anzahl der Prozesse

Prozesse	Adressräume	Beschreibung	Benutzer-ID
1	1	JES Job Monitor	STCJMON
3	1	Sperrdämon	STCLOCK
3	1	RSE-Dämon	STCRSE
2	(a)	RSE-Thread-Pool	STCRSE
2	(b)	ISPF Client Gateway (TSO Commands Service und SCLMDT)	<Benutzer-ID>
1	1u	TSO Commands Service (APPC)	<Benutzer-ID>
1	1u	CARMA (batch)	<Benutzer-ID>
1	1u	CARMA (crastart)	<Benutzer-ID>
1	1u	CARMA (ispf)	<Benutzer-ID>
1	3u	z/OS UNIX-Build (Shellbefehle)	<Benutzer-ID>
1	1u	z/OS UNIX-Shell	<Benutzer-ID>

Tabelle 42. Anzahl der Prozesse (Forts.)

Prozesse	Adressräume	Beschreibung	Benutzer-ID
1	(c)	File Manager	<Benutzer-ID>
(5)	(u)	SCLM Developer Toolkit	<Benutzer-ID>

Anmerkung:

- (a) Mindestens ein Adressraum der RSE-Thread-Pools ist aktiv. Die eigentliche Anzahl der Adressräume der RSE-Thread-Pools ist im Abschnitt „Anzahl der Adressräume“ auf Seite 217 angegeben.
- Der RSE-Dämon und alle RSE-Thread-Pools verwenden dieselbe Benutzer-ID.
- (b) Unter normalen Umständen und unter Verwendung der Standardkonfigurationsoptionen gibt es pro Benutzer einen aktiven ISPF Client Gateway. Die eigentliche Anzahl kann abweichen, wie im Abschnitt „Anzahl der Adressräume“ auf Seite 217 beschrieben.
- (c) Der File Manager-Listener verwendet einen Prozess für jedes Objekt, das bearbeitet werden muss, beispielsweise eine VSAM. Dieser Prozess bleibt solange aktiv, bis Developer for System z das Signal sendet, dass das Objekt nicht länger benötigt wird, beispielsweise indem er die VSAM schließt.
- Für SCLMDT ist ein Adressraum von ISPF Client Gateway erforderlich. SCLMDT nutzt den Adressraum gemeinsam mit TSO Commands Service.
- (u) SCLMDT-Prozesse werden in dem Adressraum von ISPF Client Gateway ausgeführt und verfügen deshalb über keinen Wert für die Anzahl von Adressräumen.
- SCLMDT-Prozesse sind temporäre Prozesse, die mit dem Abschluss der Task enden. Es können jedoch mehrere Prozesse gleichzeitig für einen einzelnen Benutzer aktiv sein. In Tabelle 42 auf Seite 220 ist die maximale Anzahl von gleichzeitig ablaufenden SCLMDT-Prozessen angegeben.
- Die meisten Aktionen, die mit MVS-Dateien verknüpft sind, verwenden TSO Commands Service, das in einer Transaktion mit ISPF Client Gateway beziehungsweise APPC aktiv sein kann.
- Ein z/OS UNIX-Build verwendet insgesamt drei Prozesse, die jeweils in ihrem eigenen Adressraum ausgeführt werden.
- Alle aufgelisteten Prozesse bleiben solange aktiv, bis der zugehörige Adressraum endet (wenn nicht anders angegeben).

Verwenden Sie die Formel in Abb. 50, um die maximale Anzahl der Prozesse zu berechnen, die Developer for System z verwendet.

$$7 + 2 * A + N * (x + y + z) + (10 + N * 0.05)$$

Abbildung 50. Maximale Anzahl von Prozessen

Dabei

- entspricht "7" der Anzahl der Prozesse, die von permanenten, aktiven Serveradressräumen verwendet werden.
- stellt "A" die Anzahl der Adressräume der RSE-Thread-Pools dar.
- stellt "N" die maximale Anzahl von gleichzeitigen Benutzern dar.
- entspricht "x" einem der folgenden Werte (abhängig von den ausgewählten Konfigurationsoptionen).

X	SCLMDT	TSO über Client-Gateway	TSO über APPC
1	Nein	Nein	Ja
2	Nein	Ja	Nein
7	Ja	Ja	Nein

- "y" entspricht einem der folgenden Werte (abhängig von den ausgewählten Konfigurationsoptionen).

Y	
0	Kein CARMA
1	CARMA (batch)
1	CARMA (crastart)
4	CARMA (ispf)

- "z" ist standardmäßig 0, kann sich jedoch abhängig von den Benutzeraktionen erhöhen.
 - Fügen Sie 1 hinzu, wenn eine z/OS UNIX-Shell geöffnet ist. Dieser Prozess bleibt solange aktiv, bis sich der Benutzer abmeldet.
 - Fügen Sie 3 hinzu, wenn ein z/OS UNIX-Build ausgeführt wird. Beachten Sie, dass die eigentliche Anzahl höher sein kann. Das hängt von den Bedürfnissen der aufgerufenen Programme ab. Diese Prozesse enden mit dem Abschluss der zugehörigen Build-Task.
 - Fügen Sie 1 für jede gleichzeitig ablaufende Interaktion mit IBM File Manager hinzu. Diese Prozesse enden, wenn das angeforderte Objekt nicht länger benötigt wird.
- "10 + N*0.05" fügt einen Puffer für temporäre Prozesse hinzu. Die erforderliche Puffergröße kann an Ihrem Standort abweichen.

Verwenden Sie die Formel in Abb. 51, um die maximale Anzahl der Prozesse zu berechnen, die ein Developer for System z-Client verwendet (temporäre Prozesse (nicht dokumentiert) werden nicht berücksichtigt).

$$(x + y + z) + 5*s$$

Abbildung 51. Anzahl von Prozessen pro Client

Dabei

- hängt "x" von den ausgewählten Konfigurationsoptionen ab und wird für die Formel dokumentiert, um die maximale Anzahl von Prozessen zu berechnen (Abb. 50 auf Seite 221).
- hängt "y" von den ausgewählten Konfigurationsoptionen ab und wird für die Formel dokumentiert, um die maximale Anzahl von Prozessen zu berechnen (Abb. 50 auf Seite 221).

- "z" ist standardmäßig 0, kann sich jedoch abhängig von den Benutzeraktionen erhöhen. "z" wird für die Formel dokumentiert, um die maximale Anzahl von Prozessen zu berechnen (Abb. 50 auf Seite 221).
- "s" entspricht 1, wenn SCLM Developer Toolkit verwendet wird. Ist dies nicht der Fall, hat "s" den Wert 0.

Die Definitionen in Tabelle 43 können die eigentliche Anzahl von Prozessen begrenzen.

Tabelle 43. Begrenzungen für Prozesse

Position	Begrenzung	Beeinträchtigte Ressourcen
BPXPRMxx	MAXPROCSYS	Begrenzt die Gesamtzahl von Prozessen
BPXPRMxx	MAXPROCUSER	Begrenzt die Anzahl von Prozessen pro z/OS UNIX-Benutzer-ID

Hinweis:

- Der RSE-Dämon und die RSE-Thread-Pools verwenden dieselbe Benutzer-ID. Da der RSE-Dämon immer wenn nötig einen neuen Thread-Pool startet, kann sich die Anzahl der Prozesse für diese Benutzer-ID erhöhen. Aus diesem Grund muss MAXPROCUSER zur Begrenzung dieses Wachstums festgelegt werden. Dies kann mit der Formel $3 + 2 \cdot A$ angegeben werden.
- Die Begrenzung in MAXPROCUSER ist für jede z/OS UNIX-Benutzer-ID (UID) eindeutig. Multiplizieren Sie die berechnete Anzahl von Prozessen pro Benutzer mit der Anzahl der gleichzeitig aktiven Clients, falls Ihre Benutzer eine Benutzer-ID gemeinsam nutzen.

Anzahl der Threads

In Tabelle 44 ist die Anzahl der Threads angegeben, die von ausgewählten Developer for System z-Funktionen verwendet wird. Dabei gibt "u" in der Spalte "Threads" an, dass der Betrag mit der Anzahl von gleichzeitigen Benutzern dieser Funktion multipliziert werden muss. Die Anzahl der Threads wird pro Prozess angegeben, da Begrenzungen auf dieser Ebene festgelegt werden.

- RSEDx: Diese Threads werden im RSE-Thread-Pool erstellt, der von mehreren Clients gemeinsam genutzt wird. Alle Threads, die in demselben Thread-Pool enden, müssen zusammengerechnet werden, um die Gesamtzahl zu erhalten.
- Aktiv: Diese Threads sind Teil des Prozesses, der eigentlich die angeforderte Funktion ausführt. Da jeder Prozess eine eigenständige Einheit ist, ist es nicht notwendig, die jeweilige Anzahl der Threads zusammenzurechnen, selbst dann nicht, wenn die Threads derselben Benutzer-ID zugeordnet sind (wenn nicht anders angegeben).
- Booten: Bootprozesse werden für das eigentliche Starten des Prozesses benötigt. Jeder Bootprozess verfügt über einen Thread und es kann mehrere aufeinanderfolgende Bootprozesse geben. Es ist nicht notwendig, die jeweilige Anzahl der Threads zusammenzurechnen.

Tabelle 44. Anzahl der Threads

Threads			Benutzer-ID	Beschreibung
RSEDx	Aktiv	Booten		
-	$3 + 1u$	-	STCJMON	JES Job Monitor
-	10	2	STCLOCK	Sperrdämon

Tabelle 44. Anzahl der Threads (Forts.)

Threads			Benutzer-ID	Beschreibung
-	11	2	STCRSE	RSE-Dämon
10 (a) + 16u	-	1 (a)	STCRSE	RSE-Thread-Pool
-	4u (b)	1u (b)	<Benutzer-ID>	ISPF Client Gateway (TSO Commands Service und SCLMDT)
-	2u	-	<Benutzer-ID>	TSO Commands Service (APPC)
1u	2u	-	STCRSE und <Benutzer-ID>	CARMA (batch)
4u	2u	-	STCRSE und <Benutzer-ID>	CARMA (crastart)
5u	4u	3u	STCRSE und <Benutzer-ID>	CARMA (ispf)
-	1u (d)	2u	<Benutzer-ID>	z/OS UNIX-Build (Shellbefehle)
6u	1u	-	STCRSE und <Benutzer-ID>	z/OS UNIX-Shell
-	2u (c)	-	<Benutzer-ID>	File Manager
-	(5)	-	<Benutzer-ID>	SCLM Developer Toolkit
1u	-	-	STCRSE	Zeitgeber für Inaktivitätszeitlimit

Anmerkung:

- (a) Mindestens ein Adressraum der RSE-Thread-Pools ist aktiv. Die eigentliche Anzahl der Adressräume der RSE-Thread-Pools ist im Abschnitt „Anzahl der Adressräume“ auf Seite 217 angegeben.
- (b) Unter normalen Umständen und unter Verwendung der Standardkonfigurationsoptionen gibt es pro Benutzer einen aktiven ISPF Client Gateway. Die eigentliche Anzahl kann abweichen, wie im Abschnitt „Anzahl der Adressräume“ auf Seite 217 beschrieben.
- (c) Es gibt für jede Interaktion mit IBM File Manager einen benutzerspezifischen Prozess (mit der angegebenen Anzahl von Threads). Diese Prozesse enden, wenn das angeforderte Objekt nicht länger benötigt wird.
- Für SCLMDT ist ein Adressraum von ISPF Client Gateway erforderlich. SCLMDT nutzt den Adressraum gemeinsam mit TSO Commands Service.
- Abhängig von der ausgewählten Aktion kann SCLMDT mehrere Einzelthreadprozesse verwenden, die mit dem Abschluss der Task enden. In Tabelle 44 auf Seite 223 ist die maximale Anzahl von gleichzeitigen SCLMDT-Threads aufgelistet.

- Die meisten Aktionen, die mit MVS-Dateien verknüpft sind, verwenden TSO Commands Service, das in einer Transaktion mit ISPF Client Gateway beziehungsweise APPC aktiv sein kann.
- (d) Ein z/OS UNIX-Build ruft verschiedene Builddienstprogramme auf, die möglicherweise Multithreaddienstprogramme sind. In Tabelle 44 auf Seite 223 ist die minimale Anzahl von gleichzeitigen z/OS UNIX-Build-Threads angegeben.
- Alle aufgelisteten Threads bleiben solange aktiv, bis der zugehörige Prozess endet (wenn nicht anders angegeben).

Verwenden Sie die Formel in Abb. 52, um die maximale Anzahl der Threads zu berechnen, die von einem RSE-Thread-Pool verwendet werden. Verwenden Sie die Formel in Abb. 53, um die maximale Anzahl der Threads zu berechnen, die JES Job Monitor verwendet.

$$9 + N * (16 + x + y + z) + (20 + N * 0.1)$$

Abbildung 52. Maximale Anzahl von Threads in einem RSE-Thread-Pool

$$3 + N$$

Abbildung 53. Maximale Anzahl von Threads in einem RSE-Thread-Pool

Dabei

- stellt "N" die maximale Anzahl von gleichzeitigen Benutzern in diesem Thread-Pool oder in JES Job Monitor dar. Die Standardeinstellungen sind auf 60 Benutzer pro Thread-Pool festgelegt.
- entspricht "x" einem der folgenden Werte (abhängig von den ausgewählten Konfigurationsoptionen).

X	SCLMDT	TSO über Client-Gateway	TSO über APPC	Zeitlimit
0	Nein	Nein	Ja	Nein
0	Nein	Ja	Nein	Nein
0	Ja	Ja	Nein	Nein
1	Nein	Nein	Ja	Ja
1	Nein	Ja	Nein	Ja
1	Ja	Ja	Nein	Ja

- "y" entspricht einem der folgenden Werte (abhängig von den ausgewählten Konfigurationsoptionen).

Y	
0	Kein CARMA
1	CARMA (batch)
4	CARMA (crastart)
5	CARMA (ispf)

- "z" ist standardmäßig 0, kann sich jedoch abhängig von den Benutzeraktionen erhöhen.
 - Fügen Sie 6 hinzu, wenn eine z/OS UNIX-Shell geöffnet ist. Diese Threads bleiben solange aktiv, bis sich der Benutzer abmeldet.
- "20 + N*0.1" fügt einen Puffer für temporäre Threads hinzu. Die erforderliche Puffergröße kann an Ihrem Standort abweichen.

Die Definitionen in Tabelle 45 können die eigentliche Anzahl von Threads in einem Prozess begrenzen. Diese Option ist vor allem für die RSE-Thread-Pools wichtig.

Tabelle 45. Begrenzungen für Threads

Position	Begrenzung	Beeinträchtigte Ressourcen
BPXPRMxx	MAXTHREADS	Begrenzt die Anzahl von Threads in einem Prozess
BPXPRMxx	MAXTHREADTASKS	Begrenzt die Anzahl von MVS-Tasks in einem Prozess
BPXPRMxx	MAXASSIZE	Begrenzt die Größe des Adressraums und damit den verfügbaren Speicher für Thread-bezogene Steuerblöcke
rsed.envvars	Xmx	Legt die maximale Größe des Java-Heapspeichers fest. Dieser Speicher ist reserviert und deshalb nicht mehr für Thread-bezogene Steuerblöcke verfügbar.
rsed.envvars	maximum.clients	Begrenzt die Anzahl von Clients (und damit ihre Threads) in einem RSE-Thread-Pool
rsed.envvars	maximum.threads	Begrenzt die Anzahl von Client-Threads in einem RSE-Thread-Pool
FEJCNFG	MAX_THREADS	Begrenzt die Anzahl von Threads in JES Job Monitor

Anmerkung: Der Wert für maximum.threads in rsed.envvars muss kleiner als der Wert für MAXTHREADS und MAXTHREADTASKS in BPXPRMxx sein.

Speicherbelegung

RSE ist eine Java-Anwendung, die voraussetzt, dass bei der Planung der Speicherbelegung für Developer for System z zwei Speicherzuordnungsbegrenzungen berücksichtigt werden: für die Größe des Java-Heapspeichers und für die Größe der Adressräume.

Begrenzung für die Größe des Java-Heapspeichers

Java bietet viele Services für die einfache Durchführung von Codierungsaufgaben für Java-Anwendungen an. Einer dieser Services betrifft die Speicherverwaltung.

Die Speicherverwaltung von Java reserviert große Speicherblöcke und verwendet diese für Speicheranforderungen der Anwendung. Dieser von Java verwaltete Speicher wird als Java-Heapspeicher bezeichnet. Die periodische Garbage-Collection (Defragmentierung) gibt nicht verwendeten Speicherplatz im Heapspeicher wieder frei und reduziert die Größe des Heapspeichers.

Die maximale Größe des Java-Heapspeichers wird in rsed.envvars mit der Anweisung Xmx definiert. Wenn diese Anweisung nicht angegeben ist, verwendet Java eine Standardgröße von 64 MB.

Jeder RSE-Thread-Pool (der die Clientaktionen bedient) ist eine separate Java-Anwendung und verfügt deshalb über einen eigenen Java-Heapspeicher. Beachten Sie, dass alle Thread-Pools dieselbe Konfigurationsdatei `rsed.envvars` verwenden und deshalb dieselbe Begrenzung für die Größe des Java-Heapspeichers gilt.

Die Belegung des Java-Heapspeichers durch den Thread-Pool hängt stark von den Aktionen der verbundenen Clients ab. Eine regelmäßige Überwachung der Belegung des Heapspeichers ist erforderlich, um die optimale Begrenzung der Größe des Heapspeichers festlegen zu können. Verwenden Sie den Operatorbefehl **modify display process**, um die Belegung des Java-Heapspeichers durch die RSE-Thread-Pools zu überwachen.

Begrenzung für die Größe der Adressräume

Alle z/OS-Anwendungen, einschließlich Java-Anwendungen sind innerhalb eines Adressraums aktiv und deshalb an die Begrenzungen für die Adressraumgröße gebunden.

Die gewünschte Adressraumgröße wird während des Systemstarts angegeben, beispielsweise mit dem Parameter "REGION" in JCL. Systemeinstellungen können jedoch die eigentliche Adressraumgröße begrenzen. Lesen Sie den Abschnitt „Größe des Adressraums“ auf Seite 153, um mehr über diese Begrenzungen zu erfahren.

- MAXASSIZE in SYS1.PARMLIB(BPXPRMxx)
- ASSIZEMAX im OMVS-Segment der Benutzer-ID, die der gestarteten Task zugeordnet ist
- Systemexits IEFUSI und IEALIMIT

RSE-Thread-Pools übernehmen die Begrenzungen für die Adressraumgröße von dem RSE-Dämon. Die Adressraumgröße muss für den Java-Heapspeicher, für Java selbst, allgemeine Speicherbereiche und alle Steuerblöcke ausreichen, die das System zur Unterstützung der Thread-Pool-Aktivität erstellt, beispielsweise ein Tasksteuerblock (TBC) pro Thread. Beachten Sie, dass einige dieser Speicher nur 16 MB groß sind.

Sie sollten die eigentliche Adressraumgröße überwachen, bevor Sie Änderungen an Einstellungen vornehmen, die diese Größe beeinflussen. Dazu gehört beispielsweise das Ändern der Größe des Java-Heapspeichers oder das Ändern der Anzahl der Benutzer, die durch einen Einzel-Thread-Pool unterstützt werden. Verwenden Sie Ihre normale Systemüberwachungssoftware, um die eigentliche Speicherbelegung von Developer for System z zu verfolgen. Wenn Sie über kein zugeordnetes Überwachungstool verfügen, können Basisinformationen mit Tools wie der SDSF DA-Ansicht oder TASID (Systeminformationstool ohne Wartung (auf "as-is"-Basis), über die ISPF-Webseite "Support and downloads" verfügbar) zusammengestellt werden.

Richtlinien für Größenschätzungen

Wie oben beschrieben hängt die eigentliche Speicherbelegung von Developer for System z stark von der Benutzeraktivität ab. Einige Aktionen verwenden eine feste Speichergröße (beispielsweise die Anmeldung), während andere Aktionen einen variablen Speicherbedarf haben (zum Beispiel das Auflisten von Dateien mit einem angegebenen übergeordneten Qualifikationsmerkmal).

- Verwenden Sie für RSE einen Adressraum mit 2 GB, um ausreichend Raum für den Java-Heapspeicher und alle Systemsteuerungsblöcke zu haben.

- Die Beispielkonfiguration `rsed.envvars` ist auf 60 Benutzer pro Thread-Pool festgelegt.
 - `maximum.clients=60`
 - `maximum.threads=1000` ($10+16*60 = 970$, sodass 1000 61 Benutzer zulässt)
- Die Beispielkonfiguration `rsed.envvars` lässt eine Java-Heapspeichergröße von 256 MB zu. Damit können 60 Clients mit einem Durchschnitt von 4 MB Speicher verwendet werden ($60*4 = 240$).

Beachten Sie, dass RSE die aktuelle Größe des Java-Heapspeichers und des Adressraums während des Systemstarts in der Konsolnachricht 'FEK004I' anzeigt.

Durchlaufen Sie eins der folgenden Szenarien, wenn die Überwachung ergibt, dass die aktuelle Größe des Java-Heapspeichers für die aktuelle Auslastung nicht ausreicht:

- Erhöhen Sie die maximale Größe des Java-Heapspeichers mithilfe der Anweisung `Xmx` in `rsed.envvars`. Stellen Sie zuvor sicher, dass der Adressraum für die Vergrößerung ausreicht.
- Verkleinern Sie die maximale Anzahl von Clients pro Thread-Pool mithilfe der Anweisung `maximum.clients` in `rsed.envvars`. RSE unterstützt immer noch dieselbe Anzahl von Clients; diese werden jedoch auf mehrere Thread-Pools verteilt.

Beispielanalyse der Speicherbelegung

Die Anzeigen in den folgenden Abbildungen zeigen einige Beispielzahlen für die Ressourcennutzung einer Standardkonfiguration von Developer for System z mit einer Änderung. Die maximale Größe des Java-Heapspeichers ist auf 10 MB gesetzt. Ein kleiner Maximalwert führt zu einer größeren prozentualen Nutzung und die Größenbegrenzung des Heapspeichers wird früher erreicht.

Max Heap Size=10MB and private AS Size=1,959MB

startup

BPXM023I (STCRSE)
ProcessId(268) Memory Usage(7%) Clients(0)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2740	72
LOCKD	1.60	28.7M	14183
RSED	4.47	32.8M	15910
RSED8	1.15	27.4M	12612

logon 1

BPXM023I (STCRSE)
ProcessId(268) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2864	81
LOCKD	1.64	28.8M	14259
RSED	4.55	32.8M	15980
RSED8	3.72	55.9M	24128

logon 2

BPXM023I (STCRSE)
ProcessId(268) Memory Usage(23%) Clients(2)

Jobname	Cpu time	Storage	EXCP
JMON	0.02	2944	86
LOCKD	1.66	28.9M	14268
RSED	4.58	32.9M	16027
RSED8	4.20	57.8M	25205

logon 3

BPXM023I (STCRSE)
ProcessId(268) Memory Usage(37%) Clients(3)

Jobname	Cpu time	Storage	EXCP
JMON	0.02	3020	91
LOCKD	1.67	29.0M	14277
RSED	4.60	32.9M	16076
RSED8	4.51	59.6M	26327

logon 4

BPXM023I (STCRSE)
ProcessId(268) Memory Usage(41%) Clients(4)

Jobname	Cpu time	Storage	EXCP
JMON	0.02	3108	96
LOCKD	1.68	29.0M	14286
RSED	4.61	32.9M	16125
RSED8	4.77	62.3M	27404

Abbildung 54. Ressourcennutzung mit 5 Anmeldungen

logon 5

```
BPXM023I (STCRSE)
ProcessId(268      ) Memory Usage(41%) Clients(4)
ProcessId(33554706) Memory Usage(13%) Clients(1)
```

Jobname	Cpu time	Storage	EXCP
JMON	0.03	3184	101
LOCKD	1.69	29.1M	14295
RSED	4.64	32.9M	16229
RSED8	4.78	62.4M	27413
RSED9	4.60	56.6M	24065

Abbildung 55. Ressourcennutzung mit 5 Anmeldungen (Fortsetzung)

Abb. 54 auf Seite 229 und Abb. 55 zeigen ein Szenario, in dem sich 5 Clients bei einem RSE-Dämon mit einem 10-MB-Java-Heapspeicher anmelden.

- Ein Thread-Pool (RSED8) befindet sich beim Start im Ruhezustand. Er verwendet ungefähr 27 MB. Davon befinden sich 0,7 MB im Java-Heapspeicher (7% von 10 MB).
- Der Thread-Pool wird aktiv, wenn der erste Client eine Verbindung herstellt. Dabei werden zusätzlich 27 MB verwendet, plus 2 MB für jeden Client, der eine Verbindung herstellt.
- Ein Teil der 2 MB pro Verbindung befindet sich ebenfalls im Java-Heapspeicher. Dies wird durch die Zunahme der Heapspeicherbelegung deutlich.
- Es gibt allerdings kein echtes Muster für die Heapspeicherbelegung, weil sie von Java-Mechanismen abhängt, die die erforderliche Speichermenge schätzen und mehr als nötig zuordnen. Durch eine regelmäßige Garbage-Collection wird Speicher freigegeben, wodurch Trends noch schwerer zu erfassen sind.
- Interne Mechanismen, die die Anzahl der Verbindungen pro Thread-Pool begrenzen, um eine ausreichende Größe des Heapspeichers für aktive Threads sicherzustellen, führen dazu, dass die fünfte Verbindung in einem neuen Thread-Pool (RSED9) erstellt wird. Diese internen Sicherheitsmaßnahmen werden bei einer ordnungsgemäßen Konfiguration gewöhnlich nicht aufgerufen, weil andere Grenzwerte (am wahrscheinlichsten `maximum.clients` in `rsed.envvars`) zuerst erreicht würden.

Max Heap Size=10MB and private AS Size=1,959MB

startup

BPXM023I (STCRSE)
ProcessId(212) Memory Usage(7%) Clients(0)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2736	71
LOCKD	1.73	30.5M	14179
RSED	4.35	32.9M	15117
RSED8	1.43	27.4M	12609

logon

BPXM023I (STCRSE)
ProcessId(212) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2864	80
LOCKD	1.76	30.6M	14255
RSED	4.48	33.0M	15187
RSED8	3.53	53.9M	24125

expand large MVS tree (195 data sets)

BPXM023I (STCRSE)
ProcessId(212) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2864	80
LOCKD	1.78	30.6M	14255
RSED	4.58	33.1M	16094
RSED8	4.28	56.1M	24740

expand small PDS (21 members)

BPXM023I (STCRSE)
ProcessId(212) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
IBMUSER2	0.22	2644	870
JMON	0.01	2864	80
LOCKD	1.78	30.6M	14255
RSED	4.61	33.1M	16108
RSED8	4.40	56.2M	24937

open medium sized member (86 lines)

BPXM023I (STCRSE)
ProcessId(212) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
IBMUSER2	0.22	2644	870
JMON	0.01	2864	80
RSED	4.61	33.1M	16108
RSED8	8.12	62.7M	27044

Abbildung 56. Ressourcennutzung beim Bearbeiten eines Members der untergliederten Datei

Abb. 56 zeigt ein Szenario, in dem sich 1 Client bei einem RSE-Dämon mit einem 10-MB-Java-Heapspeicher anmeldet und ein Member der untergliederten Datei bearbeitet.

- Die Katalogsuche mit 195 Dateinamen als Ergebnis hat ungefähr 2 MB Speicher belegt. Dieser bezieht sich auf Systemaktivitäten, weil sich die Belegung des Java-Heapspeichers nicht erhöht.
- Das Öffnen einer untergliederten Datei mit 21 Members belegt kaum Speicher im Thread-Pool, aber die Anzeige gibt an, dass TSO Commands Services aufgerufen wurden. Ein neuer Adressraum (IBMUSER2) ist aktiv, der die Regionsgröße verwendet, die dieser Benutzer-ID in TSO zugeordnet wurde. Dieser Adressraum bleibt für eine bestimmte Zeitspanne aktiv. Er kann also für zukünftige Anforderungen durch TSO Commands Service wiederverwendet werden.
- Das Öffnen eines Members zeigt ähnliche Zahlen wie das Erweitern eines übergeordneten Qualifikationsmerkmals. Die Belegung des Java-Heapspeichers bleibt gleich. Es gibt allerdings eine Speicherzunahme von 6,5 MB aufgrund von Systemaktivitäten.

Speicherbelegung im z/OS UNIX-Dateisystem

Die meisten Daten mit Bezug auf Developer for System z, die nicht in eine DD-Anweisung geschrieben werden, werden in einer z/OS UNIX-Datei gespeichert. Der Systemprogrammierer steuert, welche Daten an welcher Position gespeichert werden. Er hat allerdings keine Kontrolle über die gespeicherte Datenmenge.

Die Daten können in folgende Kategorien eingeteilt werden:

- Fehleranalyse (Protokoll- und Systemspeicherauszugsdateien), für die viele Details in Kapitel 9, „Konfigurationsprobleme lösen“, auf Seite 137 dokumentiert werden
- Protokollierung, die im Abschnitt „Prüfprotokollierung“ auf Seite 165 dokumentiert ist
- Temporäre Daten

Wie in Kapitel 9, „Konfigurationsprobleme lösen“, auf Seite 137 dokumentiert, speichert Developer for System z die RSE-bezogenen Hostprotokolle in den folgenden z/OS UNIX-Verzeichnissen:

- /var/rdz/logs für Protokolle der gestarteten RSE-Task
- /var/rdz/logs/\$LOGNAME für Benutzerprotokolle

Standardmäßig werden nur Fehlernachrichten und Warnungen in den Protokollen gespeichert. Bei einem ordnungsgemäßen Betrieb sollten diese Verzeichnisse also nur leere oder beinahe leere Dateien enthalten. (Prüfprotokolle werden dabei nicht berücksichtigt.)

Sie können das Protokollieren von Informationsnachrichten aktivieren (am besten nur auf Anweisung des IBM Support Center), wodurch die Größe der Protokolldateien deutlich zunimmt.

startup

```
$ ls -l /var/rdz/logs
total 144
-rw-rw-rw- 1 STCRSE STCGRP 33642 Jul 10 12:10 rsedaemon.log
-rw-rw-rw- 1 STCRSE STCGRP 1442 Jul 10 12:10 rseserver.log
```

logon

```
$ ls -l /var/rdz/logs
total 144
drwxrwxrwx 3 IBMUSER SYS1 8192 Jul 10 12:11 IBMUSER
-rw-rw-rw- 1 STCRSE STCGRP 36655 Jul 10 12:11 rsedaemon.log
-rw-rw-rw- 1 STCRSE STCGRP 1893 Jul 10 12:11 rseserver.log
$ ls -l /var/rdz/logs/IBMUSER
total 160
-rw-rw-rw- 1 IBMUSER SYS1 3459 Jul 10 12:11 ffs.log
-rw-rw-rw- 1 IBMUSER SYS1 0 Jul 10 12:11 ffsget.log
-rw-rw-rw- 1 IBMUSER SYS1 0 Jul 10 12:11 ffsput.log
-rw-rw-rw- 1 IBMUSER SYS1 303 Jul 10 12:11 lock.log
-rw-rw-rw- 1 IBMUSER SYS1 126 Jul 10 12:11 rmt_classloader_cache.jar
-rw-rw-rw- 1 IBMUSER SYS1 7266 Jul 10 12:11 rsecomm.log
-rw-rw-rw- 1 IBMUSER SYS1 0 Jul 10 12:11 stderr.log
-rw-rw-rw- 1 IBMUSER SYS1 0 Jul 10 12:11 stdout.log
```

logoff

```
$ ls -l /var/rdz/logs
total 80
drwxrwxrwx 3 IBMUSER SYS1 8192 Jul 10 12:11 IBMUSER
-rw-rw-rw- 1 STCRSE STCGRP 36655 Jul 10 12:11 rsedaemon.log
-rw-rw-rw- 1 STCRSE STCGRP 2208 Jul 10 12:11 rseserver.log
$ ls -l /var/rdz/logs/IBMUSER
total 296
-rw-rw-rw- 1 IBMUSER SYS1 6393 Jul 10 12:11 ffs.log
-rw-rw-rw- 1 IBMUSER SYS1 0 Jul 10 12:11 ffsget.log
-rw-rw-rw- 1 IBMUSER SYS1 0 Jul 10 12:11 ffsput.log
-rw-rw-rw- 1 IBMUSER SYS1 609 Jul 10 12:11 lock.log
-rw-rw-rw- 1 IBMUSER SYS1 126 Jul 10 12:11 rmt_classloader_cache.jar
-rw-rw-rw- 1 IBMUSER SYS1 45157 Jul 10 12:11 rsecomm.log
-rw-rw-rw- 1 IBMUSER SYS1 0 Jul 10 12:11 stderr.log
-rw-rw-rw- 1 IBMUSER SYS1 176 Jul 10 12:11 stdout.log
```

stop

```
$ ls -l /var/rdz/logs
total 80
drwxrwxrwx 3 IBMUSER SYS1 8192 Jul 10 12:11 IBMUSER
-rw-rw-rw- 1 STCRSE STCGRP 36655 Jul 10 12:11 rsedaemon.log
-rw-rw-rw- 1 STCRSE STCGRP 2490 Jul 10 12:12 rseserver.log
```

Abbildung 57. Speicherbelegung im z/OS UNIX-Dateisystem

Abb. 57 zeigt die minimale Speicherbelegung des z/OS UNIX-Dateisystems bei der Verwendung von Debugstufe 2 (Informationsnachrichten).

- Die Protokolle der gestarteten Task belegen nach dem Systemstart 34 KB. Sie werden langsam größer, wenn sich Benutzer an- bzw. abmelden oder wenn Bedienerbefehle abgesetzt werden.
- Ein Clientprotokollverzeichnis belegt nach der Anmeldung 11 KB. Es wird langsam größer, wenn der Benutzer mit der Arbeit beginnt. (Dies wird im Beispiel nicht gezeigt.)
- Durch die Abmeldung werden den Benutzerprotokollen weitere 40 KB hinzugefügt. Dies ergibt dann 51 KB.

Mit Ausnahme von Prüfprotokollen werden Protokolldateien bei jedem Neustart (bei der gestarteten RSE-Task) oder bei jeder Anmeldung (bei einem Client) überschrieben. Dadurch wird die Gesamtgröße begrenzt. Durch die Anweisung `keep.last.log` in `rsed.envvars` wird dies geringfügig geändert. Aufgrund dieser Anweisung kann RSE eine Kopie der vorherigen Protokolle beibehalten. Ältere Kopien werden immer gelöscht.

Wenn im Dateisystem mit den Prüfprotokolldateien nur noch ein kleiner freier Speicherbereich verfügbar ist, wird eine Warnung an die Konsole gesendet, sofern die Protokollierung aktiv ist. Diese Konsolnachricht (FEK103E) wird immer wieder angezeigt, bis das Speicherproblem gelöst ist. Eine Liste der von RSE generierten Konsolnachrichten finden Sie im Abschnitt „Konsolnachrichten“ auf Seite 132.

Die Definitionen in Tabelle 46 steuern, welche Daten in den Protokollverzeichnissen gespeichert werden und wo sich diese Verzeichnisse befinden.

Tabelle 46. Anweisungen für die Protokollausgabe

Position	Anweisung	Funktion
resecmm.properties	debug_level	Standardprotokolldetailstufe festlegen
rsed.envvars	keep.last.log	Eine Kopie der vorherigen Protokolle vor Start/Anmeldung beibehalten
rsed.envvars	enable.audit.log	Prüftrace der Clientaktionen beibehalten
rsed.envvars	enable.standard.log	Datenströme 'stdout' und 'stderr' des (bzw. der) Thread-Pools in eine Protokolldatei schreiben
rsed.envvars	DSTORE_TRACING_ON	DataStore-Aktionsprotokollierung aktivieren
rsed.envvars	DSTORE_MEMLOGGING_ON	DataStore-Protokollierung der Speicherbelegung aktivieren
Bedienerbefehl	modify rsecommlog <Stufe>	Die Protokolldetailstufe von rsecomm.log dynamisch ändern
Bedienerbefehl	modify rsedaemonlog <Stufe>	Die Protokolldetailstufe von rsedaemon.log dynamisch ändern
Bedienerbefehl	modify rseserverlog <Stufe>	Die Protokolldetailstufe von rseserver.log dynamisch ändern
Bedienerbefehl	modify rsestandardlog {on off}	Die Aktualisierung von std*.*.log dynamisch ändern
rsed.envvars	daemon.log	Ausgangspfad für gestartete RSE-Task und Prüfprotokolle
rsed.envvars	user.log	Ausgangspfad für Benutzerprotokolle

Zusammen mit vorausgesetzter Software wie dem ISPF-Client-Gateway schreibt Developer for System z auch temporäre Daten in `/tmp` und `/var/rdz/WORKAREA`. Das hier geschriebene Datenvolumen ist unvorhersehbar. In den Dateisystemen, in denen sich diese Verzeichnisse befinden, sollten Sie daher ausreichend freien Speicherbereich haben.

Developer for System z versucht immer, diese temporären Dateien zu bereinigen. Eine manuelle Bereinigung, wie im Abschnitt „WORKAREA-Bereinigung (optional)“ auf Seite 108 dokumentiert, kann aber jederzeit durchgeführt werden.

Definitionen von wichtigen Ressourcen

/etc/rdz/rsed.envvars

Die in rsed.envvars definierten Umgebungsvariablen werden von RSE, Java und z/OS UNIX verwendet. Die mit Developer for System z gelieferte Beispieldatei ist für kleine bis mittlere Installationen gedacht, die keine optionalen Komponenten von Developer for System z benötigen. Im Abschnitt „RSE-Konfigurationsdatei rsed.envvars“ auf Seite 33 werden alle Variablen beschrieben, die in der Beispieldatei definiert sind. Bei den folgenden Variablen müssen Sie allerdings besonders vorsichtig sein:

`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx256m"`

Festlegen der anfänglichen Heapgröße (Xms) und der maximalen Heapgröße (Xmx). Die Standardwerte sind jeweils 128M und 256M. Ändern Sie diese, um die gewünschten Werte der Heapgröße zu erzwingen. Wenn diese Anweisung in Kommentarzeichen gesetzt ist, werden die Java-Standardwerte verwendet. Diese sind 4M beziehungsweise 512M (1M und 64M für Java 5.0).

`#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dmaximum.clients=60"`

Maximale Anzahl der Clients, die ein Thread-Pool bedienen kann. Die Standardeinstellung ist 60. Entfernen Sie das Kommentarzeichen und passen Sie die Option an, um die Anzahl der Clients pro Thread-Pool zu begrenzen. Beachten Sie, dass andere Grenzwerte möglicherweise verhindern, dass RSE diese Begrenzung erreicht.

`#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dmaximum.threads=1000"`

Maximale Anzahl von aktiven Threads in einem Thread-Pool, um neue Clients zuzulassen. Die Standardeinstellung ist 1000. Entfernen Sie das Kommentarzeichen und passen Sie den Wert an, um die Anzahl der Clients pro Thread-Pool auf Basis der Threads in Gebrauch zu begrenzen. Beachten Sie, dass jede Clientverbindung mehrere Threads (16 oder mehr) verwendet und dass andere Grenzwerte verhindern können, dass RSE diese Begrenzung erreicht.

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

`#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dminimum.threadpool.process=10"`

Minimale Anzahl aktiver Thread-Pools. Die Standardeinstellung ist 1. Entfernen Sie das Kommentarzeichen und passen Sie die Option an, damit mindestens die angegebene Anzahl von Thread-Pool-Prozessen gestartet wird. Thread-Pool-Prozesse werden für die Lastverteilung der RSE-Server-Threads verwendet. Weitere neue Prozesse werden bei Bedarf gestartet. Wenn die neuen Prozesse vorab gestartet werden, werden Verzögerungen bei Verbindungen verhindert. Das System verwendet in Leerlaufzeiten allerdings mehr Ressourcen.

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

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

SYS1.PARMLIB(BPXPRMxx)

RSE ist eine Java-Anwendung, das heißt, sie ist in der z/OS UNIX-Umgebung aktiv. Auf diese Weise wird BPXPRMxx ein entscheidendes PARMLIB-Member, weil es die Parameter enthält, mit denen die z/OS UNIX-Umgebung und die Dateisysteme gesteuert werden. BPXPRMxx wird in *MVS Initialization and Tuning Reference* (IBM Form SA22-7592) beschrieben. Die folgenden Anweisungen wirken sich auf Developer for System z aus:

MAXPROCSYS(nnnnn)

Gibt die maximal zulässige Anzahl von Prozessen im System an.

Wertebereich: 'nnnnn' ist ein Dezimalwert zwischen 5 und 32767.
Standardwert: 900

MAXPROCUSER(nnnnn)

Gibt die maximale Anzahl von Prozessen an, die für eine einzelne z/OS UNIX-Benutzer-ID gleichzeitig aktiv sein dürfen. Dabei spielt es keine Rolle, wie die Prozesse erstellt wurden.

Wertebereich: 'nnnnn' ist ein Dezimalwert zwischen 3 und 32767.
Standardwert: 25

Anmerkung:

- Alle RSE-Prozesse verwenden dieselbe z/OS UNIX-Benutzer-ID (die des Benutzers, der dem RSE-Dämon zugeordnet ist), weil alle Clients als Threads in den RSE-Prozessen ausgeführt werden.
- Dieser Wert kann auch mit der Variablen PROCUSERMAX im OMVS-Sicherheitsprofilsegment des Benutzers festgelegt werden, der der gestarteten RSED-Task zugeordnet ist.

MAXTHREADS(nnnnnn)

Gibt die maximale Anzahl von pthread_created-Threads (einschließlich aktiver Threads, Threads in der Warteschlange und beendeter, aber nicht freigegebener Threads) an, die für einen einzelnen Prozess gleichzeitig aktiv sein können. Wenn Sie den Wert '0' angeben, können Anwendungen 'pthread_create' nicht verwenden.

Wertebereich: 'nnnnnn' ist ein Dezimalwert zwischen 0 und 100.000.
Standardwert: 200

Anmerkung:

- Jeder Client verwendet mindestens 16 Threads im RSE-Thread-Pool-Prozess. Im Prozess sind mehrere Clients aktiv.
- Dieser Wert kann auch mit der Variablen THREADSMAX im OMVS-Sicherheitsprofilsegment des Benutzers festgelegt werden, der der gestarteten RSED-Task zugeordnet ist. Sofern er festgelegt ist, wird der Wert von THREADSMAX für MAXTHREADS und MAXTHREADTASKS verwendet.

MAXTHREADTASKS(nnnnn)

Gibt die maximale Anzahl der MVS-Tasks an, die für einen einzelnen Prozess für pthread_created-Threads gleichzeitig aktiv sein können.

Wertebereich: 'nnnnn' ist ein Dezimalwert zwischen 0 und 32768.
Standardwert: 1000

Anmerkung:

- Für jeden aktiven Thread gibt es eine MVS-Task (TCB, Task Control Block).
- Für jede gleichzeitig ablaufende MVS-Task ist zusätzlicher Speicher erforderlich. Ein Teil davon muss unter der 16-MB-Grenze liegen.
- Jeder Client verwendet mindestens 16 Threads im RSE-Thread-Pool-Prozess. Im Prozess sind mehrere Clients aktiv.
- Dieser Wert kann auch mit der Variablen THREADSMAX im OMVS-Sicherheitsprofilsegment des Benutzers festgelegt werden, der der gestarteten RSED-Task zugeordnet ist. Sofern er festgelegt ist, wird der Wert von THREADSMAX für MAXTHREADS und MAXTHREADTASKS verwendet.

MAXUIDS(nnnnn)

Gibt die maximale Anzahl von z/OS UNIX-Benutzer-IDs (UIDs) an, die gleichzeitig arbeiten können.

Wertebereich: 'nnnnn' ist ein Dezimalwert zwischen 1 und 32767.

Standardwert: 200

MAXASSIZE(nnnnn)

Gibt die Ressourcenwerte für RLIMIT_AS an, die als Anfangswerte für neue Prozesse festgelegt werden. RLIMIT_AS gibt die Regionsgröße des Adressraums an.

Wertebereich: 'nnnnn' ist ein Dezimalwert zwischen 10.485.760 (10 Megabyte) und 2.147.483.647 (2 Gigabyte).

Standardwert: 209.715.200 (200 Megabyte)

Anmerkung:

- Dieser Wert sollte auf 2 GB gesetzt werden.
- Dieser Wert kann auch mit der Variablen ASSIZEMAX im OMVS-Sicherheitsprofilsegment des Benutzers festgelegt werden, der der gestarteten RSED-Task zugeordnet ist.

MAXFILEPROC(nnnnnn)

Gibt die maximale Anzahl der Deskriptoren für Dateien, Sockets, Verzeichnisse und andere Dateisystemobjekte an, die für einen einzelnen Prozess gleichzeitig aktiv oder zugeordnet sein können.

Wertebereich: 'nnnnnn' ist ein Dezimalwert zwischen 3 und 524.287.

Standardwert: 64000

Anmerkung:

- In einem Thread-Pool befinden sich alle zugehörigen Client-Threads in einem einzigen Prozess.
- Dieser Wert kann auch mit der Variablen FILEPROCMAX im OMVS-Sicherheitsprofilsegment des Benutzers festgelegt werden, der der gestarteten RSED-Task zugeordnet ist.

MAXMMAPAREA(nnnnn)

Gibt den Maximalwert für den Speicherbereich (in Seiten) im Datenraum an,

der für Speicherzuordnungen von z/OS UNIX-Dateien zugewiesen werden kann. Der Speicher wird erst zugewiesen, wenn die Speicherzuordnung aktiv ist.

Wertebereich: 'nnnnn' ist ein Dezimalwert zwischen 1 und 16.777.216.
Standardwert: 40960

Anmerkung: Dieser Wert kann auch mit der Variablen MMAPAREAMAX im OMVS-Sicherheitsprofilsegment des Benutzers festgelegt werden, der der gestarteten RSED-Task zugeordnet ist.

Verwenden Sie den Bedienerbefehl **SETOMVS** oder **SET OMVS**, um den Wert einer beliebigen genannten BPXPRMxx-Variablen dynamisch (bis zum nächsten einleitenden Programmladen) zu erhöhen oder zu verringern. Wenn Sie eine permanente Änderung wünschen, bearbeiten Sie das BPXPRMxx-Member, das für IPLs verwendet wird. Weitere Informationen zu diesen Bedienerbefehlen enthält die Veröffentlichung *MVS System Commands* (IBM Form SA22-7627).

Die folgenden Definitionen sind Subparameter der Anweisung NETWORK.

MAXSOCKETS(nnnnnnnnn)

Gibt die maximale Anzahl von Sockets an, die von diesem Dateisystem für diese Adressfamilie unterstützt werden. Dieser Parameter ist optional.

Wertebereich: 'nnnnnnnn' ist ein Dezimalwert zwischen 0 und 16.777.215.
Standardwert: 100

INADDRANYCOUNT(nnnn)

INADDRANYCOUNT: Gibt die Anzahl der vom System reservierten Ports für die Bindung an PORT 0 mit INADDR_ANY an, einschließlich des mit dem Parameter INADDRANYPORT angegebenen Ports. Dieser Wert wird nur für CINET (mehrere TCP/IP-Stacks) benötigt.

Wertebereich: 'nnnn' ist ein Dezimalwert zwischen 1 und 4000.
Standardwert: Wenn weder INADDRANYPORT noch INADDRANYCOUNT angegeben wurde, ist der Standardwert für INADDRANYCOUNT 1000.
Andernfalls werden keine (0) Ports reserviert.

Definitionen von verschiedenen Ressourcen

EXEC-Karte in der Server-JCL

Die folgenden Definitionen sollten der EXEC-Karte in der JCL des Servers für Developer for System z hinzugefügt werden.

REGION=0M

REGION=0M wird für die gestarteten Tasks des RSE-Dämons und von JES Job Monitor (RSED bzw. JMON) empfohlen. Aufgrund dieser Angabe ist die Größe des Adressraums nur durch den verfügbaren privaten Speicher oder durch die Systemexits IEFUSI oder IEALIMIT begrenzt. Beachten Sie, dass IBM dringend empfiehlt, diese Exits nicht für z/OS UNIX-Adressräume zu verwenden, wie den RSE-Dämon.

TIME=NOLIMIT

TIME=NOLIMIT wird zur Verwendung mit allen Servern für Developer for System z empfohlen. Der Grund ist, dass die CPU-Zeiten aller Clients für Developer for System z in den Serveradressräumen zusammengefasst werden.

FEK.#CUST.PARMLIB(FEJJCNFG)

Die in FEJJCNFG definierten Umgebungsvariablen werden von JES Job Monitor verwendet. Die mit Developer for System z gelieferte Beispieldatei ist für kleine bis mittlere Installationen gedacht. Im Abschnitt „Konfigurationsdatei für JES Job Monitor (FEJJCNFG)“ auf Seite 29 werden alle Variablen beschrieben, die in der Beispieldatei definiert sind. Bei den folgenden Variablen müssen Sie allerdings besonders vorsichtig sein:

MAX_THREADS

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

SYS1.PARMLIB(IEASYSxx)

IEASYSxx wird in *MVS Initialization and Tuning Reference* (IBM Form SA22-7592) beschrieben. Die folgenden Anweisungen wirken sich auf Developer for System z aus:

MAXUSER=nnnnn

Dieser Parameter gibt einen Wert an, den das System in den meisten Fällen verwendet, um die Anzahl der Jobs und gestarteten Tasks zu begrenzen, die während eines bestimmten einleitenden Programmladens gleichzeitig ausgeführt werden können.

Wertebereich: 'nnnnn' ist ein Dezimalwert von 0-32767. Beachten Sie, dass die Summe der für die Systemparameter MAXUSER, RSVSTRT und RSVNONR angegebenen Werte 32767 nicht übersteigen kann.

Standardwert: 255

SYS1.PARMLIB(IVTPRMxx)

IVTPRMxx legt Parameter für den Kommunikationsspeichermanager (Communication Storage Manager, CSM) fest und wird in *MVS Initialization and Tuning Reference* (IBM Form SA22-7592) beschrieben. Die folgenden Anweisungen wirken sich auf Developer for System z aus:

FIXED MAX(maxfix)

Definiert den maximalen Speicherbereich, der festgelegten CSM-Puffern zugeordnet wird.

Wertebereich: 'maxfix' ist ein Wert zwischen 1024K und 2048M.

Standardwert: 100M

ECSA MAX(maxecsa)

Definiert den maximalen Speicherbereich, der ECSA-CSM-Puffern zugeordnet wird.

Wertebereich: 'maxecsa' ist ein Wert zwischen 1024K und 2048M.

Standardwert: 100M

SYS1.PARMLIB(ASCHPMxx)

Das PARMLIB-Member ASCHPMxx enthält Planungsinformationen für den Transaktionsscheduler ASCH und wird in *MVS Initialization and Tuning Reference* (IBM Form SA22-7592) beschrieben. Die folgenden Anweisungen wirken sich auf Developer for System z aus:

MAX(nnnnn)

Ein optionaler Parameter der CLASSADD-Definition, der die maximale Anzahl von APPC-Transaktionsinitiatoren angibt, die für eine bestimmte Klasse von Transaktionsinitiatoren zulässig sind. Nachdem dieser Grenzwert erreicht wurde, werden keine neuen Adressräume erstellt. Eingehende Anforderungen werden in die Warteschlange gestellt und warten darauf, dass vorhandene Initiatoradressräume verfügbar werden. Der Wert sollte die maximal zulässige Anzahl von Adressräumen für Ihre Installation nicht überschreiten. Sie sollten auch an konkurrierende Produkte auf dem System denken, die ebenfalls Adressräume benötigen.

Wertebereich: 'nnnnn' ist ein Dezimalwert zwischen 1 und 64000.
Standardwert: 1

Anmerkung: Wenn Sie TSO Commands Service über APPC starten, muss die verwendete Transaktionsklasse genug Transaktionsinitiatoren haben, damit für jeden gleichzeitig angemeldeten Benutzer von Developer for System z ein Initiator verfügbar ist.

Überwachung

Da sich der Bedarf an Systemressourcen durch die Benutzerarbeitslast ändern kann, sollte das System regelmäßig überwacht werden, um die Ressourcennutzung zu messen. So können Rational Developer for System z und die Systemkonfiguration entsprechend Ihren Benutzeranforderungen angepasst werden. Als Unterstützung bei diesem Überwachungsprozess können die folgenden Befehle verwendet werden.

RSE überwachen

Benutzeraktivitäten in Developer for System z finden hauptsächlich in RSE-Thread-Pools statt. Zur optimalen Verwendung benötigen die Pools daher eine Überwachung. Zum RSE-Dämon können Informationen abgerufen werden, die nicht mit üblichen Systemüberwachungstools zusammengestellt werden können.

- Über Ihre üblichen Systemüberwachungstools, wie RMF, können Sie spezifische Daten zu Adressräumen zusammenstellen, wie verwendeter Realspeicher und CPU-Zeit. Wenn Sie kein Überwachungstool zugeordnet haben, können Basisinformationen mit Tools wie der SDSF-DA-Ansicht oder TASID (ein Systeminformationstool ohne Wartung (auf "as-is"-Basis), das über die ISPF-Webseite "Support and downloads" verfügbar ist) zusammengestellt werden.
- Während des Systemstarts gibt der RSE-Dämon die verfügbare Größe des Adressraums und des Java-Heapspeichers mit der Konsolnachricht 'FEK004I' aus.
FEK004I RseDaemon: Max Heap Size=65MB and private AS Size=1,959MB
- Mit dem Bedienerbefehl **MODIFY RSED,APPL=DISPLAY PROCESS** werden die RSE-Thread-Pool-Prozesse angezeigt. Im Feld "Memory Usage" wird angezeigt, wie viel des definierten Java-Heapspeichers tatsächlich verwendet wird. Weitere Informationen zu diesem Befehl enthält Kapitel 8, „Bedienerbefehle“, auf Seite 125.

```
f rsed,appl=d p
BPXM023I (STCRSE)
ProcessId(16777456) Memory Usage(33%) Clients(4) Order(1)
```

Es werden weitere Informationen bereitgestellt, wenn die Option "DETAIL" des Änderungsbefehls **DISPLAY PROCESS** verwendet wird:

```
f rsed,appl=d p,detail
BPXM023I (STCRSE)
ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
```


PROCESS LIMITS:	CURRENT	HIGHWATER	LIMIT
JAVA HEAP USAGE(%)	10	56	100
CLIENTS	0	25	60
MAXFILEPROC	83	103	64000
MAXPROCUSER	97	99	200
MAXTHREADS	9	14	1500
MAXTHREADDTASKS	9	14	1500

z/OS UNIX überwachen

Die meisten z/OS UNIX-Begrenzungen, die für Developer for System z wichtig sind, können mithilfe von Bedienerbefehlen angezeigt werden. Mit einigen Befehlen wird sogar die aktuelle Verwendung und die obere Grenze für einen bestimmten Grenzwert angezeigt. Weitere Informationen zu diesen Befehlen enthält die Veröffentlichung *MVS System Commands* (IBM Form SA22-7627).

- Mit der Anweisung LIMMSG(ALL) in SYS1.PARMLIB(BPXPRMxx) zeigt z/OS UNIX Konsolnachrichten (BPXI040I) an, wenn ein beliebiger PARMLIB-Grenzwert annähernd überschritten wird. Der Standardwert für LIMMSG ist NONE. Damit wird die Funktion inaktiviert. Verwenden Sie den Bedienerbefehl **SETOMVS LIMMSG=ALL**, um diese Funktion dynamisch (bis zum nächsten einleitenden Programmladen) zu aktivieren. Weitere Informationen zu dieser Anweisung enthält die Veröffentlichung *MVS Initialization and Tuning Reference* (IBM Form SA22-7592).
- Mit dem Bedienerbefehl **DISPLAY OMVS,OPTIONS** werden die aktuellen Werte von z/OS UNIX-Anweisungen angezeigt, die dynamisch festgelegt werden können.

```
d omvs,o
BPX0043I 13.10.16 DISPLAY OMVS 066
OMVS 000D ETC/INIT WAIT OMVS=(M7)
CURRENT UNIX CONFIGURATION SETTINGS:
MAXPROCSYS      =      256    MAXPROCUSER      =      16
MAXFILEPROC     =      256    MAXFILESIZE      = NOLIMIT
MAXCPUPTIME     =      1000    MAXUIDS        =      200
MAXPTYS         =      256
MAXMMAPAREA     =      256    MAXASSIZE       = 209715200
MAXTHREADS      =      200    MAXTHREADTASKS =      1000
MAXCORESIZE     = 4194304    MAXSHAREPAGES =      4096
IPCMSGQBYTES    = 2147483647 IPCMSGQMNUM    =     10000
IPCMSGNIDS      =      500    IPCSEMNIDS    =      500
IPCSEMNOPS      =      25     IPCSEMNSEMS   =     1000
IPCshmMPAGES    =     25600    IPCshMNIDS    =      500
IPCshMNSEGS     =      500    IPCshMSPAGES =    262144
SUPERUSER       = BPXROOT    FORKCOPY       = COW
STEPLIBLIST     =
USERIDALIASTABLE=
SERV_LINKLIB    = POSIX.DYNSERV.LOADLIB  BPXLK1
SERV_LPALIB     = POSIX.DYNSERV.LOADLIB  BPXLK1
PRIORITYPG VALUES: NONE
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGS   =      1000    SHRLIBRGNSIZE =    67108864
SHRLIBMAXPAGES  =      4096    VERSION       = /
SYSCALL COUNTS  = NO          TTYGROUP        = TTY
SYSPLEX         = NO          BRML SERVER      = N/A
LIMMSG          = NONE        AUTOCVT         = OFF
RESOLVER PROC   = DEFAULT
AUTHPGMLIST     = NONE
SWA             = BELOW
```

- Mit dem Bedienerbefehl **DISPLAY OMVS,LIMITS** werden Informationen zu aktuellen z/OS UNIX System Services-PARMLIB-Begrenzungen, ihren oberen Grenzen und zur aktuellen Systembelegung angezeigt.

```
d omvs,l
BPX0051I 14.05.52 DISPLAY OMVS 904
OMVS      0042 ACTIVE      OMVS=(69)
SYSTEM WIDE LIMITS:      LIMMSG=SYSTEM

      CURRENT  HIGHWATER  SYSTEM
      USAGE    USAGE     LIMIT
MAXPROCSYS      1         4      256
MAXUIDS         0         0      200
MAXPTYS         0         0      256
MAXMMAPAREA     0         0      256
MAXSHAREPAGES   0         10     4096
IPCMSGNIDS      0         0      500
IPCSEMNIDS      0         0      500
IPCSHMNIDS      0         0      500
IPCSHMPAGES     0         0     262144 *
IPCMSGQBYTES    ---        0     262144
IPCMSGQMNUM     ---        0     10000
IPCSHMPAGES     ---        0      256
SHRLIBRGNSIZE   0         0     67108864
SHRLIBMAXPAGES  0         0      4096
```

Wenn zusätzlich das Schlüsselwort **PID=processid** angegeben wird, zeigt der Befehl die oberen Grenzen und die aktuelle Verwendung für einen einzelnen Prozess an.

```
d,omvs,l,pid=16777456
BPX0051I 14.06.28 DISPLAY OMVS 645
OMVS      000E ACTIVE      OMVS=(76)
USER      JOBNAME  ASID      PID      PPID STATE  START  CT_SECS
STCRSE    RSED8    007E    16777456  67109106 HF---- 20.00.56 113.914
LATCHWAITPID= 0 CMD=java -Ddaemon.log=/var/rdz/logs -
PROCESS LIMITS:      LIMMSG=NONE

      CURRENT  HIGHWATER  PROCESS
      USAGE    USAGE     LIMIT
MAXFILEPROC      83      103      256
MAXFILESIZE      ---      ---     NOLIMIT
MAXPROCUSER     97      99      200
MAXQUEUEDSIGS    0         1     1000
MAXTHREADS       9       14      200
MAXTHREADTASKS   9       14     1000
IPCSHMNSEGS     0         0      500
MAXCORESIZE     ---      ---     4194304
MAXMEMLIMIT     0         0     16383P
```

- Mit dem Bedienerbefehl **DISPLAY OMVS,PFS** werden Informationen zu jedem physischen Dateisystem angezeigt, das derzeit Teil der z/OS UNIX-Konfiguration ist. Dies schließt die TCP/IP-Stacks ein.

```
d omvs,p
BPX0046I 14.35.38 DISPLAY OMVS 092
OMVS      000E ACTIVE      OMVS=(33)
PFS CONFIGURATION INFORMATION
PFS TYPE  DESCRIPTION      ENTRY      MAXSOCK  OPNSOCK  HIGHUSED
TCP      SOCKETS AF_INET  EZBPFINI  50000    244      8146
UDS        SOCKETS AF_UNIX  BPXTUINT    64        6        10
ZFS        LOCAL FILE SYSTEM
          14:32.00 RECYCLING
HFS        LOCAL FILE SYSTEM  GFUAINIT
BPXFTCLN   CLEANUP DAEMON    BPXFTCLN
BPXFTSYN   SYNC DAEMON      BPXFTSYN
BPXFPINT   PIPE              BPXFPINT
BPXFCSIN   CHAR SPECIAL      BPXFCSIN
NFS        REMOTE FILE SYSTEM GFSCINIT
PFS NAME   DESCRIPTION      ENTRY      STATUS   FLAGS
TCP41      SOCKETS          EZBPFINI   ACT      CD
TCP42      SOCKETS          EZBPFINI   ACT
TCP43      SOCKETS          EZBPFINI   INACT    SD
TCP44      SOCKETS          EZBPFINI   INACT
PFS PARM INFORMATION
HFS        SYNCDEFAULT(60) FIXED(50) VIRTUAL(100)
          CURRENT VALUES: FIXED(55) VIRTUAL(100)
NFS        biod(6)
```

- Mit dem Bedienerbefehl **DISPLAY OMVS,PID=processid** werden die Threadinformationen zu einem bestimmten Prozess angezeigt.

```
d omvs,pid=16777456
BPX0040I 15.30.01 DISPLAY OMVS 637
OMVS      000E ACTIVE      OMVS=(76)
USER      JOBNAME ASID      PID      PPID STATE  START  CT_SECS
STCRSE    RSED8   007E    16777456  67109106 HF---- 20.00.56 113.914
LATCHWAITPID= 0 CMD=java -Ddaemon.log=/var/rdz/logs -
THREAD_ID  TCB@    PRI_JOB  USERNAME  ACC_TIME SC STATE
0E08A00000000000 005E6DF0 OMVS      .927 RCV FU
0E08F00000000001 005E6C58      .001 PTX JYNV
0E09300000000002 005E6AC0      7.368 PTX JYNV
0E0CB00000000008 005C2CF0 OMVS      1.872 SEL JFNV
0E192000000003CE 005A0B70 OMVS      IBMUSER   14.088 POL JFNV
0E18D000000003CF 005A1938      IBMUSER    .581 SND JYNV
```

Netz überwachen

Für die Unterstützung einer großen Anzahl von Clients, die eine Verbindung mit dem Host herstellen, muss nicht nur Developer for System z, sondern auch Ihre Netzinfrastruktur in der Lage sein, die Arbeitslast zu verarbeiten. Die Netzverwaltung ist ein umfangreiches und gut dokumentiertes Thema, das nicht in der Dokumentation von Developer for System z behandelt wird. Aus diesem Grund werden nur die folgenden Verweise zur Verfügung gestellt.

- Mit dem Bedienerbefehl **DISPLAY NET,CSM** können Sie die Verwendung von Speicher überwachen, der mit Communication Storage Manager (CSM) verwaltet wird. Wie in *Communications Server SNA Operations* (IBM Form SC31-8779) beschrieben, können Sie mit diesem Befehl ermitteln, wie viel Speicher für ECSA- und Datenraum-Speicherpools verwendet wird.

z/OS UNIX-Dateisysteme überwachen

In Developer for System z werden z/OS UNIX-Dateisysteme verwendet, um verschiedene Datentypen (beispielsweise Protokolle und temporäre Dateien) zu speichern. Verwenden Sie den z/OS UNIX-Befehl **df**, um anzuzeigen, wie viele Dateideskriptoren noch verfügbar sind und wie viel freier Speicherbereich vor der Erstellung der nächsten Erweiterung der zugrunde liegenden HFS- oder zFS-Datei verfügbar ist.

```
$ df
Mounted on   Filesystem      Avail/Total   Files      Status
/tmp          (OMVS.TMP)      1393432/1396800 4294967248 Available
/u/ibmuser   (OMVS.U.IBMUSER) 1248/1728     4294967281 Available
/usr/lpp/rdz  (OMVS.FEK.HHOP760) 3062/43200    4294967147 Available
/var         (OMVS.VAR)      27264/31680    4294967054 Available
```

Beispielkonfiguration

Die folgende Beispielkonfiguration zeigt die erforderliche Konfiguration zur Unterstützung der folgenden Anforderungen:

- 500 simultane Clientverbindungen
- 300 simultane MVS-Builds (Batch-Job)
- 200 simultane CARMA-Verbindungen (mit der Startmethode CRASTART)
- Zeitlimitüberschreitung nach 3 Stunden Inaktivität
- Verwendung von z/OS UNIX nicht zulassen
- SCLM Developer Toolkit und die File Manager-Integration werden nicht verwendet
- Schätzung der durchschnittlichen Verwendung des Java-Heapspeichers mit 5 MB
- Benutzer verwenden eindeutige z/OS UNIX-Benutzer-IDs

Thread-Pool-Anzahl

Developer for System z versucht standardmäßig, 60 Benutzer zu einem einzigen Thread-Pool hinzuzufügen. In den Anforderungen ist allerdings angegeben, dass die Zeitlimitüberschreitung aufgrund von Inaktivität aktiv ist. In Tabelle 44 auf Seite 223 sehen Sie, dass daher pro verbundenem Client ein Thread hinzugefügt wird. Dieser Thread ist ein Zeitgeberthread und somit ständig aktiv. So wird verhindert, dass RSE 60 Benutzer einem einzigen Thread-Pool hinzufügt. ($60 \cdot (16+1) = 1020$ und `maximum.threads` ist standardmäßig auf 1000 gesetzt.)

Es wäre möglich, `maximum.threads` zu erhöhen. Weil laut Anforderung allerdings pro Benutzer ein durchschnittlicher Java-Heapspeicher von 5 MB bereitgestellt werden soll, wird `maximum.clients` auf 50 verringert. Auf diese Weise ($5 \cdot 50 = 250$) wird die standardmäßige maximale Größe des Java-Heapspeichers (256 MB) beachtet.

Mit 50 Clients pro Thread-Pool und der Anforderung zur Unterstützung von 500 Verbindungen werden somit 10 Thread-Pool-Adressräume benötigt.

Mindestbegrenzungen festlegen

Mithilfe der in diesem Kapitel bereits gezeigten Formeln und der Kriterien, die am Anfang dieses Abschnitts genannt wurden, kann die Ressourcennutzung festgelegt werden, die verarbeitet werden muss.

- Anzahl der Adressräume (Maximum)
 $3 + A + N \cdot (x + y + z) + (2 + N \cdot 0.01)$
 $3 + 10 + 500 \cdot 1 + 200 \cdot 1 + 300 \cdot 1 + (2 + 500 \cdot 0.01) = 1020$
- Anzahl der Adressräume (pro Benutzer)
 $x + y + z$
 $1 + 1 + 1 = 3$
- Anzahl der Prozesse (Maximum)
 $7 + 2 \cdot A + N \cdot (x + y + z) + (10 + N \cdot 0.05)$
 $7 + 2 \cdot 10 + 500 \cdot 2 + 200 \cdot 1 + 300 \cdot 0 + (10 + 500 \cdot 0.05) = 1562$
- Anzahl der Prozesse (pro Benutzer)
 $(x + y + z) + 5 \cdot s$
 $(2 + 1 + 0) + 5 \cdot 0 = 3$
- Anzahl der Threads - RSE-Thread-Pool
 $9 + N \cdot (16 + x + y + z) + (20 + N \cdot 0.1)$
 $9 + 60 \cdot (16 + 1 + 4 + 0) + (20 + 60 \cdot 0.1) = 1295$
- Anzahl der Threads - JES Job Monitor
 $3 + N$
 $3 + 500 = 503$
- Benutzer-IDs
 $500 + 3 = 503$

Die 3 zusätzlichen Benutzer-IDs werden für STCJMON, STCLOCK und STCRSE (die Benutzer-IDs der gestarteten Tasks für Developer for System z) benötigt.

Begrenzungen definieren

Da jetzt die Zahlen für die Ressourcennutzung bekannt sind, können die begrenzenden Anweisungen mit entsprechenden Werten angepasst werden.

- /etc/rdz/rsed.envvars
 - Xmx256m
nicht geändert
 - Dmaximum.clients=50
 - Dmaximum.threads=1000
nicht geändert
 - Dminimum.threadpool.process=10
Diese Änderung ist optional. RSE startet neue Thread-Pools, wenn erforderlich.
 - DHIDE_ZOS_UNIX=true
 - DDSTORE_IDLE_SHUTDOWN_TIMEOUT=10800000
- FEK.#CUST.PARMLIB(FEJJC�FG)
 - MAX_THREADS=503
- SYS1.PARMLIB(BPXPRMxx)
 - MAXPROCSYS(2500)
mindestens 1562, zusätzliche Puffer für Tasks hinzugefügt, die nicht zu Developer for System z gehören
 - MAXPROCUSER(25)
nicht geändert, mindestens 3
 - MAXTHREADS(1500)

mindestens 503 (für JES Job Monitor), wenn THREADSMAX im OMVS-Segment der Benutzer-ID 'STCRSE' verwendet wird, um die Begrenzung für RSE festzulegen (mindestens 1295)
 - MAXTHREADTASKS(1500)

mindestens 503 (für JES Job Monitor), wenn THREADSMAX im OMVS-Segment der Benutzer-ID 'STCRSE' verwendet wird, um die Begrenzung für RSE festzulegen (mindestens 1295)
 - MAXUIDS(700)

mindestens 503, zusätzliche Puffer für Tasks hinzugefügt, die nicht zu Developer for System z gehören
 - MAXASSIZE(209715200)

nicht geändert (Systemstandardwert: 200 MB), ASSIZEMAX im OMVS-Segment der Benutzer-ID 'STCRSE' wird verwendet
- SYS1.PARMLIB(IEASYSxx)
 - MAXUSER=2000

mindestens 1020, zusätzliche Puffer für Tasks hinzugefügt, die nicht zu Developer for System z gehören
- OMVS-Segment der Benutzer-ID 'STCRSE'
 - ASSIZEMAX(2147483647)

2 GB

Ressourcennutzung überwachen

Nachdem Sie die Systemgrenzwerte, wie unter „Begrenzungen definieren“ auf Seite 246 dokumentiert, aktiviert haben, kann die Überwachung der Ressourcennutzung durch Developer for System z starten, um zu ermitteln, ob die Anpassung von Variablen erforderlich ist. Abb. 58 zeigt die Ressourcennutzung, nachdem sich 495 Benutzer angemeldet haben. (Das Beispiel in der Abbildung zeigt nur die Anmeldung. Es sind keine Benutzeraktionen angegeben.)

```
BPXM023I (STCRSE)
ProcessId(16779764) Memory Usage(10%) Clients(50) Order(1)
ProcessId(67108892) Memory Usage(16%) Clients(50) Order(2)
ProcessId(67108908) Memory Usage(10%) Clients(50) Order(3)
ProcessId(67108898) Memory Usage(16%) Clients(50) Order(4)
ProcessId(67108916) Memory Usage(16%) Clients(50) Order(5)
ProcessId(67108897) Memory Usage(16%) Clients(50) Order(6)
ProcessId(67108921) Memory Usage(16%) Clients(50) Order(7)
ProcessId(83886146) Memory Usage(16%) Clients(50) Order(8)
ProcessId(67108920) Memory Usage(16%) Clients(50) Order(9)
ProcessId(3622      ) Memory Usage(8%) Clients(45) Order(10)
```

Jobname	Cpu time	Storage	EXCP
JMON	1.74	43.0M	2753
LOCKD	10.05	31.9M	24621
RSED	6.65	40.1M	41780
RSED1	8.17	187.0M	76566
RSED2	13.04	184.9M	78946
RSED3	17.77	181.1M	76347
RSED4	11.63	174.9M	74638
RSED5	15.27	172.9M	72883
RSED6	13.85	180.8M	75031
RSED7	9.79	174.3M	76636
RSED8	21.59	176.1M	70583
RSED8	18.88	184.7M	76953
RSED9	9.52	189.8M	80490

Abbildung 58. Ressourcennutzung der Beispielfunktion

Kapitel 14. Leistungsaspekte

z/OS ist ein sehr anpassungsfähiges Betriebssystem, bei dem (manchmal kleine) Systemänderungen eine enorme Auswirkung auf die Gesamtleistung haben können. Dieses Kapitel hebt einige der Änderungen hervor, die zu einer Verbesserung der Leistung von Developer for System z führen können.

Weitere Informationen zur Systemoptimierung finden Sie im *MVS Initialization and Tuning Guide* (IBM Form SA22-7591) sowie in der Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Dateisystem zFS verwenden

Das zFS (zSeries File System) und das HFS (Hierarchical File System) sind UNIX-Dateisysteme, die in einer z/OS UNIX-Umgebung verwendet werden können. Das zFS bietet jedoch die folgenden Features und Vorteile:

- Leistungssteigerung in der Umgebung vieler Kunden beim Zugriff auf Dateien mit einer Größe von annähernd 8 K, wenn die Dateien häufig aufgerufen und aktualisiert werden. Die Zugriffszeit bei kleineren Dateien entspricht der des HFS.
- Erstellen eines schreibgeschützten Klons eines Dateisystems in derselben Datei. Das geklonte Dateisystem kann Benutzern als schreibgeschützte Zeitpunktkopie eines Dateisystems bereitgestellt werden. Dieses optionale Feature ist nur in einer Nicht-Sysplex-Umgebung verfügbar.
- Das zFS ist das strategische z/OS UNIX-Dateisystem. Die Funktionalität des HFS wurde stabilisiert. Funktionale Erweiterungen werden jedoch nur für das zFS bereitgestellt.

Wenn Sie mehr über das zFS erfahren möchten, lesen Sie die Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Verwendung von STEPLIB vermeiden

Jeder z/OS UNIX-Prozess mit einer STEPLIB, die vom übergeordneten Element zum untergeordneten Element oder über eine Exec weitergegeben wird, belegt einen erweiterten allgemeinen Speicherbereich (ECSA, Extended Common Storage Area) von ca. 200 Bytes. Wenn die Umgebungsvariable STEPLIB nicht oder mit STEPLIB=CURRENT definiert ist, gibt z/OS UNIX alle aktiven TASKLIB-, STEPLIB- und JOBLIB-Zuordnungen während einer Funktion fork(), spawn() oder exec() weiter.

In `rsed.envvars` ist die Standardeinstellung für Developer for System z mit STEPLIB=NONE codiert. Lesen Sie hierzu die Beschreibung in der Konfigurationsdatei `rsed.envvars`. Aus den oben genannten Gründen sollten Sie diese Anweisung nicht ändern und die resultierenden Dateien stattdessen in die LINKLIST oder den LPA (Link-Pack-Bereich) stellen.

Zugriff auf Systembibliotheken verbessern

Bestimmte Systembibliotheken und Lademodule werden von z/OS UNIX und Aktivitäten der Anwendungsentwicklung besonders häufig verwendet. Wenn Sie den Zugriff auf diese Bibliotheken und Module verbessern, indem Sie sie beispielsweise zum Link-Pack-Bereich (LPA) hinzufügen, können Sie die Systemleistung steigern.

Weitere Informationen zu den nachfolgend beschriebenen SYS1.PARMLIB-Membren enthält die Veröffentlichung *MVS Initialization and Tuning Reference* (IBM Form SA22-7592).

LE-Laufzeitbibliotheken (Language Environment)

Wenn C-Programme (einschließlich der z/OS UNIX-Shell) ausgeführt werden, verwenden sie häufig Routinen aus der LE-Laufzeitbibliothek (Language Environment). Für jeden Adressraum, der ein LE-fähiges Programm ausführt, werden ungefähr 4 MB der Laufzeitbibliothek in den Speicher geladen und in jede Verzweigung kopiert.

CEE.SCEELPA

Die Datei CEE.SCEELPA enthält eine Untergruppe der LE-Laufzeitroutinen, die besonders oft von z/OS UNIX verwendet werden. Sie sollten diese Datei zu SYS1.PARMLIB(LPALSTxx) hinzufügen, um einen maximalen Leistungsgewinn zu erzielen. Wenn Sie dieser Empfehlung folgen, werden die Module nur einmal von der Platte gelesen und an einer gemeinsam genutzten Position gespeichert.

Anmerkung: Fügen Sie die folgende Anweisung zu SYS1.PARMLIB(PROGxx) hinzu, wenn Sie die Lademodule lieber zum dynamischen LPA (Link-Pack-Bereich) hinzufügen möchten:

```
LPA ADD MASK(*) DSN(CEE.SCEELPA)
```

Außerdem sollten Sie die LE-Laufzeitbibliotheken CEE.SCEERUN und CEE.SCEERUN2 in die LINKLIST stellen, indem Sie die Dateien zu SYS1.PARMLIB(LNKLSTxx) oder SYS1.PARMLIB(PROGxx) hinzufügen. Auf diese Weise entfällt der z/OS UNIX-Systemaufwand für die STEPLIB und das Ein-/Ausgabevolumen verringert sich infolge der Verwaltung durch LLA und VLF oder ähnliche Produkte.

Anmerkung: Fügen Sie aus denselben Gründen ebenfalls die C/C++-DLL-Klassenbibliothek CBC.SCLBDLL zur LINKLIST hinzu.

Wenn Sie sich entschließen, diese Bibliotheken nicht in die LINKLIST zu stellen, müssen Sie in der Datei rsed.envvars die entsprechende STEPLIB-Anweisung konfigurieren. Lesen Sie hierzu die Beschreibung in der Konfigurationsdatei rsed.envvars. Obwohl diese Methode immer zusätzlichen virtuellen Speicher verwendet, können Sie die Leistung verbessern, indem Sie die LE-Laufzeitbibliotheken für LLA oder ein ähnliches Produkt definieren. Dadurch werden die Ein-/Ausgaben reduziert, die für das Laden der Module erforderlich sind.

Anwendungsentwicklung

Auf Systemen, deren primäre Aktivität die Anwendungsentwicklung ist, kann auch eine Leistungsverbesserung erreicht werden, wenn der Linkage-Editor in den dynamischen LPA gestellt wird. Hierfür müssen die folgenden Zeilen zu SYS1.PARMLIB(PROGxx) hinzugefügt werden:

```
LPA ADD MODNAME(CEEINIT,CEEELIB,CEEV003,EDCV) DSN(CEE.SCEERUN)
LPA ADD MODNAME(IEFIB600,IEFXB603) DSN(SYS1.LINKLIB)
```

Für die C/C++-Entwicklung können Sie außerdem die Compilerdatei CBC.SCCNCMP zu SYS1.PARMLIB(LPALSTxx) hinzufügen.

Die obigen Anweisungen sind Beispiele für mögliche LPA-Kandidaten. Die Anforderungen an Ihrem Standort können jedoch andere Maßnahmen erfordern. Informationen zur Aufnahme anderer LE-Lademodule in den dynamischen LPA enthält

die Veröffentlichung *Language Environment Customization* (IBM Form SA22-7564). Wie Lademodule von C/C++-Compilern in den dynamischen LPA gestellt werden, erfahren Sie in der Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Durchsatz der Sicherheitsprüfung verbessern

Wenn Sie den Durchsatz der für z/OS UNIX durchgeführten Sicherheitsprüfung verbessern möchten, definieren Sie in der Klasse FACILITY Ihrer Sicherheitssoftware das Profil BPX.SAFFASTPATH. Dadurch wird für ein breites Spektrum von Operationen der Systemaufwand für die z/OS UNIX-Sicherheitsprüfungen verringert, z. B. für die Überprüfung des Dateizugriffs und des IPC-Zugriffs sowie für die Überprüfung der Eigentumsrechte an Prozessen. Weitere Informationen zu diesem Profil enthält die Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Anmerkung: Benutzer benötigen keine Berechtigung für das Profil BPX.SAFFASTPATH.

Auslastungsverwaltung

An jedem Standort gelten ganz bestimmte Anforderungen. Das Betriebssystem z/OS kann so angepasst werden, dass die verfügbaren Ressourcen optimal genutzt werden, um diese Anforderungen zu erfüllen. Bei der Auslastungsverwaltung definieren Sie Leistungsziele und ordnen jedem dieser Ziele eine geschäftliche Bedeutung zu. Sie definieren Arbeitsziele mit Geschäftsbegriffen, und das System entscheidet, wie viele Ressourcen (z. B. CPU und Speicher) der Arbeit zugeordnet werden müssen, um das angestrebte Ziel zu erreichen.

Indem Sie für die Prozesse von Developer for System z die richtigen Ziele festlegen, können Sie für eine ausgeglichene Leistung des Produkts sorgen. Nachfolgend sind dazu einige allgemeine Richtlinien aufgelistet.

- Ordnen Sie die APPC-Transaktion (falls Sie verwendet wird) einer TSO-Leistungsgruppe zu.
- Fügen Sie eine Leistungsgruppe für gestartete Tasks (SYSSTC) zu den Serveradressräumen von Developer for System z hinzu: JES Job Monitor (JMON), Sperrdämon (LOCKD), RSE-Dämon (RSED) und RSE-Thread-Pools (RSEDx).

Weitere Informationen zu diesem Thema finden Sie in der Veröffentlichung *MVS Planning Workload Management* (IBM Form SA22-7602).

Feste Java-Heapgröße

Bei einem Heapspeicher fester Größe gibt es keine Erweiterung oder Verkleinerung, was in bestimmten Situationen zu einer deutlichen Leistungssteigerung führen kann. Generell ist die Verwendung eines Heapspeichers mit fester Größe jedoch keine gute Idee, weil sie den Start der Garbage-Collection hinauszögert, bis der Heapspeicher voll ist. Die dann ausgeführte Garbage-Collection ist dementsprechend umfangreich. Außerdem steigt das Fragmentierungsrisiko, sodass eine Heapkomprimierung erforderlich ist. Heapspeicher mit fester Größe sollten Sie daher nur nach gründlichen Tests bzw. unter Anleitung des IBM Support Center verwenden. Weitere Informationen zu Heapgrößen und Garbage-Collections enthält der *Java Diagnostics Guide* (IBM Form SC34-6650).

Der Heapspeicher einer z/OS Java Virtual Machine (JVM) hat standardmäßig eine Anfangsgröße von 1 Megabyte. Die maximale Größe liegt bei 64 Megabytes. Die Grenzwerte können Sie mit den Java-Befehlszeilenoptionen `-Xms` (Anfangsgröße) und `-Xmx` (maximale Größe) setzen.

In Developer for System z sind Java-Befehlszeilenoptionen in der Steueranweisung `_RSE_JAVAOPTS` der Datei `rsed.envvars` definiert. Eine diesbezügliche Beschreibung finden Sie im Abschnitt „Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren“ auf Seite 42.

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx128m"
```

Java-Option -Xquickstart

Anmerkung: Die Java-Option `-Xquickstart` ist nur sinnvoll, wenn Sie für den RSE-Server die alternative Startmethode mit REXEC/SSH verwenden. Diese Methode ist im Abschnitt „REXEC (oder SSH) verwenden (optional)“ auf Seite 102 beschrieben.

Mit der Option `-Xquickstart` kann die Startzeit einiger Java-Anwendungen verbessert werden. `-Xquickstart` bewirkt die teiloptimierte Ausführung des JIT-Compilers und ermöglicht so eine schnelle Kompilierung. Durch diese schnelle Kompilierung wird wiederum die Startzeit verkürzt.

`-Xquickstart` ist für Anwendungen geeignet, die eine kurze Laufzeit haben, und insbesondere für jene, bei denen die Ausführungszeit nicht auf eine geringe Anzahl von Methoden konzentriert ist. Die Option `-Xquickstart` kann den Durchsatz verringern, wenn sie für Anwendungen genutzt wird, die eine lange Laufzeit haben und häufig verwendete Methoden enthalten.

Fügen Sie am Ende der Datei `rsed.envvars` die folgende Anweisung hinzu, um die Option `-Xquickstart` für den RSE-Server zu aktivieren:

```
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xquickstart"
```

Gemeinsame Klassennutzung durch mehrere JVMs

Die IBM Java Virtual Machine (JVM) bietet ab Version 5 die Möglichkeit, dass JVMs die Bootstrap-Klassen und Anwendungsklassen gemeinsam nutzen können, indem sie sie in einem Cache innerhalb des gemeinsam genutzten Speichers ablegt. Bei der gemeinsamen Nutzung von Klassen verwenden mehrere JVMs einen Cache gemeinsam, sodass insgesamt weniger virtueller Speicher belegt wird. Die gemeinsame Klassennutzung verkürzt außerdem die Startzeit für eine JVM, nachdem der Cache erstellt wurde.

Der Cache für gemeinsam genutzte Klassen ist von den aktiven JVMs unabhängig und bleibt über die Lebensdauer der JVM hinweg bestehen, die den Cache erstellt hat. Da der Cache für gemeinsam genutzte Klassen länger bestehen bleibt als jede JVM, wird er durch dynamische Aktualisierungen an alle Änderungen angepasst, die ggf. an JARs oder Klassen im Dateisystem vorgenommen wurden.

Der Systemaufwand für das Erstellen eines neuen Cache und das Füllen des Cache mit Daten ist minimal. Das Starten einer einzelnen JVM dauert im Vergleich zur gemeinsamen Nutzung von Klassen in der Regel 0 bis 5 % länger. Der genaue Unterschied im Zeitaufwand hängt davon ab, wie viele Klassen geladen werden. Bei einem mit Daten gefüllten Cache verkürzt sich die Startzeit für eine JVM im Vergleich zu einem System ohne gemeinsame Klassennutzung normalerweise um 10

bis 40 %. Die tatsächliche Beschleunigung ist vom Betriebssystem und von der Anzahl der geladenen Klassen abhängig. Bei mehreren gleichzeitig aktiven JVMs macht sich die Reduzierung der Gesamtstartzeit deutlicher bemerkbar.

Wenn Sie mehr über die gemeinsame Nutzung von Klassen erfahren möchten, lesen Sie den *Java SDK and Runtime Environment User Guide*.

Gemeinsame Klassennutzung aktivieren

Fügen Sie am Ende der Datei `rsed.envvars` die nachstehenden Anweisungen hinzu, um die gemeinsame Klassennutzung für den RSE-Server zu aktivieren. Die erste Anweisung definiert einen Cache mit dem Namen 'RSE' und mit Gruppenzugriff. Sie ermöglicht den Start des RSE-Servers, auch wenn die gemeinsame Klassennutzung scheitert. Die zweite Anweisung ist optional und setzt die Cachegröße auf 6 Megabytes. (Der Systemstandardwert liegt bei 16 MB.) Die dritte Anweisung fügt die Parameter für die gemeinsame Klassennutzung zu den Java-Startoptionen hinzu.

```
_RSE_CLASS_OPTS=-Xshareclasses:name=RSE,groupAccess,nonFatal
#_RSE_CLASS_OPTS=$_RSE_CLASS_OPTS -Xscmx6m
_RSE_JAVAOPTS=$_RSE_JAVAOPTS $_RSE_CLASS_OPTS"
```

Anmerkung: Wie im Abschnitt „Cachesicherheit“ erwähnt, müssen alle Benutzer, die die gemeinsam genutzte Klasse verwenden, dieselbe primäre Gruppen-ID (GID) haben. Das bedeutet, dass in der Sicherheitssoftware dieselbe Standardgruppe für die Benutzer definiert sein muss bzw. dass verschiedene Standardgruppen in den OMVS-Segmenten der Benutzer dieselbe GID haben.

Cachegrößenbegrenzung

Die theoretische maximale Größe des gemeinsam genutzten Cache liegt bei 2 GB. Die Cachegröße, die Sie angeben können, wird durch den auf dem System verfügbaren physischen Hauptspeicher und den verfügbaren Auslagerungsspeicher begrenzt. Da der virtuelle Adressraum eines Prozesses sowohl vom Cache für gemeinsam genutzte Klassen als auch vom Java-Heapspeicher verwendet wird, führt eine Erhöhung der maximalen Java-Heapgröße dazu, dass Sie einen entsprechend kleineren Cache für gemeinsam genutzte Klassen erstellen können.

Cachesicherheit

Der Zugriff auf den Cache für gemeinsam genutzte Klassen wird durch Berechtigungen des Betriebssystems und Java-Sicherheitsberechtigungen beschränkt.

Standardmäßig wird für die Erstellung von Klassencaches die Sicherheit auf Benutzerebene verwendet, sodass nur der Benutzer, der den Cache erstellt hat, auf den Cache zugreifen kann. Unter z/OS UNIX gibt es die Option `groupAccess`, die allen Benutzern Zugriff gewährt, die zur Primärgruppe des Benutzers gehören, der den Cache erstellt hat. Zerstört werden kann ein Cache unabhängig von der verwendeten Zugriffsebene nur von dem Benutzer, der ihn erstellt hat, oder von einem Benutzer 'root' (UID 0).

Wenn Sie mehr über zusätzliche Sicherheitsoptionen bei Verwendung eines Java-Sicherheitsmanagers erfahren möchten, lesen Sie den *Java SDK and Runtime Environment User Guide*.

SYS1.PARMLIB(BPXPRMxx)

Einige der Einstellungen von SYS1.PARMLIB(BPXPRMxx) wirken sich bei gemeinsam genutzten Klassen auf den Durchsatz aus. Falsche Einstellungen können dazu führen, dass die gemeinsam genutzten Klassen nicht funktionieren. Diese Einstellungen können sich auch auf die Leistung auswirken. Weitere Informationen zur Verwendung dieser Parameter und zu ihrer Auswirkung auf die Leistung enthalten die Veröffentlichungen *MVS Initialization and Tuning Reference* (IBM Form SA22-7592) und *UNIX System Services Planning* (IBM Form GA22-7800). Die hinsichtlich der Verarbeitung gemeinsam genutzter Klassen wichtigsten BPXPRMxx-Parameter sind folgende:

- MAXSHAREPAGES, IPCSHMPAGES, IPCSHMMPAGES und IPCSHMNSEGS

Diese Einstellungen beeinflussen, wie viele gemeinsam genutzte Speicherseiten der JVM zur Verfügung stehen. Für einen z/OS UNIX-System Service (31 Bit) hat die gemeinsam genutzte Seite eine feste Größe von 4 KB. Gemeinsam genutzte Klassen versuchen standardmäßig, einen Cache mit einer Größe von 16 MB zu erstellen. Sie sollten IPCSHMMPAGES deshalb auf einen Wert größer als 4096 setzen.

Wenn Sie die Cachegröße mit -Xscmx festlegen, rundet die JVM den Wert auf das nächste volle Megabyte auf. Berücksichtigen Sie dies, wenn Sie IPCSHMMPAGES auf Ihrem System setzen.

- IPCSEMNIDS und IPCSEMNSEMS

Diese Einstellungen beeinflussen, wie viele Semaphore für UNIX-Prozesse zur Verfügung stehen. Gemeinsam genutzte Klassen verwenden für die Kommunikation zwischen JVMs IPC-Semaphore.

Plattenspeicherplatz

Der Cache für gemeinsam genutzte Klassen benötigt zum Speichern von Kennungsdaten der auf dem System vorhandenen Caches Plattenspeicherplatz. Diese Daten werden unter /tmp/javasharedresources gespeichert. Wenn das Verzeichnis mit den Kennungsdaten gelöscht wird, kann die JVM nicht die gemeinsam genutzten Klassen auf dem System identifizieren und muss den Cache neu erstellen.

Dienstprogramme für Cacheverwaltung

Der Java-Zeilenbefehl -Xshareclasses kann mit verschiedenen Optionen verwendet werden, zu denen auch Dienstprogramme für die Cacheverwaltung gehören. Drei dieser Dienstprogramme sind im folgenden Beispiel enthalten (\$ ist die z/OS UNIX-Eingabeaufforderung). Eine vollständige Übersicht über die unterstützten Befehlszeilenoptionen enthält der *Java SDK and Runtime Environment User Guide*.

```
$ java -Xshareclasses:listAllCaches
Shared Cache      OS shmid      in use      Last detach time
RSE               401412       0           Mon Jun 18 17:23:16 2007
```

Could not create the Java virtual machine.

```
$ java -Xshareclasses:name=RSE,printStats
```

Current statistics for cache "RSE":

```
base address      = 0x0F300058
end address       = 0x0F8FFFF8
allocation pointer = 0x0F4D2E28
```



```
cache size      = 6291368
free bytes      = 4355696
ROMClass bytes  = 1912272
Metadata bytes  = 23400
Metadata % used = 1%
```

```
# ROMClasses      = 475
# Classpaths      = 4
# URLs            = 0
# Tokens          = 0
# Stale classes   = 0
% Stale classes   = 0%
```

Cache is 30% full

Could not create the Java virtual machine.

```
$ java -Xshareclasses:name=RSE,destroy
JVMSHRC010I Shared Cache "RSE" is destroyed
Could not create the Java virtual machine.
```

Anmerkung:

- Cachedienstprogramme führen die erforderliche Operation für den angegebenen Cache aus, ohne die JVM zu starten. Die Nachricht "Could not create the Java virtual machine." ist daher normal.
- Ein Cache kann nur zerstört werden, wenn alle JVMs, die den Cache benutzen, beendet sind und der Benutzer, der den Befehl ausführt, über ausreichende Berechtigungen verfügt.

Kapitel 15. CICSTS-Aspekte

Traditionell ist das Definieren von Ressourcen für CICS dem CICS-Administrator vorbehalten. Dass Anwendungsentwickler nur ungern mit dieser Aufgabe betraut werden, hat verschiedene Gründe:

- Die meisten CICS-Ressourcendefinitionen enthalten aufgrund ihrer Komplexität, ihrer Wechselwirkung mit anderen Ressourcendefinitionen und aufgrund von Geschäftsstandards viele Parameter. Für eine korrekte Definition sind daher die Kenntnisse eines CICS-Administrators erforderlich. Fehlerhafte Definitionen können zu unerwarteten Ergebnissen mit möglichen Auswirkungen auf die gesamte CICS-Region führen.
- Die meisten Kundenunternehmen stellen CICS-Entwicklungs- und -Testumgebungen bereit, die für mehrere Anwendungsgruppen und Entwickler zur gemeinsamen Nutzung verfügbar sein müssen. Viele Kundenunternehmen haben Service-Level-Agreements etabliert, die eine strikte Kontrolle dieser Umgebungen vorsehen.

Developer for System z unterstützt diesen Ansatz, denn CICS-Administratoren haben die Möglichkeit, die Standardwerte für CICS-Ressourcendefinitionen zu kontrollieren und die Anzeigemerkmale von CICS-Ressourcendefinitionsparametern mithilfe des CRD-Servers zu steuern, der Teil von Application Deployment Manager ist.

Der CICS-Administrator kann beispielsweise bestimmte Parameter für CICS-Ressourcendefinitionen bereitstellen, die nicht vom Anwendungsentwickler aktualisiert werden können. Andere Parameter für CICS-Ressourcendefinitionen können mit oder ohne Vorgabe von Standardwerten zur Aktualisierung freigegeben werden. Der CICS-Ressourcendefinitionsparameter kann auch ausgeblendet werden, um unnötige Komplexität zu vermeiden.

Sobald der Anwendungsentwickler mit den CICS-Ressourcendefinitionen zufrieden ist, können sie in der aktiven CICS-Testumgebung installiert werden. Sie können die Definitionen aber auch zur weiteren Bearbeitung und zur Genehmigung durch einen CICS-Administrator in ein Manifest exportieren. Der CICS-Administrator kann Änderungen an Ressourcendefinitionen mit dem Verwaltungsdienstprogramm (Batchdienstprogramm) oder dem Manifestverarbeitungstool implementieren.

Anmerkung: Das Manifestverarbeitungstool ist ein Plug-in zum IBM CICS-Explorer.

Weitere Informationen zu den erforderlichen Schritten für die Konfiguration von Application Deployment Manager auf Ihrem Hostsystem enthält Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73.

Durch die Anpassung von Application Deployment Manager werden die folgenden Services zu Developer for System z hinzugefügt:

- Auf dem Client: IBM CICS Explorer stellt eine Eclipse-basierte Infrastruktur für die Anzeige und Verwaltung von CICS-Ressourcen bereit und verbessert die Integration der verschiedenen CICS-Tools.
- Auf dem Client: Der Editor für CICS-Ressourcendefinitionen (CRD)

- Auf dem Host: Der CRD-Server (CICS Resource Definition) wird als CICS-Anwendung unter z/OS ausgeführt.

Zum CRD-Server von Application Deployment Manager gehören der Server selbst, ein CRD-Repository, die zugehörigen CICS-Ressourcendefinitionen, Bindungsdateien für Web-Services (sofern die Web-Service-Schnittstelle verwendet wird) und ein Beispielhandler für Pipelinenachrichten. Der CRD-Server muss in einer Webverwaltungsregion (WOR, Web Owning Region) ausgeführt werden, die in der Dokumentation zu Developer for System z als 'primäre CICS-Verbindungsregion' bezeichnet wird.

Weitere Informationen zu den Application Deployment Manager-Services, die im aktuellen Release von Developer for System z enthalten sind, finden Sie im Information Center für Developer for System z: <http://publib.ibm.com/infocenter/ratdevz/v7r6/index.jsp>.

RESTful oder Web-Service

CICS Transaction Server stellt ab Version 4.1 Unterstützung für eine HTTP-Schnittstelle zur Verfügung, die mithilfe von RESTful-Prinzipien (Representational State Transfer) entworfen wurde. Diese RESTful-Schnittstelle ist jetzt die strategische CICS-Schnittstelle, die von Clientanwendungen verwendet wird. Die ältere Web-Service-Schnittstelle wurde eingefroren. Erweiterungen werden nur für die RESTful-Schnittstelle entwickelt.

Application Deployment Manager hält diese Absichtserklärung ein. Für alle Services, die ab Developer for System z Version 7.6 neu sind, ist der RESTful-CRD-Server erforderlich.

Falls gewünscht, können die RESTful- und Web-Service-Schnittstellen gleichzeitig in einer CICS-Region aktiv sein. In diesem Fall sind in der Region zwei CRD-Server aktiv. Beide Server verwenden gemeinsam dasselbe CRD-Repository. Beachten Sie, dass CICS einige Warnungen zu doppelten Definitionen ausgibt, wenn die zweite Schnittstelle in der Region definiert wird.

Primäre und nicht primäre Verbindungsregionen

Eine CICS-Testumgebung kann aus mehreren MRO-Verbindungsregionen (Multi-Region Option) bestehen. Für diese Regionen haben sich im Laufe der Zeit inoffizielle Bezeichnungen eingeschlichen. So werden sie unter anderem als Terminalverwaltungsregion, Webverwaltungsregion, Anwendungsverwaltungsregion und Datenverwaltungsregion bezeichnet.

In einer Webverwaltungsregion wird die Unterstützung für CICS-Web-Services implementiert. In dieser Region muss der CRD-Server von Application Deployment Manager ausgeführt werden. Für Application Deployment Manager ist dies die primäre CICS-Verbindungsregion. Der CRD-Client implementiert eine Web-Service-Verbindung zur primären CICS-Verbindungsregion.

Nicht primäre CICS-Verbindungsregionen sind alle anderen Regionen, für die der CRD-Server Services bereitstellen kann. Zu diesen Services gehört die Anzeige von Ressourcen im IBM CICS-Explorer und das Definieren von Ressourcen mit dem Editor für CICS-Ressourcendefinitionen.

Wenn CICS-Ressourcendefinitionen der primären CICS-Verbindungsregion mit dem CICSplex SM Business Application Services (BAS) verwaltet wird, kann der CRD-Server auch für alle anderen von den BAS verwalteten CICS-Regionen Services bereitstellen.

Nicht von den BAS verwaltete CICS-Regionen erfordern zusätzliche Änderungen, um Services vom CRD-Server nutzen zu können.

CICS-Ressourceninstallation protokollieren

Aktionen, die der CRD-Server für die CICS-Ressourcen ausführt, werden in der CICS-CSDL-TD-Warteschlange protokolliert, die in der Regel auf DD MSGUSR in Ihrer CICS-Region zeigt.

Wenn Ihre CICS-Ressourcendefinitionen mit den CICSplex SM Business Application Services (BAS) verwaltet werden, muss die EYUPARM-Anweisung BASLOGMSG von CICSplex SM auf (YES) gesetzt sein, damit die Protokolle erstellt werden.

Application Deployment Manager, Sicherheit

Sicherheit des CRD-Repositorys

Die VSAM-Datei für das CRD-Server-Repository enthält alle Standardressourcendefinitionen und muss daher vor Aktualisierungen geschützt werden. Entwickler müssen jedoch die Möglichkeit haben, die hier gespeicherten Werte zu lesen. Beispiele für RACF-Befehle zum Schützen des CRD-Repositorys enthält der Abschnitt „Dateiprofile definieren“ auf Seite 181.

Pipelinesicherheit

Wenn CICS eine SOAP-Nachricht empfängt, wird sie von einer Pipeline verarbeitet. Eine Pipeline ist eine Gruppe von Nachrichtenhandlern, die nacheinander ausgeführt werden. CICS liest die Pipelinekonfiguration, um festzustellen, welche Nachrichtenhandler in der Pipeline aufgerufen werden sollen. Ein Nachrichtenhandler ist ein Programm, in dem Web-Service-Anforderungen und -Antworten auf spezielle Weise verarbeitet werden können.

Application Deployment Manager stellt ein Beispiel für eine Pipelinekonfigurationsdatei bereit, das Aufrufe für einen Nachrichtenhandler und ein Verarbeitungsprogramm für den SOAP-Header enthält.

Der Pipelinenachrichtenhandler (ADNTMSGH) wird für die Sicherheit verwendet. Er verarbeitet die Benutzer-ID und das Kennwort im SOAP-Header. ADNTMSGH wird von der Beispielpipelinekonfigurationsdatei referenziert und muss deshalb in die CICS-RPL-Kette gestellt werden.

Transaktionssicherheit

Eine von einer Pipeline aufgerufene Anwendung wird standardmäßig unter der Transaktions-ID CPIH ausgeführt. CPIH wird normalerweise gesetzt, wenn ein Mindestmaß an Berechtigungen erforderlich ist.

Developer for System z stellt mehrere Transaktionen bereit, die der CRD-Server beim Definieren und Abfragen von CICS-Ressourcen verwendet. Die Transaktions-IDs legt der CRD-Server je nach angeforderter Operation fest. Weitere Informationen zum Anpassen der Transaktions-IDs enthält Kapitel 4, „Application Deployment Manager (optional)“, auf Seite 73.

Transaktion	Beschreibung
ADMS	Für Änderungen an CICS-Ressourcen, die vom Manifestverarbeitungstool angefordert werden. Diese Transaktion ist normalerweise für CICS-Administratoren bestimmt. Diese Transaktion erfordert eine hohe Berechtigungsstufe.
ADMI	Für Anforderungen, die CICS-Ressourcen definieren, installieren oder deinstallieren. Diese Transaktion kann je nach Standortrichtlinien eine mittlere Berechtigungsstufe erfordern.
ADMR	Für alle anderen Anforderungen, die CICS-Umgebungsinformationen oder -Ressourceninformationen abrufen. Diese Transaktion kann je nach Standortrichtlinien ein Mindestmaß an Berechtigungen erfordern.

Einige oder alle der hier genannten Ressourcendefinitionsanforderungen der CRD-Servertransaktionen sollten geschützt werden. Sie sollten zumindest die Aktualisierungsbefehle schützen (Aktualisierung der Standard-Web-Service-Parameter, der Standarddeskriptorparameter und der Bindung zwischen Dateinamen), damit diese Befehle für das Definieren globaler Standardwerte für Ressourcen ausschließlich von CICS-Administratoren abgesetzt werden können.

Wenn die Transaktion zugeordnet wird, stellt die Sicherheitsprüfung für CICS-Ressourcen (sofern aktiviert) sicher, dass die Benutzer-ID berechtigt ist, die Transaktions-ID zu verwenden.

Die Ressourcenüberprüfung wird von der Option RESSEC der aktiven Transaktion, dem Systeminitialisierungsparameter RESSEC und - für den CRD-Server - vom Systeminitialisierungsparameter XPCT gesteuert.

Die Ressourcenüberprüfung findet nur statt, wenn der Systeminitialisierungsparameter XPCT einen anderen Wert als NO hat und die Option RESSEC der Transaktionsdefinition auf YES oder der Systeminitialisierungsparameter RESSEC auf ALWAYS gesetzt ist.

Die folgenden RACF-Befehle geben ein Beispiel für den Schutz von CRD-Servertransaktionen. Weitere Informationen zum Definieren der CICS-Sicherheit enthält der *RACF Security Guide for CICS*.

- RALTER GCICSTRN SYSADM UACC(NONE) ADDMEM(ADMS)
- PERMIT SYSADM CLASS(GCICSTRN) ID(#cicsadmin)
- RALTER GCICSTRN DEVELOPER UACC(NONE) ADDMEM(ADMI)
- PERMIT DEVELOPER CLASS(GCICSTRN) ID(#cicsdeveloper)
- RALTER GCICSTRN ALLUSER UACC(READ) ADDMEM(ADMR)
- SETROPTS RACLIST(TCICSTRN) REFRESH

Mit SSL verschlüsselte Kommunikation

Die SSL-Verschlüsselung des Datenstroms wird unterstützt, wenn der Application Deployment Manager-Client die Web-Service-Schnittstelle verwendet, um den CRD-Server aufzurufen. Die Verwendung von SSL für diese Kommunikation wird durch das Schlüsselwort SSL(YES) in der CICSTS-Definition TCPIP SERVICE gesteuert, wie in *RACF Security Guide for CICSTS* dokumentiert.

Ressourcensicherheit

CICSTS stellt die Funktionalität zur Verfügung, Ressourcen und die Befehle für die Bearbeitung zu schützen. Bestimmte Application Deployment Manager-Aktionen (beispielsweise Berechtigungen zum Bearbeiten neuer Ressourcentypen erteilen) schlagen möglicherweise fehl, wenn die Sicherheit aktiv ist, aber nicht vollständig konfiguriert ist.

Prüfen Sie bei einem Funktionsfehler in Application Deployment Manager das CICS-Protokoll auf Nachrichten wie die folgende. Ergreifen Sie Maßnahmen zur Fehlerbehebung, die im *RACF Security Guide for CICSTS* dokumentiert sind.

```
DFHXS1111 %date %time %applid %tranid Security violation by user
%userid at netname %portname for resource %resource in class
%classname. SAF codes are (X'safresp',X'safreas'). ESM codes are
(X'esmresp',X'esmreas').
```

Verwaltungsdienstprogramm

Mit dem von Developer for System z bereitgestellten Verwaltungsdienstprogramm können CICS-Administratoren die Standardwerte für CICS-Ressourcendefinitionen vorgeben. Diese Standardwerte können schreibgeschützt oder für den Anwendungsentwickler editierbar sein.

Das Verwaltungsdienstprogramm stellt die folgenden Funktionen bereit:

- CICSplex-Name für CICSplex-verwaltete Testumgebungen
- CICSplex-SM-Bereitstellungsgruppe
- Angabe der Einstellung für die Manifestexportregel
- Standardwerte und Anzeigeberechtigung für CICS-Ressourcenattribute
- Für VSAM-Dateidefinitionen verwendete Bindung einer logischen CICS-Datei an eine physische

Das Verwaltungsdienstprogramm wird vom Beispieljob ADNJSAPU in der Datei FEK.#CUST.JCL aufgerufen. Für die Verwendung dieses Dienstprogramms ist das Zugriffsrecht UPDATE für das CRD-Repository erforderlich.

ADNJSAPU ist in FEK.#CUST.JCL enthalten, sofern der z/OS-Systemprogrammierer während der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben hat. Weitere Details hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Anmerkung: Vor Ausführung des Jobs ADNJSPAU muss das CRD-Repository in CICS geschlossen werden. Nach dem Job können Sie das Repository wieder öffnen. Geben Sie nach der Anmeldung bei CICS beispielsweise die folgenden Befehle ein, um die Datei zu schließen bzw. zu öffnen:

- CEMT S FILE(ADNREPF0) CLOSED
- CEMT S FILE(ADNREPF0) OPEN

Das CRD-Repository für eine CICS-Testumgebung wird mit Eingabesteueranweisungen aktualisiert, für die die folgenden Syntaxregeln gelten:

- Ein Stern an erster Stelle bezeichnet eine Kommentarzeile.
- Ein Befehl "DEFINE" muss an Position 1 beginnen. Auf den Befehl muss ein Leerzeichen und dann ein gültiges Schlüsselwort, wie TRANSACTION folgen.
- Ein Schlüsselwortwert muss unmittelbar auf ein Schlüsselwort folgen. Zwischenschritte sind nicht zulässig. Die einzige Ausnahme bilden die Schlüsselwörter UPDATE, PROTECT und HIDDEN für die Anzeigeberechtigung. Sie haben keinen Wert.
- Schlüsselwortwerte werden in runde Klammern eingeschlossen.
- Ein Schlüsselwort und der zugehörigen Wert müssen in nur einer Zeile enthalten sein.

Die folgenden Beispieldefinitionen folgen der Struktur der DFHCSDUP-Befehle, wie sie im *CICS Resource Definition Guide for CICSTS* definiert sind. Als einzige Abweichung wurden die folgenden Schlüsselwörter für die Anzeigeberechtigung eingefügt, um die Attributwerte in drei Berechtigungsgruppen zusammenzufassen:

UPDATE	Auf dieses Schlüsselwort folgende Attribute können von einem Anwendungsentwickler mit Developer for System z aktualisiert werden. Dieses Schlüsselwort wird standardmäßig für übergangene Attribute verwendet.
PROTECT	Auf dieses Schlüsselwort folgende Attribute werden angezeigt, können jedoch nicht von einem Anwendungsentwickler mit Developer for System z aktualisiert werden.
HIDDEN	Auf dieses Schlüsselwort folgende Attribute werden nicht angezeigt und können nicht von einem Anwendungsentwickler mit Developer for System z aktualisiert werden.

Sehen Sie sich das folgende Codebeispiel für ADNJSPAU an.

```

//ADNJSPAU JOB <JOB-PARAMETER>
//*
//ADNSPAU EXEC PGM=ADNSPAU,REGION=1M
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//ADMREP DD DISP=OLD,DSN=FEK.#CUST.ADNREP0
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
*
* CICSplex-SM-Parameter
*
DEFINE CPSMNAME( )
*DEFINE STAGINGGROUPNAME(ADMSTAGE)
*
* Manifestexportregel
*
DEFINE MANIFESTEXPORTRULE(installOnly)
*
* Standardwerte für CICS-Ressourcendefinitionen
* Für übergangene Attribute wird standardmäßig UPDATE verwendet.
*
* Standardattribute für DB2TRAN
*
DEFINE DB2TRAN()
    UPDATE DESCRIPTION()
    ENTRY()
    TRANSID()
*
* Standardattribute für DOCTEMPLATE
*
DEFINE DOCTEMPLATE()
    UPDATE DESCRIPTION()
    TEMPLATENAME()
    FILE() TSQUEUE() TDQUEUE() PROGRAM() EXITPGM()
    DDNAME(DFHHTML) MEMBERNAME()
    HFSFILE()
    APPENDCRLF(YES) TYPE(EBCDIC)
*
* Standardattribute für FILE
*
DEFINE FILE()
    UPDATE DESCRIPTION()
    RECORDSIZE() KEYLENGTH()
    RECORDFORMAT(V) ADD(NO)
    BROWSE(NO) DELETE(NO) READ(YES) UPDATE(NO)
    REMOTESYSTEM() REMOTENAME()
    PROTECT DSNNAME() RLSACCESS(NO) LSRPOOLID(1) STRINGS(1)
    STATUS(ENABLED) OPENTIME(FIRSTREF)
    DISPOSITION(SHARE) DATABUFFERS(2) INDEXBUFFERS(1)
    TABLE(NO) MAXNUMRECS(NOLIMIT)
    READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS)
    UPDATEMODEL(LOCKING) LOAD(NO)
    JNLREAD(NONE) JOURNAL(NO)
    JNLSYNCREAD(NO) JNLUPDATE(NO)
    JNLADD(NONE) JNLSYNCSWRITE(YES)
    RECOVERY(NONE) FWDRECOVLOG(NO)
    BACKUPTYPE(STATIC)
    PASSWORD() NSRGROUP()
    CFDTPOOL() TABLENAME()

```

Abbildung 59. ADNJSPAU - CICS-Verwaltungsdienstprogramm (Teil 1 von 3)

```

*
* Standardattribute für MAPSET
*
DEFINE MAPSET()
    UPDATE  DESCRIPTION()
    PROTECT RESIDENT(NO) STATUS(ENABLED)
           USAGE(NORMAL) USELPACOPY(NO)
** Standardattribute für PROCESSTYPE
*
DEFINE PROCESSTYPE()
    UPDATE  DESCRIPTION()
           FILE(BTS)
    PROTECT STATUS(ENABLED)
           AUDITLOG() AUDITLEVEL(OFF)
*
* Standardattribute für PROGRAM
*
DEFINE PROGRAM()
    UPDATE  DESCRIPTION()
           CEDF(YES) LANGUAGE(LE370)
           REMOTESYSTEM() REMOTENAME() TRANSID()
    PROTECT API(CICSAPI) CONCURRENCY(QUASIRENT)
           DATALOCATION(ANY) DYNAMIC(NO)
           EXECCKEY(USER) EXECUTIONSET(FULLAPI)
           RELOAD(NO) RESIDENT(NO)
           STATUS(ENABLED) USAGE(NORMAL) USELPACOPY(NO)
           HIDDEN JVM(NO) JVMCLASS() JVMPROFILE(DFHJVMPR)
*
* Standardattribute für TDQUEUE
*
DEFINE TDQUEUE()
    UPDATE  DESCRIPTION()
           TYPE(INTRA)
* Partitionsexterne Parameter
    DDNAME() DSNAME()
    REMOTENAME() REMOTESYSTEM() REMOTELLENGTH(1)
    RECORDSIZE() BLOCKSIZE(0) RECORDFORMAT(UNDEFINED)
    BLOCKFORMAT() PRINTCONTROL() DISPOSITION(SHR)
* Partitionsinterne Parameter
    FACILITYID() TRANSID() TRIGERRLEVEL(1)
    USERID()
* Indirekte Parameter
    INDIRECTNAME()
    PROTECT WAIT(YES) WAITACTION(REJECT)
* Partitionsexterne Parameter
    DATABUFFERS(1)
    SYSOUTCLASS() ERROROPTION(IGNORE)
    OPENTIME(INITIAL) REWIND(LEAVE) TYPEFILE(INPUT)
* Partitionsinterne Parameter
    ATIFACILITY(TERMINAL) RECOVSTATUS(NO)

```

Abbildung 59. ADNJSPAU - CICSTS-Verwaltungsdienstprogramm (Teil 2 von 3)

```

*
* Standardattribute für TRANSACTION
*
DEFINE TRANSACTION()
    UPDATE  DESCRIPTION()
            PROGRAM()
            TWASIZE(0)
            REMOTESYSTEM() REMOTENAME() LOCALQ(NO)
    PROTECT PARTITIONSET() PROFILE(DFHCICST)
            DYNAMIC(NO) ROUTABLE(NO)
            ISOLATE(YES) STATUS(ENABLED)
            RUNAWAY(SYSTEM) STORAGECLEAR(NO)
            SHUTDOWN(DISABLED)
            TASKDATAKEY(USER) TASKDATALOC(ANY)
            BREXIT() PRIORITY(1) TRANCLASS(DFHTCL00)
            DTIMOUT(NO) RESTART(NO) SPURGE(NO) TPURGE(NO)
            DUMP(YES) TRACE(YES) CONFDATA(NO)
            OTSTIMEOUT(NO) WAIT(YES) WAITTIME(00,00,00)
            ACTION(BACKOUT) INDOUBT(BACKOUT)
            RESSEC(NO) CMDSEC(NO)
            TRPROF()
            ALIAS() TASKREQ()
            XTRANID() TPNAME() XTPNAME()
|
| *
| * URDIMAP-Attribute
| *
| DEFINE URIMAP()
|     UPDATE  USAGE(CLIENT)
|             DESCRIPTION()
|             PATH(/required/path)
|             TCPIPSERVICE()
|             TRANSACTION()
|             PROGRAM()
|     PROTECT ANALYZER(NOANALYZER)
|             ATOMSERVICE()
|             CERTIFICATE()
|             CHARACTERSET()
|             CIPHERS()
|             CONVERTER()
|             HFSFILE()
|             HOST(host.mycompany.com)
|             HOSTCODEPAGE()
|             LOCATION()
|             MEDIATYPE()
|             PIPELINE()
|             PORT(NO)
|             REDIRECTTYPE(NONE)
|             SCHEME(HTTP)
|             STATUS(ENABLED)
|             TEMPLATENAME()
|             USERID()
|             WEBSERVICE()
|
| *
| * Optionaler Dateiname für die Bindung von Dateinamen an VSAM-Dateinamen
| *
| *DEFINE DSBINDING() DSNAME()
| /*

```

Abbildung 59. ADNJSAPU - CICSTS-Verwaltungsdienstprogramm (Teil 3 von 3)

Migrationshinweise zum Verwaltungsdienstprogramm

Die Unterstützung von UIRIMAP wurde dem Verwaltungsdienstprogramm in Developer for System z Version 7.6.1 hinzugefügt. Um die Unterstützung von UIRIMAP verwenden zu können, muss der VSAM-Datei des CRD-Repositorys eine maximale Satzgröße von 3000 zugeordnet werden. Bis zur Version 7.6.1 von Developer for System z verwendet der Zuordnungsjob des CRD-Beispielrepositorys eine maximale Satzgröße von 2000.

Wenn Sie ein älteres CRD-Repository verwenden, führen Sie die folgenden Schritte aus, um die Unterstützung von UIRIMAP zu aktivieren:

1. Erstellen Sie eine Sicherung Ihres vorhandenen CRD-Repositorys, FEK.#CUST.ADNREPF0.
2. Löschen Sie das vorhandene CRD-Repository.
3. Passen Sie den Job FEK.SFEKSAMP(ADNVCRD) an und übergeben Sie ihn, um ein neues CRD-Repository zuzuordnen und zu initialisieren. Anpassungsanweisungen finden Sie in der im Member enthaltenen Dokumentation.
4. Passen Sie den Job FEK.SFEKSAMP(ADNJSPAU) an und übergeben Sie ihn, um das Verwaltungsdienstprogramm zum Füllen des neuen CRD-Repositorys zu verwenden.

Anmerkung:

- Die Migration des vorhandenen CRD-Repositorys ist nicht erforderlich, da das Verwaltungsdienstprogramm den gesamten Inhalt des CRD-Repositorys bei jeder Ausführung ersetzt.
- Für das CRD-Repository gibt es keine Versionskompatibilitätsprobleme. Der gesamte Client- und Host-Code, der von Developer for System z unterstützt wird, funktioniert mit der jeweiligen maximalen Satzgröße. Die Unterstützung von UIRIMAP wird jedoch inaktiviert, wenn die maximale Satzgröße nicht 3000 beträgt.

Nachrichten des Verwaltungsdienstprogramms

Das Verwaltungsdienstprogramm setzt die folgenden Nachrichten an die DD-Karte SYSPRINT ab. Die Nachrichten CRAZ1803E, CRAZ1891E, CRAZ1892E und CRAZ1893E enthalten Dateistatuscodes, VSAM-Rückkehrcodes, VSAM-Funktionscodes und VSAM-Rückkopplungscodes. Rückkehr-, Funktions- und Rückkopplungscodes für VSAM sind in der Veröffentlichung *DFSMS Macro Instructions for Data Sets* (IBM Form SC26-7408) dokumentiert. Dateistatuscodes sind in der Veröffentlichung *Enterprise COBOL for z/OS Language Reference* (IBM Form SC27-1408) dokumentiert.

CRAZ1800I

completed successfully on line <last control statement line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer wurde erfolgreich beendet.

Benutzeraktion: Keine

CRAZ1801W

completed with warnings on line <last control statement line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer wurde mit Warnungen beendet, die während der Verarbeitung von Steueranweisungen festgestellt wurden.

Benutzeraktion: Überprüfen Sie die weiteren Warnungen.

CRAZ1802E

encountered an error on line < line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat einen schwerwiegenden Fehler festgestellt.

Benutzeraktion: Überprüfen Sie die weiteren Warnungen.

CRAZ1803E

Repository open error, status=<file status code> RC=<VSAM return code> FC=<VSAM function code> FB=<VSAM feedback code>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat einen schwerwiegenden Fehler beim Öffnen des CRD-Repositorys festgestellt.

Benutzeraktion: Überprüfen Sie die VSAM-Statuscodes, -Rückkehr-, -Funktions- und -Rückkopplungscodes.

CRAZ1804E

Unrecognized input record on line <line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat eine nicht erkannte Eingabesteueranweisung gefunden.

Benutzeraktion: Überprüfen Sie, ob auf einen Befehl **DEFINE** ein Leerzeichen und dann das Schlüsselwort CPSMNAME, STAGINGGROUPNAME, MANIFESTEXPORTRULE, DSBINDING, DB2TRAN, DOCTEMPLATE, FILE, MAPSET, PROCESSTYPE, PROGRAM, TDQUEUE oder TRANSACTION folgt.

CRAZ1805E

Processing keyword <keyword> on line <line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer verarbeitet die Eingabesteueranweisung (Schlüsselwort **DEFINE**).

Benutzeraktion: Keine

CRAZ1806E

Invalid manifest export rule on line <line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat eine ungültige Manifestexportregel gefunden.

Benutzeraktion: Überprüfen Sie, ob der Wert des Schlüsselworts MANIFESTEXPORTRULE 'installOnly', 'exportOnly' oder 'both' lautet.

CRAZ1807E

Missing DSNNAME keyword on line <line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat eine Steueranweisung **DEFINE DSBINDING** verarbeitet, bei der das Schlüsselwort **DSNNAME** fehlt.

Benutzeraktion: Überprüfen Sie, ob die Steueranweisung **DEFINE DSBINDING** das Schlüsselwort **DSNNAME** enthält.

CRAZ1808E

Invalid keyword value for keyword <keyword> on line <line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat eine Steueranweisung **DEFINE** verarbeitet und für das benannte Schlüsselwort einen ungültigen Wert festgestellt.

Benutzeraktion: Überprüfen Sie, ob die Länge und der Wert des benannten Schlüsselworts korrekt ist.

CRAZ1890W

Keyword syntax error on line <line number>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat eine Steueranweisung DEFINE verarbeitet und für ein Schlüsselwort oder den Wert eines Schlüsselwortes einen Syntaxfehler festgestellt.

Benutzeraktion: Überprüfen Sie, ob der Wert des Schlüsselworts in runde Klammern eingeschlossen ist und unmittelbar auf das Schlüsselwort folgt. Das Schlüsselwort und der zugehörige Wert müssen sich in derselben Zeile befinden.

CRAZ1891W

**Repository duplicate key write error, status=<file status code>
RC=<VSAM return code> FC=<VSAM function code> FB=<VSAM feed-
back code>**

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat beim Schreiben in das CRD-Repository einen doppelt vorhandenen Schlüssel gefunden. Dies ist ein Fehler.

Benutzeraktion: Überprüfen Sie die VSAM-Statuscodes, -Rückkehr-, -Funktions- und -Rückkopplungscodes.

CRAZ1892W

**Repository write error, status=<file status code> RC=<VSAM return
code> FC=<VSAM function code> FB=<VSAM feedback code>**

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat beim Schreiben in das CRD-Repository einen schwerwiegenden Fehler festgestellt.

Benutzeraktion: Überprüfen Sie die VSAM-Statuscodes, -Rückkehr-, -Funktions- und -Rückkopplungscodes.

CRAZ1893W

**Repository read error, status=<file status code> RC=<VSAM return
code> FC=<VSAM function code> FB=<VSAM feedback code>**

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat einen schwerwiegenden Fehler beim Lesen des CRD-Repositorys festgestellt.

Benutzeraktion: Überprüfen Sie die VSAM-Statuscodes, -Rückkehr-, -Funktions- und -Rückkopplungscodes.

Kapitel 16. TSO-Umgebung anpassen

Dieser Anhang soll Sie beim Imitieren einer TSO-Anmeldeprozedur durch das Hinzufügen von DD-Anweisungen und Dateien zur TSO-Umgebung in Developer for System z unterstützen.

TSO Commands Service

TSO Commands Service ist die Komponente von Developer for System z, mit der TSO-Befehle und ISPF-Befehle (Batchbefehle) ausgeführt werden und die das Ergebnis an den anfordernden Client zurückgibt. Diese Befehle können implizit vom Produkt oder explizit vom Benutzer angefordert werden.

Die im Lieferumfang von Developer for System z enthaltenen Beispielmuster erstellen eine minimale TSO/ISPF-Umgebung. Falls die Entwickler in Ihrem Unternehmen den Zugriff auf angepasste Bibliotheken oder auf Bibliotheken anderer Anbieter benötigen, muss der z/OS-Systemprogrammierer zur Umgebung von TSO Commands Service die erforderlichen DD-Anweisungen und Bibliotheken hinzufügen. Die zugrunde liegende Logik ist identisch mit der des TSO-Anmeldeverfahrens, auch wenn die Implementierung in Developer for System z eine andere ist.

Anmerkung: TSO Commands Service ist ein nicht interaktives Befehlszeilentool, sodass Befehle oder Prozeduren, die die Eingabe von Daten oder die Anzeige von ISPF-Anzeigen erfordern, nicht funktionieren. Für die Ausführung derartiger Befehle/Prozeduren benötigen Sie einen 3270-Emulator wie den Host-Connect-Emulator, der im Lieferumfang der Clientkomponente von Developer for System z enthalten ist.

Zugriffsmethoden

Seit Version 7.1 bietet Developer for System z Optionen für den Zugriff auf TSO Commands Service an.

- TSO/ISPF-Client-Gateway-Service von ISPF mit einem erforderlichen ISPF-Mindestservicelevel. Dies ist die in den bereitgestellten Beispielen verwendete Standardmethode.
- Eine APPC-Transaktion (wie in den Vorgängerreleases von Version 7.1)

Anmerkung: Der TSO/ISPF-Client-Gateway-Service von ISPF ersetzt die in Version 7.1 verwendete Funktion von SCLM Developer Toolkit.

Bestimmen Sie anhand von `rsed.envvars`, welche Zugriffsmethode von Hosts ab Version 7.1 verwendet wird. Die Datei `rsed.envvars` befindet sich im Verzeichnis `/etc/rdz`, wenn bei der Konfiguration die Standardeinstellungen verwendet wurden.

- Falls die Anweisung `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` nicht vorhanden ist (oder - wie es die Standardeinstellung vorsieht - auf Kommentar gesetzt ist), wird der TSO/ISPF-Client-Gateway-Service von ISPF verwendet.
- Ist die Anweisung `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` vorhanden (und nicht auf Kommentar gesetzt), wird APPC verwendet.

TSO/ISPF-Client-Gateway als Zugriffsmethode verwenden

Basisanpassung – ISPF.conf

Die Konfigurationsdatei ISPF.conf (standardmäßig im Verzeichnis /etc/rdz/) definiert die von Developer for System z verwendete TSO/ISPF-Umgebung. Es gibt nur eine aktive Konfigurationsdatei ISPF.conf, die alle Benutzer von Developer for System z verwenden.

Im Hauptabschnitt der Konfigurationsdatei sind die DD-Namen und die zugehörigen Dateiverkettungen definiert. Sehen Sie sich dazu das folgende Beispiel an:

```
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
ispmllib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
myDD=HLQ1.LLQ1,HLQ2.LLQ2
```

- Jede DD-Definition darf nur eine Zeile umfassen (mehrere Zeilen werden nicht unterstützt). Es gibt daher keine Zeilenlängenbegrenzung.
- Bei den Definitionen muss die Groß-/Kleinschreibung nicht beachtet werden und Leerzeichen werden ignoriert.
- Kommentarzeilen beginnen mit einem Stern (*).
- Auf die DD-Namen folgt ein Gleichheitszeichen (=) und dann die Dateiverkettung. Mehrere Dateinamen sind jeweils durch ein Komma (,) voneinander getrennt.
- Dateiverkettungen werden in der Reihenfolge ihrer Auflistung durchsucht.
- Die Dateien müssen vollständig qualifiziert angegeben werden. Sie dürfen nicht in Anführungszeichen (') gesetzt sein und keine Variablen enthalten.
- Alle Dateien werden mit DISP=SHR angelegt.
- Neue DD-Namen können bei Bedarf hinzugefügt werden, müssen jedoch die Regeln (JCL) für DD-Namen befolgen und dürfen keinen Konflikt mit anderen Konfigurationsparametern in ISPF.conf hervorrufen. ISPPROF wird dynamisch vom TSO/ISPF-Client-Gateway-Service angelegt (DISP=NEW,DELETE).

Erweitert – Vorhandene ISPF-Profile verwenden

Das TSO/ISPF-Client-Gateway erstellt standardmäßig ein temporäres ISPF-Profil für TSO Commands Service. Sie können das TSO/ISPF-Client-Gateway aber auch anweisen, die Kopie eines vorhandenen ISPF-Profils zu verwenden. Der Schlüssel dazu ist die Anweisung `_RSE_CMDSERV_OPTS` in `rsed.envvars`.

```
#_RSE_CMDSERV_OPTS="$_RSE_CMDSERV_OPTS&ISPPROF=&SYSUID..ISPPROF"
```

Entfernen Sie das Kommentarzeichen für die Anweisung (indem Sie das Nummernzeichen (#) vor der Anweisung entfernen) und geben Sie den vollständig qualifizierten Dateinamen des vorhandenen ISPF-Profils an, um diese Funktion zu nutzen.

Im Dateinamen können die folgenden Variablen verwendet werden:

- `&SYSUID`. als Ersatz für die Benutzer-ID des Entwicklers
- `&SYSPREF`. als Ersatz für das TSO-Präfix des Entwicklers

Anmerkung:

- Wenn der in "ISPPROF" übergebene Dateiname ungültig ist, wird stattdessen ein temporäres leeres ISPF-Profil verwendet.
- Am Ende der Sitzung wird das ISPF-Profil (temporär oder kopiert) gelöscht. Änderungen am Profil werden nicht in das vorhandene ISPF-Profil aufgenommen.

Erweitert – Zuordnungs-Exec verwenden

Die Anwendung allocjob in ISPF.conf (die standardmäßig auf Kommentar gesetzt ist) zeigt auf eine Exec, mit der weitere Dateien nach Benutzer-ID angelegt werden können.

```
*allocjob = FEK.#CUST.CNTL(CRAISPRX)
```

Um diese Funktion zu nutzen, entfernen Sie das Kommentarzeichen für die Anweisung (indem Sie den Stern (*) vor der Anweisung entfernen) und geben Sie den vollständig qualifizierten Verweis auf die Zuordnungs-Exec an.

- Die Exec wird nach der Zuordnung von ISPPROF und der in ISPF.conf definierten DD-Anweisungen, jedoch vor der Initialisierung von ISPF ausgeführt. Stellen Sie sicher, dass Ihre Zuordnungs-Exec diese Definitionen nicht rückgängig macht.
- An die Exec wird ein Parameter übergeben (die Benutzer-ID des Aufrufenden).
- In der Beispielbibliothek FEK.#CUST.CNTL ist eine Beispiel-Exec CRAISPRX enthalten, sofern Sie bei der Anpassung und Übergabe des Jobs FEK.SFEKSAMP(FEKSETUP) keine andere Position angegeben haben. Weitere Details enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.

Anmerkung: Da die Exec vor der Initialisierung von ISPF aufgerufen wird, können Sie **VPUT** und **VGET** nicht verwenden. Sie können jedoch eine PDS(E) oder eine VSAM-Datei verwenden, um eine eigene Implementierung dieser Funktionen zu erstellen.

Erweitert – Mehrere Zuordnungs-Execs verwenden

ISPF.conf unterstützt nur den Aufruf einer Zuordnungs-Exec. Für Aufrufe weiterer Execs von dieser Exec aus gibt es jedoch keine Begrenzung. Die Benutzer-ID des Clients, die als Parameter übergeben wird, ermöglicht den Aufruf personalisierter Zuordnungs-Execs. Sie können beispielsweise prüfen, ob das Member `USERID'.EXEC(ALLOC)'` vorhanden ist, und dieses ggf. ausführen.

Mit einer ausgeklügelten Variante dieses Members können Sie die vorhandenen TSO-Anmeldeverfahren wie folgt nutzen:

- Lesen Sie eine benutzerspezifische Konfigurationsdatei, beispielsweise `USERID'.FEKPROF'`.
- Stellen Sie fest, welches Anmeldeverfahren in der Datei angegeben ist.
- Lesen Sie die angegebene Prozedur in `SYS1.PROCLIB` und führen Sie eine Syntaxanalyse der Prozedur durch, um die enthaltenen DD-Anweisungen und Dateizuordnungen zu finden.
- Legen Sie die Datei ähnlich wie im realen Anmeldeverfahren an.

Erweitert – Mehrere 'ISPF.conf'-Dateien mit mehreren Konfigurationen von Developer for System z

Falls die oben beschriebenen Szenarien mit Zuordnungs-Execs Ihren Anforderungen nicht genügen, können Sie andere Instanzen des RSE-Kommunikationsservers von Developer for System z mit jeweils einer eigenen ISPF.conf-Datei erstellen. Die folgende Methode hat im Wesentlichen den Nachteil, dass die Benutzer von Developer for System z eine Verbindung zu verschiedenen Servern auf demselben Host herstellen müssen, um die gewünschte TSO-Umgebung zu erhalten.

Anmerkung: Wenn Sie eine zweite Instanz des RSE-Servers erstellen möchten, müssen Sie nur Konfigurationsdateien, Start-JCL und die Definition gestarteter Tasks duplizieren und anschließend aktualisieren. Eine Neuinstallation des Produkts ist nicht notwendig. Es wird auch kein Code dupliziert.

```
$ cd /etc/rdz
$ mkdir /etc/rdz/tso2
$ cp rsed.envvars /etc/rdz/tso2
$ cp ISPF.conf /etc/rdz/tso2
$ ls /etc/rdz/tso2
ISPF.conf          rsed.envvars
$ oedit /etc/rdz/tso2/rsed.envvars
-> ändern: _CMDSERV_CONF_HOME=/etc/rdz/tso2
-> Kommentarzeichen entfernen und ändern: -Ddaemon.log=/var/rdz/logs/tso2
-> am ENDE hinzufügen:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
$ oedit /etc/rdz/tso2/ISPF.conf
-> ändern: nach Bedarf ändern
```

Mit den Befehlen im vorherigen Beispiel werden die Konfigurationsdateien für Developer for System z, die geändert werden müssen, in ein neu erstelltes Verzeichnis tso2 kopiert. Die Variable _CMDSERV_CONF_HOME in rsed.envvars muss aktualisiert werden, damit sie das neue Ausgangsverzeichnis in ISPF.conf definiert. Außerdem muss daemon.log aktualisiert werden, um eine neue Protokollposition zu definieren. (Die Position wird automatisch erstellt, wenn sie nicht vorhanden ist.) Mit der Aktualisierung von CLASSPATH wird sichergestellt, dass RSE die Konfigurationsdateien auffindet, die nicht nach tso2 kopiert wurden. Die Datei ISPF.conf selbst können Sie entsprechend Ihren Anforderungen aktualisieren. Beachten Sie, dass der ISPF-Arbeitsbereich (Variable _CMDSERV_WORK_HOME in rsed.envvars) von beiden Instanzen gemeinsam genutzt werden kann.

Jetzt müssen Sie nur noch eine neue gestartete Task für RSE erstellen, die eine neue Portnummer und die neuen Konfigurationsdateien in /etc/rdz/tso2 verwendet.

Weitere Informationen zu den oben gezeigten Aktionen finden Sie in den entsprechenden Abschnitten dieser Veröffentlichung.

APPC als Zugriffsmethode verwenden

Basisanpassung – JCL für APPC-Transaktion

Die Definition einer APPC-Transaktion besteht aus APPC-Parametern und Transaktions-JCL. Die Beispiel-JCL zur Erstellung einer APPC-Transaktion für Developer for System z, FEK.#CUST.JCL(FEKAPPCC), enthält zwei Optionen für das Definieren der Transaktions-JCL, mit und ohne ISPF-Unterstützung.

```
//SYSIN DD DDNAME=SYSINISP * verwenden Sie SYSINTS0 oder SYSINISP
```

Der Client ruft die in der Transaktions-JCL definierte TSO/ISPF-Umgebung ab. Befolgen Sie daher die allgemeinen DD-Regeln, wenn Sie in diesem Abschnitt die Umgebung für den Client anpassen.

```
...
//CMDSEV EXEC PGM=IKJEFT01,DYNAMBR=50,
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
//SYSPROC DD DISP=SHR,DSN=FEK.SFEKPROC
//ISPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
// SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD DUMMY
//MYDD DD DISP=SHR,DSN=HLQ1.LLQ1
// DISP=SHR,DSN=HLQ2.LLQ2
```

Anmerkung: Eine vorhandene APPC-Transaktion kann in den APPC-ISPF-Anzeigen modifiziert werden.

Erweitert – Vorhandene ISPF-Profile verwenden

Wenn Sie die ISPF-Unterstützung ausgewählt haben, erstellt Developer for System z standardmäßig ein temporäres ISPF-Profil für TSO Commands Service. Sie können Developer for System z jedoch auch anweisen, die Kopie eines vorhandenen ISPF-Profiles zu verwenden. Wie im Beispieljob FEK.SFEKSAMP(FEKAPPCC) beschrieben, müssen Sie folgende Schritte ausführen:

- Entfernen Sie das Kommentarzeichen für den COPY-Schritt in der Transaktions-JCL (EXEC und zugehörige DD-Karten).
- Passen Sie &SYSUID..ISPPROF an den Dateinamen im ISPF-Profil des Benutzers an.
- Setzen Sie die erste DD-Anweisung ISPPROF im Schritt CMDSEV auf Kommentar und entfernen Sie das Kommentarzeichen vor der zweiten DD-Anweisung.

```
...
//COPY EXEC PGM=IEBCOPY * Klonen des vorhandenen ISPF-Profiles (optional)
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=&SYSUID..ISPPROF//SYSUT2 DD DISP=(MOD,PASS),DSN=&&PROF,
// UNIT=SYSALLDA,
// LIKE=&SYSUID..ISPPROF/*
//CMDSEV EXEC PGM=IKJEFT01,DYNAMBR=50,
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
//*ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//* SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//ISPPROF DD DISP=(OLD,DELETE,DELETE),DSN=&&PROF
```

Anmerkung: Wenn ein ungültiger Dateiname verwendet wird, scheitert der Start der APPC-Transaktion (und damit von TSO Commands Service).

Erweitert – Zuordnungs-Exec verwenden

Die Beispieltransaktions-JCL ruft TSO Commands Service direkt auf, indem Sie den Namen des Service (FEKFRSRV) als Parameter an das Programm IKJEFT01 übergibt. Sie können diesen Schritt so ändern, dass eine andere Exec aufgerufen wird, die Zuordnungen ausgehend von der aktuellen Benutzer-ID vornimmt und dann TSO Commands Service aufruft.

Im Gegensatz zum TSO/ISPF-Client-Gateway als Zugriffsmethode können die im ISPF-Profil des Benutzers gespeicherten Variablen von dieser Exec genutzt werden, um die Anpassung der Umgebung zu unterstützen. Beachten Sie jedoch, dass Aktualisierungen des Profils am Ende der Sitzung verloren gehen, da Sie eine temporäre Kopie und nicht das eigentliche Profil verwenden.

Die Verwendung einer Zuordnungs-Exec in der APPC-Transaktion wird nicht unterstützt. Die obige Beschreibung wird auf "as-is"-Basis bereitgestellt.

Erweitert – Mehrere APPC-Transaktionen mit mehreren Konfigurationen von Developer for System z

Falls Sie mehrere eindeutige TSO-Umgebungen benötigen, können Sie verschiedene Instanzen des RSE-Kommunikationsservers von Developer for System z mit jeweils einer eigenen APPC-Transaktion erstellen. Die folgende Methode hat im Wesentlichen den Nachteil, dass die Benutzer von Developer for System z eine Verbindung zu verschiedenen Servern auf demselben Host herstellen müssen, um die gewünschte TSO-Umgebung zu erhalten.

Anmerkung: Wenn Sie eine zweite Instanz des RSE-Servers erstellen möchten, müssen Sie nur Konfigurationsdateien, Start-JCL und die Definition gestarteter Tasks duplizieren und anschließend aktualisieren. Eine Neuinstallation des Produkts ist nicht notwendig. Es wird auch kein Code dupliziert.

```
$ cd /etc/rdz
$ mkdir /etc/rdz/tso2
$ cp rsed.envvars /etc/rdz/tso2
$ ls /etc/rdz/tso2/
rsed.envvars
$ oedit /etc/rdz/tso2/rsed.envvars
  -> Kommentarteichen entfernen und ändern: _FEKFSCMD_TP_NAME_=FEKFTS02
  -> Kommentarteichen entfernen und ändern: -Ddaemon.log=/var/rdz/logs/tso2
  -> am ENDE hinzufügen:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

Die obigen Befehle erstellen ein neues Verzeichnis tso2 und kopieren die Konfigurationsdateien von Developer for System z, für die Änderungen erforderlich sind, an die neue Position. Die Variable `_FEKFSCMD_TP_NAME_` in `rsed.envvars` muss so aktualisiert werden, dass sie den Namen der neuen APPC-Transaktion definiert. Außerdem muss `daemon.log` aktualisiert werden, um eine neue Protokollposition für den Dämon zu definieren. (Die Position wird automatisch erstellt, wenn sie noch nicht vorhanden ist.) Mit der Aktualisierung von `CLASSPATH` wird sichergestellt, dass RSE die Konfigurationsdateien auffindet, die nicht nach tso2 kopiert wurden.

```
//FEKAPPCC JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID/*
//TPADD EXEC PGM=ATBSDFMU
//SYSSDLIB DD DISP=SHR,DSN=SYS1.APPCTP
//SYSSDOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DDNAME=SYSINISP * verwenden Sie SYSINTSO oder SYSINISP
//SYSINISP DD DATA,DLM='QT'
TPADD
TPNAME(FEKFTS02)
ACTIVE(YES)
TPSCHED_DELIMITER(DLM1)
KEEP_MESSAGE_LOG(ERROR)
MESSAGE_DATA_SET(&SYSUID..FEKFTS02.&TPDATE..&TPTIME..LOG)
DATASET_STATUS(MOD)
CLASS(A)
JCL_DELIMITER(DLM2)
//FEKFTS02 JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
/*
//CMDSERV EXEC PGM=IKJEFT01,DYNAMNBR=50,
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
//SYSPROC DD DISP=SHR,DSN=FEK.SFEKPROC
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
// SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD DUMMY
//MYDD DD DISP=SHR,DSN=HLQ1.LLQ1
// DISP=SHR,DSN=HLQ2.LLQ2
DLM2
DLM1
QT
```

Abbildung 60. FEKAPPCC - Erstellen einer zweiten APPC-Transaktion

Erstellen Sie als Nächstes eine neue APPC-Transaktion, indem Sie den Beispieljob `FEK.#CUST.JCL(FEKAPPCC)` anpassen und übergeben, wie im obigen Beispiel gezeigt wird. Zusätzlich zur normalen Anpassung (die in der JCL beschrieben ist) müssen Sie `TPNAME` in `TPNAME(FEKFTS02)` ändern, um eine Übereinstimmung mit der Definition von `_FEKFSCMD_TP_NAME_` in der neuen `rsed.envvars` zu erzielen. Sie sollten außerdem den Namen in der Variablen `MESSAGE_DATA_SET` und den JOB-Namen der Transaktions-JCL ändern.

Jetzt müssen Sie nur noch eine neue gestartete Task für RSE erstellen, die eine neue Portnummer und die neuen Konfigurationsdateien in `/etc/rdz/tso2` verwendet.

Weitere Informationen zu den oben gezeigten Aktionen finden Sie in den entsprechenden Abschnitten dieser Veröffentlichung.

Kapitel 17. Mehrere Instanzen ausführen

In bestimmten Situationen, z. B. beim Testen eines Upgrades, kann die Ausführung mehrerer aktiver Instanzen von Developer for System z auf demselben System erwünscht sein. Manche Ressourcen können jedoch nicht gemeinsam genutzt werden, z. B. TCP/IP-Ports, sodass die Standardeinstellungen nicht immer anwendbar sind. Anhand der Informationen in diesem Anhang können Sie die Koexistenz verschiedener Instanzen von Developer for System z planen, um sie dann gestützt auf dieses Konfigurationshandbuch anzupassen.

Die gemeinsame Nutzung bestimmter Komponenten von Developer for System z durch zwei (oder mehr) Instanzen ist zwar möglich, wird jedoch NUR empfohlen, wenn die Softwareversionen identisch sind und es außer Änderungen an Konfigurationsmembern keine weiteren Änderungen gibt. Developer for System z bietet genug Anpassungsspielraum für die Erstellung mehrerer Instanzen ohne Überschneidung. Wir raten Ihnen dringend, diese Anpassungsfeatures zu nutzen.

Anmerkung:

- FEK und /usr/lpp/rdz sind das während der Produktinstallation verwendete übergeordnete Qualifikationsmerkmal und der Pfad. FEK.#CUST, /etc/rdz und /var/rdz sind die während der Anpassung des Produkts verwendeten Standardpositionen. (Weitere Informationen hierzu enthält der Abschnitt „Anpassungskonfiguration“ auf Seite 15.)
- Sie sollten Developer for System z in einem privaten Dateisystem (HFS oder zFS) installieren, um das Deployment der z/OS UNIX-Produktkomponenten zu vereinfachen.
- Wenn Sie kein privates Dateisystem verwenden können, sollten Sie für den Transport der z/OS UNIX-Verzeichnisse von einem System zu einem anderen ein Archivierungstool wie den z/OS UNIX-Befehl tar nutzen. Auf diese Weise bleiben die Attribute (z. B. für die Programmsteuerung) für die Dateien und Verzeichnisse von Developer for System z erhalten.

Weitere Informationen zu den folgenden Beispielbefehlen für die Archivierung und Wiederherstellung des Installationsverzeichnis von Developer for System z enthält die Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802).

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

Identische Konfiguration in einem Sysplex

Konfigurationsdateien (und Code) für System z können auf verschiedenen Systemen in einem Sysplex gemeinsam genutzt werden. Jedes System führt dabei eine eigene identische Kopie von Developer for System z aus, wenn einige Richtlinien eingehalten werden.

- Die Protokolldateien sollten an eindeutigen Positionen gespeichert werden, damit ein System nicht die Informationen eines anderen Systems überschreibt. Indem Sie die z/OS UNIX-Protokolle mit den Anweisungen `daemon.log` und

user.log in rsed.envvars an eine bestimmte Position weiterleiten, können Sie die Konfigurationsdateien gemeinsam nutzen, wenn Sie ein systemspezifisches z/OS UNIX-Dateisystem im angegebenen Pfad anhängen. Auf diese Weise werden alle Protokolle an derselben logischen Position gespeichert. Durch das zugrunde liegende, nicht gemeinsam genutzte Dateisystem befinden sie sich allerdings an verschiedenen physischen Positionen.

- Konfigurationsverzeichnisse, wie /etc/rdz/ und /var/rdz/projects/, können im Sysplex gemeinsam genutzt werden, da Developer for System z sie nur im Lesezugriffsmodus verwendet.
- Verzeichnisse für temporäre Daten, wie /tmp/ und /var/rdz/WORKAREA/, sollten pro System eindeutig sein, da die Namen von temporären Dateien keinem Sysplex zugeordnet sind.
- Wenn Sie den Code gemeinsam nutzen, sollten Sie auch die Konfigurationsdateien gemeinsam nutzen. So stellen Sie sicher, dass nach dem Anwenden einer Wartung alle Systeme synchronisiert sind.

Identische Softwareversionen mit unterschiedlichen Konfigurationsdateien

Unter ganz bestimmten Umständen können Sie (fast) alle anpassbaren Komponenten gemeinsam nutzen. Eines der Beispiele ermöglicht für die Nutzung vor Ort den Zugriff ohne SSL und für die Nutzung an einem anderen Standort die mit SSL verschlüsselte Kommunikation.

Achtung: Mit der gemeinsam genutzten Konfiguration ist es NICHT möglich, ein Wartungsrelease, eine technische Vorschau oder ein neues Release sicher zu testen.

Wenn Sie eine andere Instanz einer aktiven Installation von Developer for System z konfigurieren möchten, führen Sie erneut die Anpassungsschritte für die Komponenten aus, die unterschiedlich sind. Verwenden Sie dazu verschiedene Dateien, Verzeichnisse und Ports, um Überschneidungen mit der aktuellen Konfiguration zu vermeiden.

In dem oben erwähnten SSL-Beispiel kann die aktuelle RSE-Dämonkonfiguration geklont werden. Im Anschluss daran können Sie die geklonte Konfiguration aktualisieren. Als Nächstes können Sie die JCL für den RSE-Dämonstart klonen und dann durch Angabe eines neuen TCP/IP-Ports und der Position der aktualisierten Konfigurationsdateien anpassen. Die MVS-Anpassungen (JES Job Monitor usw.) können von SSL-Instanzen und Nicht-SSL-Instanzen gemeinsam genutzt werden. Dies würde die folgenden Aktionen erforderlich machen:

```
$ cd /etc/rdz
$ mkdir /etc/rdz/ssl
$ cp rsed.envvars /etc/rdz/ssl
$ cp ssl.properties /etc/rdz/ssl
$ ls /etc/rdz/ssl/
rsed.envvars    ssl.properties
$ oedit /etc/rdz/ssl/rsed.envvars
-> Kommentarzeichen entfernen und ändern: -Ddaemon.log=/var/rdz/logs/ssl
    -> am ENDE hinzufügen:
    # -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
    CFG_BASE=/etc/rdz
    CLASSPATH=.:$CFG_BASE:$CLASSPATH
    # --
$ oedit /etc/rdz/ssl/ssl.properties
-> ändern: nach Bedarf ändern
```

Mit den vorangegangenen Befehlen werden die Konfigurationsdateien für Developer for System z, die geändert werden müssen, in ein neu erstelltes Verzeichnis `ssl` kopiert. Die Variable `daemon.log` in `rsed.envvars` muss aktualisiert werden, um eine neue Protokollposition zu definieren. (Die Position wird automatisch erstellt, wenn sie noch nicht vorhanden ist.) Die Aktualisierung für `CLASSPATH` stellt sicher, dass RSE die Konfigurationsdateien finden kann, die nicht nach `ssl` kopiert wurden. Die Datei `ssl.properties` selbst können Sie entsprechend Ihren Anforderungen aktualisieren.

Jetzt müssen Sie nur noch eine neue gestartete Task für RSE erstellen, die eine neue Portnummer und die neuen Konfigurationsdateien in `/etc/rdz/ssl` verwendet.

Weitere Informationen zu den oben gezeigten Aktionen finden Sie in den entsprechenden Abschnitten dieser Veröffentlichung.

Alle anderen Situationen

Wenn Codeänderungen vorgenommen werden müssen (Wartungsrelease, technische Neuentwicklungen, neues Release) oder Ihre Änderungen ziemlich komplex sind, sollten Sie Developer for System z neu installieren. In diesem Abschnitt sind mögliche Konfliktpunkte zwischen den verschiedenen Installationen beschrieben.

Die folgende Liste gibt Ihnen einen kurzen Überblick über die Elemente, die bei den Instanzen von Developer for System z unbedingt verschieden sein sollten oder müssen:

- SMP/E CSI
- Installationsbibliotheken
- TCP/IP-Port von JES Job Monitor und die zugehörige Konfigurationsdatei `FE-JJCNFG`
- Start-JCL für JES Job Monitor
- APPC-Transaktionsname
- RSE-Konfigurationsdateien, `rsed.envvars`, `*.properties` und `*.settings`
- RSE-TCP/IP-Port
- Start-JCL für RSE

Die einzelnen Elemente sind in der folgenden Übersicht detaillierter beschrieben.

- SMP/E CSI
 1. Installieren Sie jede Instanz von Developer for System z in einem separaten CSI. SMP/E verhindert eine zweite Installation derselben FMID in einem CSI, lässt jedoch die Installation einer weiteren FMID zu. Wenn es sich bei der zweiten FMID um eine neuere Version handelt, wird die vorhandene Version des Produkts gelöscht. Wenn es sich bei der zweiten FMID um eine ältere Version handelt, scheitert die Installation aufgrund doppelter Komponentennamen.
- Installationsbibliotheken
 1. Installieren Sie jede Instanz von Developer for System z in gesonderten Dateien und Verzeichnissen. Denken Sie daran, dass Sie den z/OS UNIX-Pfad nur ändern können, indem Sie den IBM Standardpfad `/usr/lpp/rdz` mit einem Präfix versehen. Ein gültiges Beispiel ist `/service/usr/lpp/rdz`.
 2. Der Konfigurationsjob für Anpassung `FEK.SFEKSAMP(FEKSETUP)` erstellt die Dateien und Verzeichnisse, in denen die Konfigurationsdateien gespeichert werden. Sie müssen diesen Job mit eindeutigen Namen für Dateien und Ver-

zeichnungen übergeben, weil die Konfigurationsdateien eindeutig sein müssen und das Überschreiben vorhandener Anpassungen vermieden werden soll.

- Obligatorische Komponenten
 1. Die Konfigurationsdatei von JES Job Monitor, FEK.#CUST.PARMLIB(FEJJCNFG), enthält die TCP/IP-Portnummer von JES Job Monitor und kann deshalb nicht gemeinsam genutzt werden. Das Member selbst kann umbenannt werden (sofern die JCL ebenfalls aktualisiert wird). Sie können somit alle angepassten Versionen dieses Members in eine Datei stellen, wenn Sie die Aktualisierungen nicht in der Installationsdatei ausführen.
 2. Die Start-JCL für JES Job Monitor, FEK.#CUST.PROCLIB(JMON), verweist auf FEJJCNFG und kann daher auch nicht gemeinsam genutzt werden. Nach der Umbenennung des Members (und - beim Starten als Benutzerjob - der JOB-Karte) können Sie die gesamte JCL in dieselbe Datei stellen.
 3. Die RSE-Konfigurationsdatei /etc/rdz/rsed.envvars enthält Verweise auf den Installationspfad und optional auf die Serverprotokollposition und muss deshalb eindeutig sein. Der Dateiname ist obligatorisch, sodass Sie die verschiedenen Kopien nicht in demselben Verzeichnis speichern können.
 4. Die Konfigurationsdatei ISPF.conf enthält einen Verweis FEK.SFEKPROC(FEKFRSRV) auf den TSO Commands-Server. Dieser ist von der Softwareversion abhängig, sodass Sie pro Instanz eine Datei ISPF.conf erstellen müssen.
 5. Alle anderen z/OS UNIX-basierten Konfigurationsdateien (z. B. *.properties) müssen in demselben Verzeichnis wie rsed.envvars enthalten sein und können nicht gemeinsam genutzt werden, weil sich rsed.envvars an einer nicht gemeinsam nutzbaren Position befinden muss.
 6. Die Start-JCL für RSE, FEK.#CUST.PROCLIB(RSED), kann nicht gemeinsam genutzt werden, da sie die TCP/IP-Portnummer definiert und auf das Installations- sowie das Konfigurationsverzeichnis verweist, die jeweils eindeutig sein müssen. Nach der Umbenennung des Members (und - beim Starten als Benutzerjob - der JOB-Karte) können Sie die gesamte JCL in dieselbe Datei stellen.
 7. Die Start-JCL für den Sperrdämon, FEK.#CUST.PROCLIB(LOCKD), kann nicht gemeinsam genutzt werden, da sie auf das Installations- sowie das Konfigurationsverzeichnis verweist, die jeweils eindeutig sein müssen. Nach der Umbenennung des Members (und - beim Starten als Benutzerjob - der JOB-Karte) können Sie die gesamten JCLs in dieselbe Datei stellen.
- Optionale Komponenten
 1. Der REXEC- und der SSH-TCP/IP-Port können ohne Einschränkungen gemeinsam genutzt werden.
 2. Die APPC-Transaktion enthält einen Verweis auf den TSO Commands-Server, FEK.SFEKPROC(FEKFRSRV). Dieser ist von der Softwareversion abhängig, sodass Sie pro Instanz eine APPC-Transaktion erstellen müssen. Denken Sie daran, dass die Variable _FEKFSCMD_TP_NAME_ in rsed.envvars definiert werden muss, weil sich der APPC-Transaktionsname geändert hat.
 3. Einige ELAXF*-Prozeduren verweisen auf die Ladebibliothek HLQ.SFEKLOAD von Developer for System z. Lesen Sie im Abschnitt „ELAXF*-Prozeduren für ferne Builderstellung“ auf Seite 26 die Anmerkung zu JCLLIB, um eine Lösung zur Bereitstellung verschiedener Sets für den Benutzer zu finden.

4. Zum Aktivieren von zwei Instanzen der gespeicherten DB2-Prozedur müssen die folgenden Tasks ausgeführt werden. Beachten Sie, dass die folgende Beschreibung ohne Unterstützung von IBM und ohne jede Gewährleistung bereitgestellt wird:
 - a. Kopieren Sie HLQ.SFEKPROC(ELAXMREX) in ein anders benanntes Member, z. B. in ELAXMRXX.
 - b. Kopieren Sie das Beispielmember HLQ.SFEKSAMP(ELAXMSAM) in ein anders benanntes Member, z. B. in ELAXMWDZ.
 - c. Ändern Sie das Beispielmember HLQ.SFEKSAMP(ELAXMJCL) so, dass es diese Namensänderungen widerspiegelt. Beispiel:


```
//SYSIN DD *
CREATE PROCEDURE SYSPROC.ELAXMRXX
  ( IN FUNCTION_REQUEST  VARCHAR(20)      CCSID EBCDIC
  ...
  , OUT RETURN_VALUE     VARCHAR(255)     CCSID EBCDIC )
PARAMETER STYLE GENERAL RESULT SETS 1
LANGUAGE REXX            EXTERNAL NAME  ELAXMRXX
COLLID DSNREXCS          WLM ENVIRONMENT ELAXMWDZ
PROGRAM TYPE MAIN        MODIFIES SQL DATA
STAY RESIDENT NO         COMMIT ON RETURN NO
ASUTIME NO LIMIT         SECURITY USER;

COMMENT ON PROCEDURE SYSPROC.ELAXMRXX IS
'PLI & COBOL PROCEDURE PROCESSOR (ELAXMRXX), INTERFACE LEVEL 0.01';

GRANT EXECUTE ON PROCEDURE SYSPROC.ELAXMRXX TO PUBLIC;
//
```
 - d. Fahren Sie mit der im Abschnitt „Gespeicherte DB2-Prozedur (optional)“ auf Seite 89 beschriebenen Anpassung fort, verwenden Sie jetzt jedoch die neuen Member.
 - e. Im Assistenten für gespeicherte DB2-Prozeduren auf dem Client muss der neue WLM-Umgebungsname (z. B. ELAXMWDZ) verwendet werden.
5. Die BIDI-Unterstützung in CICS-Regionen basiert auf einem Member der Ladebibliothek und kann daher nicht releaseübergreifend gemeinsam genutzt werden. Wenn der Name des Lademoduls jedoch für alle Instanzen derselbe ist, können Sie die neueste Version in allen Instanzen, sogar releaseübergreifend, nutzen. Die Abwärtskompatibilität ist nicht gegeben, wenn der Name des Lademoduls geändert wurde.
6. Die ADM-Lademodule (Application Deployment Manager) in CICS-Regionen sind abwärtskompatibel, sodass die neueste Version releaseübergreifend gemeinsam genutzt werden kann.
7. ADM-CRD-VSAM ist abwärtskompatibel. Die neueste Version kann demzufolge releaseübergreifend gemeinsam genutzt werden.
8. Die ADM-CICS-Ressourcendefinitionen sind abwärtskompatibel, sodass die neueste Version releaseübergreifend gemeinsam genutzt werden kann.
9. Die CARMA-VSAMs können sich in jeder Softwareversion ändern und sollten daher nicht gemeinsam genutzt werden.

Kapitel 18. Leitfaden für die Migration

Hinweise zur Migration

Dieser Abschnitt hebt die Installations- und Konfigurationsänderungen im Vergleich zu früheren Produktreleases hervor. Darüber hinaus finden Sie hier allgemeine Richtlinien für die Migration auf dieses Release. Weitere Informationen hierzu finden Sie in den entsprechenden Abschnitten dieses Handbuchs.

- Wenn Sie mit einer früheren Version von IBM Rational Developer for System z, IBM WebSphere Developer for System z, IBM WebSphere Developer für zSeries oder IBM WebSphere Studio Enterprise Developer arbeiten, sollten Sie die zugehörigen Anpassungsdateien speichern, BEVOR Sie das Upgrade auf die aktuelle Version von IBM Rational Developer for System z Version 7.6.1 installieren.
- Falls Sie planen, mehrere Instanzen von Developer for System z auszuführen, lesen Sie Kapitel 17, „Mehrere Instanzen ausführen“, auf Seite 277.

Anmerkung: Die hier aufgelisteten Migrationsinformationen beziehen sich auf Versionen von Developer for System z, die zum Zeitpunkt der Herausgabe dieser Veröffentlichung noch unterstützt wurden.

Bereits konfigurierte Dateien sichern

Wenn Sie mit einer früheren Version von Developer for System z arbeiten, sollten Sie die zugehörigen Anpassungsdateien speichern, bevor Sie die aktuelle Version von IBM Developer for System z installieren.

Anpassungsfähige Dateien von Developer for System z finden Sie an den folgenden Positionen:

- Version 7.5
 - 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 /etc/rdz/scldmt/* kopiert, wobei '*' CONFIG/, CONFIG/PROJECT/ und CONFIG/script/ entspricht.
- Version 7.1
 - FEK.SFEKSAMP
 - CRA.SCRASAMP
 - /usr/lpp/wd4z/rse/lib/, anpassungsfähige Dateien sollten nach /etc/wd4z/ kopiert werden.
- Version 7.0
 - FEK.SFEKSAMP
 - CRA.SCRASAMP
 - /usr/lpp/wd4z/rse/lib/, anpassungsfähige Dateien sollten nach /etc/wd4z/ kopiert werden.

Ältere Konfigurationen von Developer for System z dokumentieren ebenfalls Änderungen der Konfigurationsdateien anderer Produkte.

- Version 7.5

- SYS1.PARMLIB(ASCHPMxx)
Definieren Sie eine APPC-Transaktionsklasse für TSO Commands Service.
- SYS1.PARMLIB(BPXPRMxx)
Legen Sie z/OS UNIX-Systemstandardwerte fest.
- SYS1.PARMLIB(COMMNDxx)
Starten Sie die Server während des IPL.
- SYS1.PARMLIB(LPALSTxx)
Fügen Sie FEK.SFEKLPA zum LPA hinzu.
- SYS1.PARMLIB(PROGxx)
Berechtigen Sie FEK.SFEKAUTH für APF.
Fügen Sie FEK.SFEKAUTH und FEK.SFEKLOAD der LINKLIST hinzu.
- (APPC)
Definieren Sie eine APPC-Transaktion für TSO Commands Service.
- (WLM)
Ordnen Sie das APPC-Transaktionsprogramm einer TSO-Leistungsgruppe zu.
- (WLM)
Ordnen Sie eine Anwendungsumgebung für eine gespeicherte DB2-Prozedur zu.
- Version 7.1
 - SYS1.PARMLIB(ASCHPMxx)
Definieren Sie eine APPC-Transaktionsklasse für TSO Commands Service.
 - SYS1.PARMLIB(BPXPRMxx)
Legen Sie z/OS UNIX-Systemstandardwerte fest.
 - SYS1.PARMLIB(PROGxx)
Berechtigen Sie FEK.SFEKLOAD für APF.
 - /etc/services
Definieren Sie den RSE-Dämonport.
 - /etc/inetd.conf
Definieren Sie den RSE-Dämonservice.
 - /etc/SCLMDT/COMFIG/ISPF.conf
Definieren Sie die Position für den TSO Commands-Server.
 - (APPC)
Definieren Sie eine APPC-Transaktion für TSO Commands Service.
 - (WLM)
Ordnen Sie das APPC-Transaktionsprogramm einer TSO-Leistungsgruppe zu.
 - (WLM)
Ordnen Sie eine Anwendungsumgebung für eine gespeicherte DB2-Prozedur zu.
- Version 7.0
 - SYS1.PARMLIB(ASCHPMxx)
Definieren Sie eine APPC-Transaktionsklasse für TSO Commands Service.
 - SYS1.PARMLIB(BPXPRMxx)
Legen Sie z/OS UNIX-Systemstandardwerte fest.
 - SYS1.PARMLIB(PROGxx)
Berechtigen Sie FEK.SFEKLOAD für APF.

- /etc/services
Definieren Sie den RSE-Dämonport.
- /etc/inetd.conf
Definieren Sie den RSE-Dämonservice.
- (APPC)
Definieren Sie eine APPC-Transaktion für TSO Commands Service.
- (WLM)
Ordnen Sie das APPC-Transaktionsprogramm einer TSO-Leistungsgruppe zu.
- (WLM)
Ordnen Sie eine Anwendungsumgebung für eine gespeicherte DB2-Prozedur zu.

Migrationshinweise für Version 7.6.1

Die folgenden Migrationshinweise sind spezifisch für Version 7.6.1. Sie gelten für eine Migration von Version 7.6 oder als Zusätze zu den vorhandenen Migrationshinweisen für Version 7.6.

- Application Deployment Manager - Vorhandene ADN*-Module in der CICS-RPL-Verknüpfung müssen aktualisiert werden.
- Application Deployment Manager - Die folgenden Beispielmembers wurden aktualisiert, um dem Verwaltungsdienstprogramm die Unterstützung von URIMAP hinzuzufügen.
 - ADNJSPAU
 - ADNVCRD
- Application Deployment Manager - Eine vorhandene VSAM-Datei des CRD-Repositorys muss ersetzt werden, um die Unterstützung von URIMAP zu aktivieren.
- CARMA - Unterstützung von Layout mit variabler Länge für VSAM-Dateien für angepasste CARMA-Informationen, CRASTRS
- CARMA - Neue Beispielmembers sind hinzugekommen:
 - CRA#VS2 - CRASTRS auf Format mit variabler Länge migrieren
- JES Job Monitor - Verwendung von "_CEE_ENVFILE_S" in der gestarteten JCL-Task
- JES Job Monitor - Die folgenden FEJJCNFG-Anweisungen sind jetzt optional:
 - HOST_CODEPAGE
- PROCLIB - Neue PROCLIB-Members sind hinzugekommen.
 - ELAXFDCL
- RSE - 64-Bit-Java-Version wird jetzt unterstützt.
- RSE - Neue Bedienerbefehle sind hinzugekommen (seit Version 7.6.1.0):
 - MODIFY DISPLAY PROCESS,DETAIL
- RSE - Die folgenden nicht anpassbaren Anweisungen wurden geändert oder sind neu in rsed.envvars (seit Version 7.6.0.0):
 - (_RSE_JAVAOPTS) -DDSTORE_KEEPALIVE_RESPONSE_TIMEOUT
 - (_RSE_JAVAOPTS) -DDSTORE_IO_SOCKET_READ_TIMEOUT
 - (_RSE_JAVAOPTS) -DRSECOMM_LOGFILE_MAX
- RSE - Neue optionale Anweisungen wurden zu rsed.envvars hinzugefügt (seit Version 7.6.0.0 und 7.6.0.1):
 - (_RSE_JAVAOPTS) -Denable.automount

- | – (_RSE_JAVAOPTS) -Ddeny.nozero.port
- | – (_RSE_JAVAOPTS) -Dsingle.logon
- | – (_RSE_JAVAOPTS) -Dprocess.cleanup.interval
- | • RSE - Die folgenden Konsolnachrichten wurden geändert oder sind neu (seit
- | Version 7.6.0.1 und 7.6.1.0):
- | – FEK001I
- | – FEK210I

Migration von Version 7.5 auf Version 7.6

IBM Rational Developer for System z, FMID HHOP760

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben daher FEK.* und /usr/lpp/rdz/*.
- Application Deployment Manager - Vorhandene ADN*-Module in der CICS-RPL-Verknüpfung müssen aktualisiert werden.
- Application Deployment Manager - Neue Lademodule, die Teil der CICS-RPL-Verknüpfung sein müssen, wurden zur Unterstützung der CICS-RESTful-Schnittstelle hinzugefügt.
 - ADNANAL
 - ADNCRD41
 - ADNREST
- Application Deployment Manager - Neue Beispielmembers wurden zur Unterstützung der CICS-RESTful-Schnittstelle hinzugefügt.
 - ADNCSDRS
 - ADNCSDTX
 - ADNTXNC
- Application Deployment Manager - Vorhandene Beispielmembers wurden umbenannt.
 - ADNARCSD -> ADNCSDAR
 - ADNCMSGH -> ADNMSGHC
 - ADNMFEST -> ADNVMFST
 - ADNPCCSD -> ADNCSDWS
 - ADNSMSGH -> ADNMSGHS
 - ADNVSAM -> ADNVCRD
- Ein neuer Produktions-RAM wird für den Zugriff auf CA Endevor® SCM bereitgestellt.
 - CRARNDVR
- CARMA - Neue Beispielmembers wurden für die Unterstützung des CA Endevor® SCM-RAMs bereitgestellt.
 - FEK.#CUST.JCL(CRA#VCAD)
 - FEK.#CUST.JCL(CRA#VCAS)
 - FEK.#CUST.CNTL(CRASUBCA)
 - FEK.#CUST.PARMLIB(CRASHOW)
 - FEK.#CUST.PARMLIB(CRATMAP)
 - FEK.SFEKPROC(CRANDVRA)
 - /etc/rdz/crstart.endevor.conf

- CARMA - Neue Beispielmuster wurden zur Unterstützung der Zusammenführung von RAM-Definitionen hinzugefügt.
 - CRA#UADD
 - CRA#UQRY
- File Manager-Integration - Die Schnittstelle für Stapelverarbeitung für den Zugriff auf File Manager wird nicht länger unterstützt.
- File Manager-Integration - Die Konfigurationsdatei FMIEXT.properties wurde vollständig geändert und muss ersetzt werden.
- JES Job Monitor - Im FEJJMON-Lademodul sind ab Version 7.5.0.1 LE-Optionen eingebettet. Dies erfordert möglicherweise Definitionsänderungen für Ihre gestarteten Tasks. Weitere Details hierzu finden Sie in der Beispiel-JCL FEK.SFEKSAMP-(FEJJJCL).
- JES Job Monitor - FEJJCENFG wurden neue optionale Anweisungen hinzugefügt (Version 7.5.0.1 und 7.5.1.0).
 - APPLID
 - CONSOLE_NAME
 - GEN_CONSOLE_NAME
- JES Job Monitor - Der neue Befehl 'Show JCL' wird ab Version 7.5.1.0 unterstützt. Dies erfordert möglicherweise eine Aktualisierung Ihrer Sicherheitssoftware.
- Sperrdämon – Der Sperrdämon (LOCKD) ist ab Version 7.5.0.1 eine neue gestartete Task. Möglicherweise wird diese gestartete Task abgefragt, um den Client von Developer for System z zu ermitteln, der eine Dateisperre enthält. (Systembefehle stoppen auf der Adressraumbene, die dem RSE-Thread-Pool entspricht.)
- SCLMDT - Die Standardposition für die Konfigurationsdateien von SCLMDT-Projekte wurde geändert.
 - /var/rdz/sclmdt
- RSE - Neue Bedienerbefehle sind hinzugekommen.
 - MODIFY RSESTANDARDLOG
- RSE - Neue erforderliche Anweisungen wurden in rsed.envvars (in Version 7.5.0.1 und 7.6.0.0) hinzugefügt.
 - _RSE_LOCKD_PORT
 - (_RSE_JAVAOPTS) -Dlock.daemon.port
 - (_RSE_JAVAOPTS) -Dlock.daemon.cleanup.interval
 - _RSE_LOCKD_CLASS
 - _RSE_HOST_CODEPAGE
 - (_RSE_JAVAOPTS) -Dfile.encoding
 - (_RSE_JAVAOPTS) -Dconsole.encoding
- RSE - Neue optionale Anweisungen wurden in rsed.envvars (seit Version 7.5.0.1, 7.5.1.0 und 7.6.0.0) hinzugefügt.
 - (_RSE_JAVAOPTS) -Duser.log
 - (_RSE_JAVAOPTS) -Dkeep.last.log
 - (_RSE_JAVAOPTS) -Denable.standard.log
 - (_RSE_JAVAOPTS) -DDSTORE_LOG_DIRECTORY
 - (_RSE_JAVAOPTS) -DHIDE_ZOS_UNIX
 - (_RSE_JAVAOPTS) -Denable.certificate.mapping
 - GSK_CRL_SECURITY_LEVEL
 - GSK_LDAP_SERVER
 - GSK_LDAP_PORT

- GSK_LDAP_USER
- GSK_LDAP_PASSWORD
- RSE - In rsed.envvars wurden einige optionale Anweisungen geändert.
 - (_RSE_JAVAOPTS) -Ddaemon.log
 - (_RSE_JAVAOPTS) -Xmx
 - SCLMDT_CONF_HOME
- RSE - Der Datei ssl.properties wurden neue optionale Anweisungen hinzugefügt (ab Version 7.5.1.0 und 7.6.0.0).
 - server_keystore_label
 - server_keystore_type
- RSE - Der RSE-Dämon unterstützt ab Version 7.5.1.0 die X.509-Clientzertifikats-authentifizierung. Dies erfordert bei Verwendung eine Aktualisierung Ihrer aktuellen Zertifikats- und Sicherheitskonfigurationen.
- RSE - Die Sicherheit wurde erhöht. Verbindungsanforderungen schlagen bei PassTicket- und FEKAPPL-Fehlern fehl.
- RSE - Die Standardposition für alle Protokolldateien (Dämon- und Benutzerprotokolle) wurde geändert.
 - /var/rdz/logs
 - /var/rdz/logs/\$LOGNAME
- RSE - Eine neue Beispiel-JCL wurde bereitgestellt, um Protokoll- und Konfigurationsdaten für Developer for System z zusammenzustellen.
 - FEKLOGS

Konfigurierbare Dateien

Tabelle 47 gibt einen Überblick über Dateien, die in Version 7.6 angepasst werden. Die Beispielbibliotheken von Developer for System z, FEK.SFEKSAMP, FEK.SFEKSAMV und /usr/lpp/rdz/samples/, enthalten mehr als die hier aufgelisteten anpassbaren Member. Sie enthalten z. B. auch CARMA-Beispielquellcode und Jobs für die Kompilierung.

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

Tabelle 47. Anpassungen bei Version 7.6

Member/Datei	Standardposition	Zweck	Migrationshinweise
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für die Erstellung von Dateien und Verzeichnissen und zum Füllen derselben mit anpassbaren Dateien	Aktualisiert zum Einschließen neuer anpassbarer Member
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JCL für JES Job Monitor	Hinzugefügte Option zum Ändern der LE-Optionen
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Name des JMON-Members bei Lieferung	Siehe JMON-Member
RSED	FEK.SFEKSAMP(FEKRSDD) [FEK.#CUST.PROCLIB]	JCL für den RSE-Dämon	Keine
FEKRSDD	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Name des RSED-Members bei Lieferung	Siehe RSED-Member

Tabelle 47. Anpassungen bei Version 7.6 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
LOCKD	FEK.SFEKSAMP (FEKLOCKD) [FEK.#CUST.PROCLIB]	JCL für Sperrdämon	NEU, Anpassung erforderlich
FEKLOCKD	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(LOCKD)]	Name des LOCKD-Members bei Lieferung	Siehe LOCKD-Member
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL für ferne Projektbuilds usw.	Keine
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für Sicherheitsdefinitionen	Kleinere Aktualisierungen
FEJJCNGF	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Konfigurationsdatei für JES Job Monitor	Neue optionale Anweisungen sind hinzugekommen.
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL für TSO-Übergabe	Keine
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für CARMA-Nachrichten	Keine
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für CARMA-Konfiguration	Keine
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für angepasste CARMA-Informationen	Keine
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CLIST für CARMA-Batchstart	Keine
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CLIST für CARMA-Batchstart für den CA Endevor® SCM-RAM	NEU, Anpassung optional
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für den CA Endevor® SCM-RAM	NEU, Anpassung optional
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CARMA-Konfiguration für den CA Endevor® SCM-RAM	NEU, Anpassung optional
CRANDVRA	FEK.SFEKPROC	CARMA-Zuordnungs-REXX für den CA Endevor® SCM-RAM	NEU, Anpassung optional
CRAISPRX	FEK.SFEKSAMP [FEK.#CUST.CNTL]	DD-Beispielzuordnungs-Exec für CARMA bei Verwendung des TSO/ISPF-Client-Gateways	Keine
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für SCLM-RAM-Nachrichten	Keine
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der SCLM-RAM-Dateien	Keine
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM für PDS-RAM-Nachrichten	Keine

Tabelle 47. Anpassungen bei Version 7.6 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
CRA#CRAM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung des Skeleton-RAM	Keine
CRA#VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der CARMA-Konfigurationsdatei (VSAM) für den CA Endevor® SCM-RAM	NEU, Anpassung optional
CRA#VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung der VSAM-Datei für angepasste CARMA-Informationen für den CA Endevor® SCM-RAM	NEU, Anpassung optional
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Zusammenführen von RAM-Definitionen	NEU, Anpassung optional
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Extrahieren von RAM-Definitionen	NEU, Anpassung optional
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Beispiel Quellcode für die Ersetzung von IRXJCL	Keine
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung von CRAXJCL	Keine
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des RESTful-CRD-Servers für die primäre CICS-Region	NEU, Anpassung optional
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren von alternativen Transaktions-IDs für die primäre CICS-Region	NEU, Anpassung optional
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Erstellen von alternativen Transaktions-IDs	NEU, Anpassung optional
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung von ADNMSGHS	Umbenannt, hieß ADNCMSGH
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOLE]	Beispiel Quellcode für den Pipelinennachrichtenhandler	Umbenannt, hieß ADNSMSGH
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung des CRD-Repositorys	Umbenannt, hieß ADNVSAM
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des Web-Service-CRD-Servers für die primäre CICS-Region	Umbenannt, hieß ADNPCCSD
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des CRD-Servers für nicht primäre CICS-Regionen	Umbenannt, hieß ADNARCSD
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Aktualisierung der CRD-Standardwerte	Für den RESTful-Service wurden Definitionen hinzugefügt. Dies erfordert eine neue Anpassung.

Tabelle 47. Anpassungen bei Version 7.6 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
ADNMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung und zum Definieren des Manifestrepositorys	Umbenannt, hieß ADNMFEST
ELAXMSAM	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL-Prozedur des WLM-Adressraums für den Stored Procedure Builder für PL/I und COBOL	Keine
ELAXMJCL	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des Stored Procedure Builder für COBOL und PL/I für DB2	Keine
FEKAPPC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Erstellen einer APPC-Transaktion	Keine
FEKAPPCL	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Anzeigen einer APPC-Transaktion	Keine
FEKAPPCX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Löschen einer APPC-Transaktion	Keine
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Erfassen von Protokolldateien	NEU, Anpassung optional
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-Umgebungsvariablen	Ältere Kopien müssen durch diese ersetzt werden. (Anschließend müssen die Anpassungen erneut vorgenommen werden.)
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF-Client-Gateway, Konfigurationsdatei	ISP.SISPCLIB wurde SYSPROC für SCLMDT hinzugefügt.
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei	Keine
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei für die Verwendung von CRASTART	Keine
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei für die Verwendung von CRASTART für den CA Endevor® SCM-RAM	NEU, Anpassung optional
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-SSL-Konfigurationsdatei	Neue optionale Anweisungen sind hinzugekommen.
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-Tracekonfigurationsdatei	Keine
propertiescfg.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für hostbasierte Eigenschaftsgruppen	Keine

Tabelle 47. Anpassungen bei Version 7.6 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
projectcfg.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für hostbasierte Projekte	Keine
FMIEXT.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für File Manager-Integration	Ältere Kopien müssen durch diese ersetzt werden. (Anschließend müssen die Anpassungen erneut vorgenommen werden.)
uchars.settings	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für nicht editierbare Zeichen	Keine

Migration von Version 7.1 auf Version 7.5

IBM Rational Developer for System z, FMID HHOP750

- Die SMP/E-Standardinstallationsposition für MVS-Komponenten wurde nicht geändert. Sie bleibt daher FEK.*.
- Die SMP/E-Standardinstallationsposition für z/OS UNIX-Komponenten wurde in /usr/lpp/rdz/* geändert.
- Common Access Repository Manager (CARMA) wurde mit Developer for System z Version 7.5 zusammengeführt, sodass Sie CARMA nicht als gesondertes Produkt installieren müssen.
- SCLM Developer Toolkit wurde mit Developer for System z Version 7.5 zusammengeführt und muss nicht mehr als gesondertes Produkt installiert werden.
- In Version 7.5 ersetzt der TSO/ISPF-Client-Gateway-Service von ISPF die in Version 7.1 verwendete Funktion von SCLM Developer Toolkit für die Verbindung zu TSO Commands Service. Die APPC-Verbindungsmethode wird weiterhin unterstützt.
- In Version 7.5 ist der RSE-Server eine gestartete Task und kein von INETD verwalteter Prozess mehr. Der RSE-Server funktioniert jetzt auch nach einem Einzelservermodell. In früheren Versionen gab es dagegen für jede Client-Host-Verbindung einen privaten RSE-Server.
- Alle Module, die eine APF-Berechtigung erfordern (JES Job Monitor und SCLM Developer Toolkit), wurden in Version 7.5 in FEK.SFEKAUTH verschoben und erfordern eine Aktualisierung der vorhandenen APF-Definitionen.
- Das Lademodul von JES Job Monitor wurde in Version 7.5 in FEK.SFEKAUTH verschoben, sodass die Prozedur der vorhandenen gestarteten Task aktualisiert werden muss.
- CARMA-Lademodule wurden in neue Bibliotheken verschoben. Dies erfordert eine Aktualisierung des vorhandenen Scripts CRASUBMT für den Serverstart.
- Lademodule von SCLM Developer Toolkit wurden in neue Bibliotheken verschoben, was eine Aktualisierung der vorhandenen LINKLIST-Definitionen erfordert.
- ELAXFTS0 ist eine neue Build-Beispielprozedur, die es seit Version 7.1.1 gibt. ELAXFCP1 und ELAXFPP1 sind seit Version 7.5 neu.
- Die Datei uchars.settings ist eine neue Konfigurationsdatei für nicht editierbare Zeichen.

- Die Datei `propertiescfg.properties` ist eine neue Konfigurationsdatei für Standardmerkmalgruppen.
- `FEJJCNFG`, `CRASRV.properties` und `FMIEXT.properties` enthalten neue optionale Anweisungen.
- In Version 7.5 wurde `rsed.envvars` geändert und muss ersetzt werden.
- Die zu Version 7.5 gelieferte Beispieldatei `ISPF.conf` ist mit der von SCLM Developer Toolkit in Version 7.1 verwendeten Datei vergleichbar.
- Einige der Anpassungen, die an Application Deployment Manager vorgenommen wurden, müssen erneut ausgeführt werden.
- Application Deployment Manager stellt neue Funktionen bereit, die eine Anpassung erfordern.
- Die Sicherheitseinstellungen für den RSE-Server haben sich in Version 7.5 drastisch geändert.
- Das Sicherheitsprofil `MVS.MCSOPER.JMON` ist in Version 7.5 neu für JES Job Monitor hinzugekommen.
- Das CARMA-Startscript wurde umbenannt und an eine neue Position verschoben. Dies erfordert eine Aktualisierung der vorhandenen Konfigurationsdatei `CRASRV.properties`.
- Das FMI-Startscript wurde umbenannt und an eine neue Position verschoben. Dies erfordert eine Aktualisierung der vorhandenen Konfigurationsdatei `FMIEXT.properties`.
- Zur bidirektionalen Unterstützung wurde ein neues Lademodul hinzugefügt. Dies erfordert eine Aktualisierung der vorhandenen CICS-DFHRPL-Verkettung, wenn Sie nicht die Bibliothek `FEK.SFEKLOAD` verwenden.
- Änderungen der `MAXPROCUSER`-Parameter der `SYS1.PARMLIB(BPXPRMxx)` werden nun ebenfalls dokumentiert.

Konfigurierbare Dateien

Tabelle 48 gibt einen Überblick über Dateien, die in Version 7.5 angepasst werden. Die Beispielbibliotheken von Developer for System z, `FEK.SFEKSAMP`, `FEK.SFEKSAMV` und `/usr/lpp/rdz/samples/`, enthalten mehr als die hier aufgelisteten anpassbaren Member. Sie enthalten z. B. auch CARMA-Beispielquellcode und Jobs für die Kompilierung.

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

Tabelle 48. Anpassungen bei Version 7.5

Member/Datei	Standardposition	Zweck	Migrationshinweise
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für die Erstellung von Dateien und Verzeichnissen und zum Füllen derselben mit anpassbaren Dateien	NEU, Anpassung erforderlich
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JCL für JES Job Monitor	STEPLIB in SFEKAUTH geändert
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	JCL für den RSE-Dämon	NEU, Anpassung erforderlich

Tabelle 48. Anpassungen bei Version 7.5 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL für ferne Projektbuilds usw.	ELAXFTSO, ELAXFCP1 und ELAXFPP1 sind neu.
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL für Sicherheitsdefinitionen	NEU, erforderlich
FEJJCNG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Konfigurationsdatei für JES Job Monitor	<ul style="list-style-type: none"> Einige Anweisungen sind jetzt optional. Neue optionale Anweisungen sind hinzugekommen.
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL für TSO-Übergabe	Der Jobname kann jetzt eine Variable sein.
CRAISPRX	FEK.SFEKSAMP [FEK.#CUST.CNTL]	DD-Beispielzuordnungs-Exec für CARMA bei Verwendung des TSO/ISPF-Client-Gateways	NEU, Anpassung optional
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Beispiel Quellcode für die Ersetzung von IRXJCL	NEU, Anpassung optional
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Kompilierung von CRAXJCL	NEU, Anpassung optional
ADNSMSGH	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Beispiel Quellcode für den Pipelinennachrichtenhandler	Ältere Kopien müssen durch diese ersetzt werden. (Anschließend müssen die Anpassungen erneut vorgenommen werden.)
ADNPCCSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zum Definieren des CRD-Servers für die primäre CICS-Region	Ältere Kopien müssen durch diese ersetzt werden. (Anschließend müssen die Anpassungen erneut vorgenommen werden.)
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Aktualisierung der CRD-Standardwerte	NEU, Anpassung optional
ADNMFEST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL zur Erstellung und zum Definieren des Manifestrepositorys	NEU, Anpassung optional

Tabelle 48. Anpassungen bei Version 7.5 (Forts.)

Member/Datei	Standardposition	Zweck	Migrationshinweise
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE-Umgebungsvariablen	Ältere Kopien müssen durch diese ersetzt werden. (Anschließend müssen die Anpassungen erneut vorgenommen werden.)
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF-Client-Gateway, Konfigurationsdatei	Identisch mit der Datei ISPF.conf in SCLMDT Version 7.1
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei	<ul style="list-style-type: none"> • Position und Name des Start-Scripts wurden geändert. • Neue optionale Anweisungen sind hinzugekommen.
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA-Konfigurationsdatei für die Verwendung von CRASTART	NEU, Anpassung optional
FMIEXT.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für File Manager-Integration	<ul style="list-style-type: none"> • Position und Name des Start-Scripts wurden geändert. • Neue optionale Anweisungen sind hinzugekommen.
uchars.settings	/usr/lpp/rdz/samples/ [/etc/rdz/]	Konfigurationsdatei für nicht editierbare Zeichen	NEU, Anpassung optional

Migration von Version 7.0 auf Version 7.1

IBM Rational Developer for System z, FMID HHOP710

- Die SMP/E-Standardinstallationspositionen für MVS- und z/OS UNIX-Komponenten wurden nicht geändert. Sie bleiben daher FEK.* und /usr/lpp/wd4z/*.
- **Ergänzung:** Konfigurationsoption - Ausführung von TSO/ISPF-Befehlen über eine APPC-Transaktion oder über SCLM Developer Toolkit
- **Änderung:** Die APPC-Transaktion nutzt ein neues ISPF-Feature.
- **Ergänzung:** Die folgenden anpassbaren Member sind neu hinzugekommen:
 - SAMPLIB ELAXFADT
 - SAMPLIB ADNCMSGH
 - /usr/lpp/wd4z/rse/lib/FMIEXT.properties

- **Änderung:** Die folgenden Member wurden verschoben:
 - SFEKDLL(FEJBDTRX) -> SFEKLOAD(FEJBDTRX)
- **Änderung:** Die folgenden anpassbaren Member wurden geändert:
 - SAMPLIB FEKFAPPCC
 - /usr/lpp/wd4z/rse/lib/rsed.envvars
 - /usr/lpp/wd4z/rse/lib/setup.env.zseries
 - /usr/lpp/wd4z/rse/lib/server.zseries

IBM Common Access Repository Manager (CARMA), FMID HCMA710

- Die SMP/E-Standardinstallationsposition für MVS-Komponenten wurde nicht geändert. Sie bleibt daher CRA.*.
- **Änderung:** Protokollaufzeichnungen werden in die DD-Anweisung CARMA-LOG geschrieben.
- **Änderung:** Die VSAM für CARMA-Nachrichten (CRAMSG) und die VSAM für Konfiguration wurden aktualisiert.
- **Ergänzung:** Die folgenden anpassbaren Member sind neu hinzugekommen:
 - SAMPLIB CRA#ECOB
 - SAMPLIB CRA#EPDS
 - SAMPLIB CRA#ERAM
 - SAMPLIB CRA#ESLM
- **Umbenennung:** Die folgenden anpassbaren Member wurden umbenannt:
 - SAMPLIB CRAREPR -> CRA\$VDEF
 - SAMPLIB CRAMREPR -> CRA\$VMSG
 - SAMPLIB CRASREPR -> CRA\$VSTR
 - SAMPLIB CRASALX -> CRA#ASLM
 - SAMPLIB CRACOBJ1 -> CRA#CCB1
 - SAMPLIB CRACOBJ2 -> CRA#CCB2
 - SAMPLIB CRACLICM -> CRA#CCLT
 - SAMPLIB CRARAMCS -> CRA#CPDS
 - SAMPLIB CRARAMCM -> CRA#CRAM
 - SAMPLIB CRATREPR -> CRA#VPDS
 - SAMPLIB CRALREPR -> CRA#VSLM
 - SAMPLIB CRACLIRN -> CRA#XCLT
- **Änderung:** Die folgenden anpassbaren Member wurden geändert:
 - CLIST CRASUBMT

Konfigurierbare Dateien

Tabelle 23 gibt einen Überblick über Dateien, die in Version 7.1 angepasst werden. Die Beispielbibliotheken von CARMA und Developer for System z, CRA.SCRASAMP, FEK.SFEKSAMP und /usr/lpp/wd4z/rse/lib/, enthalten mehr als die hier aufgelisteten anpassbaren Member. Sie enthalten z. B. auch CARMA-Beispiel Quellcode und Jobs für die Kompilierung.

Tabelle 49. Anpassungen bei Version 7.1

Member/Datei	Standardposition	Zweck	Migrationshinweise
ELAXF*	FEK.SFEKSAMP	JCL für ferne Projektbuilds und sonstige Jobs	ELAXFADT ist neu.
CRA\$VMSG	CRA.SCRASAMP	JCL zur Erstellung der VSAM für CARMA-Nachrichten	<ul style="list-style-type: none"> • Umbenannt, hieß CRAMREPR • Die durch diesen Job erstellte VSAM ist aktualisiert.
CRA\$VDEF	CRA.SCRASAMP	JCL zur Erstellung der VSAM für CARMA-Konfiguration	<ul style="list-style-type: none"> • Umbenannt, hieß CRAREPR • Die durch diesen Job erstellte VSAM ist aktualisiert.
CRA\$VSTR	CRA.SCRASAMP	JCL zur Erstellung der VSAM für angepasste CARMA-Informationen	Umbenannt, hieß CRASREPR
CRASUBMT	CRA.SCRASAMP	CLIST für CARMA-Batchstart	DD-Anweisung CARMALOG hinzufügen
CRA#VSLM	CRA.SCRASAMP	JCL zur Erstellung der VSAM für SCLM-RAM-Nachrichten	Umbenannt, hieß CRALREPR
CRA#ASLM	CRA.SCRASAMP	JCL zur Erstellung der SCLM-RAM-Dateien	Umbenannt, hieß CRASALX
CRA#VPDS	CRA.SCRASAMP	JCL zur Erstellung der VSAM für PDS-RAM-Nachrichten	Umbenannt, hieß CRATREPR
CRA#CRAM	CRA.SCRASAMP	JCL zur Kompilierung des Skeleton-RAM	Umbenannt, hieß CRARAMCM
FEKAPPCC	FEK.SFEKSAMP	JCL zum Erstellen einer APPC-Transaktion	Unterstützung von NEST in ISPF nutzen
rsed.envvars	/usr/lpp/wd4z/rse/lib/ [/etc/wd4z/]	RSE-Umgebungsvariablen	Ältere Kopien müssen durch diese ersetzt werden. (Anschließend müssen die Anpassungen erneut vorgenommen werden.)
FMEXT.properties	/usr/lpp/wd4z/rse/lib/ [/etc/wd4z/]	Konfigurationsdatei für File Manager-Integration	NEU, Anpassung bei Nutzung erforderlich

Anhang A. SSL- und X.509-Authentifizierung konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von SSL (Secure Sockets Layer) oder beim Überprüfen und/oder Modifizieren einer vorhandenen Konfiguration auftreten könnten. Dieser Anhang stellt auch eine Beispielkonfiguration zur Verfügung, um Benutzer zu unterstützen, die sich mit einem X.509-Zertifikat selbst authentifizieren.

Sichere Kommunikation bedeutet, dass Ihr DFV-Partner derjenige ist, der er zu sein vorgibt, und dass Informationen in einer Weise übertragen werden, die es anderen erschwert, die Daten abzufangen und zu lesen. SSL bietet diese Fähigkeiten für ein TCP/IP-Netz an. SSL verwendet digitale Zertifikate für Ihre Identifikation und ein Protokoll mit öffentlichen Schlüsseln, um die Kommunikation zu verschlüsseln. Weitere Informationen zu digitalen Zertifikaten und zu dem von SSL verwendeten Protokoll mit öffentlichen Schlüsseln finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Welche Aktionen erforderlich sind, um die SSL-Kommunikation für Developer for System z zu konfigurieren, hängt von den genauen Anforderungen am jeweiligen Standort, vom verwendeten RSE-Kommunikationsverfahren und von den am Standort verfügbaren Ressourcen ab.

In diesem Anhang werden Sie die aktuellen RSE-Definitionen klonen, damit Sie eine zweite RSE-Dämonverbindung haben, die SSL verwendet. Außerdem werden Sie Ihre eigenen Sicherheitszertifikate erstellen, die von den verschiedenen Teilnehmern der RSE-Verbindung verwendet werden.

- „Speicherpositionen für private Schlüssel und Zertifikate festlegen“ auf Seite 300
- „Schlüsseldatei mit RACF erstellen“ auf Seite 301
- „Vorhandene RSE-Konfiguration klonen“ auf Seite 303
- „Koexistenz durch Aktualisieren von rsed.envvars aktivieren“ auf Seite 303
- „Aktualisierung von ssl.properties durchführen, um SSL zu aktivieren“ auf Seite 303
- „Neuen RSE-Dämon erstellen, um SSL zu aktivieren“ auf Seite 304
- „Verbindung testen“ auf Seite 305
- „Unterstützung der X.509-Clientauthentifizierung hinzufügen (optional)“ auf Seite 307
- „Schlüsseldatenbank mit gskkyman erstellen (optional)“ auf Seite 308
- „Keystore mit keytool erstellen (optional)“ auf Seite 311

In diesem Anhang wird die folgende einheitliche Namenskonvention verwendet:

- Zertifikat: rdzrse
- Schlüssel- und Zertifikatspeicher: rdzssl.*
- Kennwort: rsessl
- Benutzer-ID für Dämon: stcrse

Für einige der nachfolgenden Tasks wird vorausgesetzt, dass Sie aktivierter z/OS UNIX-Benutzer sind. Zum Aktivieren können Sie den TSO-Befehl **OMVS** absetzen. Mit dem Befehl **exit** können Sie zu TSO zurückkehren.

Speicherpositionen für private Schlüssel und Zertifikate festlegen

Die von SSL verwendeten Identitätszertifikate und Schlüssel für die Verschlüsselung/Entschlüsselung werden in einer Schlüsseldatei gespeichert. Die jeweiligen Implementierungen dieser Schlüsseldatei sind vom Anwendungstyp abhängig.

Alle Implementierungen folgen jedoch dem gleichen Prinzip. Ein Befehl generiert ein Schlüsselpaar (einen öffentlichen Schlüssel und einen zugehörigen privaten Schlüssel). Anschließend wird der öffentliche Schlüssel in ein selbst signiertes Zertifikat (X.509) eingeschlossen, das als Zertifikatskette mit einem Element gespeichert wird. Diese Zertifikatskette und der private Schlüssel werden als ein (mit einem Aliasnamen bezeichneter) Eintrag in einer Schlüsseldatei gespeichert.

Der RSE-Dämon ist eine System SSL-Anwendung und verwendet eine Schlüsseldatenbankdatei. Diese Schlüsseldatenbank kann eine von gskkyman erstellte physische Datei oder eine von Ihrer SAF-kompatiblen Sicherheitssoftware (z. B. RACF) verwaltete Schlüsseldatei sein. Der (vom Dämon gestartete) RSE-Server ist eine Java-SSL-Anwendung und verwendet eine von keytool erstellte Keystoredatei oder eine Schlüsseldatei, die von Ihrer Sicherheitssoftware verwaltet wird.

Tabelle 50. Mechanismen für den SSL-Zertifikatsspeicher

Zertifikatsspeicher	Erstellt und verwaltet von	RSE-Dämon	RSE-Server
Schlüsseldatei	SAF-kompatibles Sicherheitsprodukt	unterstützt	unterstützt
Schlüsseldatenbank	gskkyman in z/OS UNIX	unterstützt	/
Keystore	Java-Keytool	/	unterstützt

Für die Verbindung über SSL benötigen Sie den Keystore und die Schlüsseldatenbank (als z/OS UNIX-Datei oder als SAF-kompatible Schlüsseldatei):

- Keystore (RACF oder keytool)
- Schlüsseldatenbank (RACF oder gskkyman)

Anmerkung:

- Für die Verwaltung von Zertifikaten sind SAF-kompatible Schlüsseldateien die bevorzugte Methode.
- Ein gemeinsam genutztes Zertifikat kann verwendet werden, wenn der RSE-Dämon und der RSE-Server dieselbe Zertifikatsverwaltungsmethode verwenden.
- Der RSE-Dämon muss programmgesteuert ausgeführt werden. Die Verwendung von System SSL impliziert, dass SYS1.SIEALNKE von Ihrer Sicherheitssoftware programmgesteuert eingestellt wurde.
- Für die Ausführung einer System SSL-Anwendung (Dämonverbindung) muss SYS1.SIEALNKE in der LINKLIST oder STEPLIB enthalten sein. Wenn Sie die STEPLIB-Methode bevorzugen, fügen Sie am Ende von rsed.envvars die folgende Anweisung hinzu.

STEPLIB=\$STEPLIB:SYS1.SIEALNKE

Beachten Sie jedoch Folgendes:

- 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. Bibliothe-

ken verlieren ihre APF-Berechtigung, wenn sie mit STEPLIB-Bibliotheken ohne APF-Berechtigung gemischt werden.

- System SSL verwendet ICSF (Cryptographic Service Facility), sofern diese Serviceeinrichtung verfügbar ist. ICSF stellt Unterstützung für Hardwareverschlüsselung bereit und wird anstelle der System SSL-Softwarealgorithmen verwendet. Weitere Informationen hierzu enthält die Veröffentlichung *System SSL Programming* (IBM Form SC24-5901).

Weitere Informationen zu RACF und digitalen Zertifikaten finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683). Die Dokumentation zu gskkyman ist in der Veröffentlichung *System SSL Programming* (IBM Form SC24-5901) enthalten. Die Dokumentation zu keytool ist unter <http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html> verfügbar.

Schlüsseldatei mit RACF erstellen

Führen Sie diesen Schritt nicht aus, wenn Sie gskkyman zum Erstellen der RSE-Dämonschlüsseldatenbank und keytool zum Erstellen des RSE-Server-Keystores verwenden.

Der Befehl **RACDCERT** installiert und verwaltet private Schlüssel und Zertifikate in RACF. RACF unterstützt die Verwaltung mehrerer privater Schlüssel und Zertifikate in einer Gruppe. Diese Gruppen werden als Schlüsseldateien bezeichnet.

Details zum Befehl **RACDCERT** enthält die Veröffentlichung *Security Server RACF Command Language Reference* (IBM Form SA22-7687).

```
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ACCESS(READ) ID(stcrse)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ACCESS(READ) ID(stcrse)
SETROPTS RACLIST(FACILITY) REFRESH
```

```
RACDCERT ID(stcrse) GENCERT SUBJECTSDN(CN('rdz rse ssl') +
OU('rdz') O('IBM') L('Raleigh') SP('NC') C('US')) +
NOTAFTER(DATE(2017-05-21)) WITHLABEL('rdzrse') KEYUSAGE(HANDSHAKE)
```

```
RACDCERT ID(stcrse) ADDRING(rdzssl.racf)
RACDCERT ID(stcrse) CONNECT(LABEL('rdzrse') RING(rdzssl.racf) +
DEFAULT USAGE(PERSONAL))
```

Das obige Beispiel beginnt mit der Erstellung der notwendigen Profile und dem Berechtigen der Benutzer-ID STCRSE für den Zugriff auf die Schlüsseldateien und auf Zertifikate, deren Eigner diese Benutzer-ID ist. Die Benutzer-ID muss mit der für die Ausführung des SSL-RSE-Dämons verwendeten Benutzer-ID übereinstimmen. Der nächste Schritt ist die Erstellung eines neuen, selbst signierten Zertifikats mit der Bezeichnung rdzrse. Es ist kein Kennwort erforderlich. Dieses Zertifikat wird dann einer neu erstellten Schlüsseldatei (rdzssl.racf) hinzugefügt. Für die Schlüsseldatei ist ebenso wie für das Zertifikat kein Kennwort erforderlich.

Das Ergebnis können Sie wie folgt mit der Option list überprüfen:

```
RACDCERT ID(stcrse) LIST
Digital certificate information for user STCRSE:
```

```
Label: rdzrse
Certificate ID: 2QjW10Xi0sXZ1aaEqZmihUBA
Status: TRUST
Start Date: 2007/05/24 00:00:00
```

```

End Date: 2017/05/21 23:59:59
Serial Number:
>00<
Issuer's Name:
>CN=rdz rse ssl.OU=rdz.O=IBM.L=Raleigh.SP=NC.C=US<
Subject's Name:
>CN=rdz rse ssl.OU=rdz.O=IBM.L=Raleigh.SP=NC.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
  Ring Owner: STCRSE
  Ring:
>rdzssl.racf<

```

Signiertes Zertifikat verwenden (optional)

Zertifikate können selbst signiert oder von einer Zertifizierungsstelle (CA) signiert sein. Bei einem von einer CA signierten Zertifikat garantiert die CA, dass der Eigentümer des Zertifikats derjenige ist, der er zu sein vorgibt. Durch den Signierungsprozess werden Ihrem Zertifikat die Berechtigungsnachweise der CA hinzugefügt (hierbei handelt es sich auch um ein Zertifikat). Dadurch wird es zu einer mehrteiligen Zertifikatskette.

Wenn Sie ein Zertifikat verwenden, das von einer Zertifizierungsstelle signiert wurde, können Sie Fragen zur Vertrauensprüfung durch den Client von Developer for System z vermeiden, wenn der Client der Zertifizierungsstelle bereits vertraut.

Zum Erstellen und Verwenden eines von einer CA signierten Zertifikats führen Sie die folgenden Schritte aus:

1. Erstellen Sie ein selbst signiertes Zertifikat.
RACDCERT ID(stcrse) GENCERT WITHLABEL('rdzrse') . . .
 2. Erstellen Sie für dieses Zertifikat eine Signierungsanforderung.
RACDCERT ID(stcrse) GENREQ (LABEL('rdzrse')) DSN(dsn)
 3. Senden Sie die Zertifizierungsanforderung an die von Ihnen gewählte Zertifizierungsstelle.
 4. Prüfen Sie, ob die Berechtigungsnachweise der CA (ebenfalls ein Zertifikat) bereits bekannt sind.
RACDCERT CERTAUTH LIST
 5. Markieren Sie das CA-Zertifikat als vertrauenswürdig.
RACDCERT CERTAUTH ALTER(LABEL('CA cert')) TRUST
Fügen Sie alternativ der Datenbank das CA-Zertifikat hinzu.
RACDCERT CERTAUTH ADD(dsn) WITHLABEL('CA cert') TRUST
 6. Fügen Sie der Datenbank das signierte Zertifikat hinzu. Dieses ersetzt das selbst signierte Zertifikat.
RACDCERT ID(stcrse) ADD(dsn) WITHLABEL('rdzrse') TRUST
- Anmerkung:** Löschen Sie NICHT das selbst signierte Zertifikat, bevor es ersetzt wurde. Dadurch verlieren Sie den privaten Schlüssel, der mit dem Zertifikat verbunden ist und das Zertifikat wird unbrauchbar.
7. Erstellen Sie eine Schlüsseldatei.
RACDCERT ID(stcrse) ADDRING(rdzssl.racf)
 8. Fügen Sie der Schlüsseldatei das signierte Zertifikat hinzu.
RACDCERT ID(stcrse) CONNECT(ID(stcrse) LABEL('rdzrse'))
RING(rdzssl.racf)

9. Fügen Sie der Schlüsseldatei das von der CA signierte Zertifikat hinzu.

```
RACDCERT ID(stcrse) CONNECT(CERTAUTH LABEL('CA cert'))
RING(rdzssl.racf))
```

Vorhandene RSE-Konfiguration klonen

In diesem Schritt wird eine neue Instanz der RSE-Konfigurationsdateien erstellt, damit die SSL-Konfiguration parallel mit den vorhandenen Instanzen ausgeführt werden kann. Bei den folgenden Beispielbefehlen wird davon ausgegangen, dass sich die Konfigurationsdateien in /etc/rdz/ befinden. Dies ist die im Abschnitt „Anpassungskonfiguration“ auf Seite 15 verwendete Standardposition.

```
$ cd /etc/rdz
$ mkdir ssl
$ cp rsed.envvars ssl
$ cp ssl.properties ssl
$ ls ssl
rsed.envvars    ssl.properties
```

Die oben aufgeführten z/OS UNIX-Befehle erstellen ein Unterverzeichnis mit der Bezeichnung ssl und füllen es mit den Konfigurationsdateien, für die Änderungen erforderlich sind. Die anderen Konfigurationsdateien, das Installationsverzeichnis und die MVS-Komponenten können gemeinsam genutzt werden, weil sie nicht SSL-spezifisch sind.

Indem die meisten vorhandenen Konfigurationsdateien wiederverwendet werden, kann der Fokus auf die Änderungen gelegt werden, die zur Konfiguration von SSL tatsächlich erforderlich sind. Außerdem kann eine erneute vollständige RSE-Konfiguration vermieden werden. (Beispielsweise muss für ISPF.conf keine neue Position definiert werden.)

Koexistenz durch Aktualisieren von rsed.envvars aktivieren

Bisher sind die Definitionen eine exakte Kopie der aktuellen Konfiguration. Dies impliziert, dass die Protokolle des neuen RSE-Dämons die aktuellen Serverprotokolldateien überschreiben. RSE muss auch die Positionen kennen, an denen die Konfigurationsdateien auffindbar sind, die nicht in das ssl-Verzeichnis kopiert wurden. Beide Probleme können sie durch geringfügige Änderungen an rsed.envvars lösen.

```
$ oedit /etc/rdz/ssl/rsed.envvars
-> Kommentarzeichen entfernen und ändern: -Ddaemon.log=/var/rdz/logs/ssl
-> am ENDE hinzufügen:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

Die oben beschriebenen Änderungen definieren eine neue Protokollposition. (Wenn die Protokollposition nicht vorhanden ist, wird diese vom RSE-Dämon erstellt.) Durch die Änderungen wird auch der CLASSPATH aktualisiert, sodass die SSL-RSE-Prozesse zunächst das aktuelle Verzeichnis (/etc/rdz/ssl) und dann das Ursprungsverzeichnis (/etc/rdz) nach Konfigurationsdateien durchsucht.

Aktualisierung von ssl.properties durchführen, um SSL zu aktivieren

Durch die Aktualisierung der Datei ssl.properties wird RSE angewiesen, die Kommunikation mit SSL zu verschlüsseln.


```
$ oedit /etc/rdz/ssl/ssl.properties
-> ändern: enable_ssl=true
-> Kommentarzeichen entfernen und ändern: daemon_keydb_file=rdzssl.racf
-> Kommentarzeichen entfernen und ändern: daemon_key_label=rdzrse
-> Kommentarzeichen entfernen und ändern: server_keystore_file=rdzssl.racf
-> Kommentarzeichen entfernen und ändern: server_keystore_label=rdzrse
-> Kommentarzeichen entfernen und ändern: server_keystore_type=JCERACFKS
```

Die oben beschriebenen Änderungen aktivieren SSL und teilen dem RSE-Dämon und RSE-Server mit, dass ihr (gemeinsam genutztes) Zertifikat in der Schlüsseldatei `rdzssl.racf` unter der Bezeichnung `rdzrse` gespeichert ist. Mit dem Schlüsselwort `JCERACFKS` wird dem RSE-Server mitgeteilt, dass eine SAF-kompatible Schlüsseldatei als Schlüsselspeicher verwendet wird.

Neuen RSE-Dämon erstellen, um SSL zu aktivieren

Wie bereits angegeben werden wir eine zweite Verbindung erstellen, die SSL verwendet. Dafür muss ein neuer RSE-Dämon erstellt werden. Der RSE-Dämon kann eine gestartete Task oder ein Benutzerjob sein. Für die anfängliche Testkonfiguration werden wir einen Benutzerjob verwenden. Bei den folgenden Anweisungen wird davon ausgegangen, dass die Beispiel-JCL in `FEK.#CUST.PROCLIB(RSED)` enthalten ist. Dies ist die im Abschnitt „Anpassungskonfiguration“ auf Seite 15 verwendete Standardposition:

1. Erstellen Sie ein neues Member `FEK.#CUST.PROCLIB(RSEDSSL)` und kopieren Sie die Beispiel-JCL `FEK.#CUST.PROCLIB(RSED)` in dieses Member.
2. Passen Sie `RSEDSSL` an, indem Sie am Anfang eine Jobkarte und am Ende eine `EXEC`-Anweisung hinzufügen. Geben Sie außerdem eine neue Portnummer (4047) und die Position der SSL-bezogenen Konfigurationsdateien (`/etc/rdz/ssl`) an. Vergleichen Sie hierzu das folgende Codebeispiel. Beachten Sie, dass die Verwendung der Benutzer-ID `STCRSE` zwingend ist, weil dieser Benutzer-ID in einem vorherigen Schritt die entsprechende Zugriffsberechtigung auf Zertifikate und Schlüsseldateien erteilt wurde.

```
//RSEDSSL JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),USER=STCRSE
//*
/* RSE-DÄMON - SSL
/*
//RSED      PROC IVP='',                                * 'IVP' für einen IVP-Test
//          PORT=4047,
//          HOME='/usr/lpp/rdz',
//          CNFG='/etc/rdz/ssl'
//
//RSE       EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
//          PARM='PGM &HOME./bin/rsed.sh &IVP &PORT &CNFG'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//PEND
//
//RSED      EXEC RSED
//
```

Abbildung 61. *RSEDSSL - RSE-Dämonbenutzerjob für SSL*

Anmerkung: Die Benutzer-ID, die dem Job `RSEDSSL` zugeordnet ist, muss über dieselben Berechtigungen verfügen wie der ursprüngliche RSE-Dämon. Das `FACILITY`-Profil `BPX.SERVER` und das `PTKDATA`-Profil `IRRPTAUTH.FEKAPPL.*` stellen hierbei die Schlüsselemente dar.

Verbindung testen

Die SSL-Hostkonfiguration ist vollständig, und der RSE-Dämon für SSL kann mit der Übergabe des zuvor erstellten Jobs FEK.#CUST.PROCLIB(RSESSL) gestartet werden.

Die neue Konfiguration kann jetzt getestet werden, indem eine Verbindung mit dem Client mit Developer for System z hergestellt wird. Da Sie für SSL eine neue Konfiguration (durch Klonen der vorhandenen Konfiguration) erstellt haben, müssen Sie nun auf dem Client eine neue Verbindung mit dem Port 4047 für den RSE-Dämon konfigurieren.

Wenn die Verbindung hergestellt ist, beginnen Host und Client mit dem Handshakeverfahren, um einen sicheren Pfad einzurichten. Im Rahmen dieses Handshakeverfahrens werden Zertifikate ausgetauscht. Wenn die Clientkomponente von Developer for System z das Hostzertifikat oder die signierende CA nicht erkennt, fragt sie beim Benutzer an, ob dieses Zertifikat vertrauenswürdig ist.

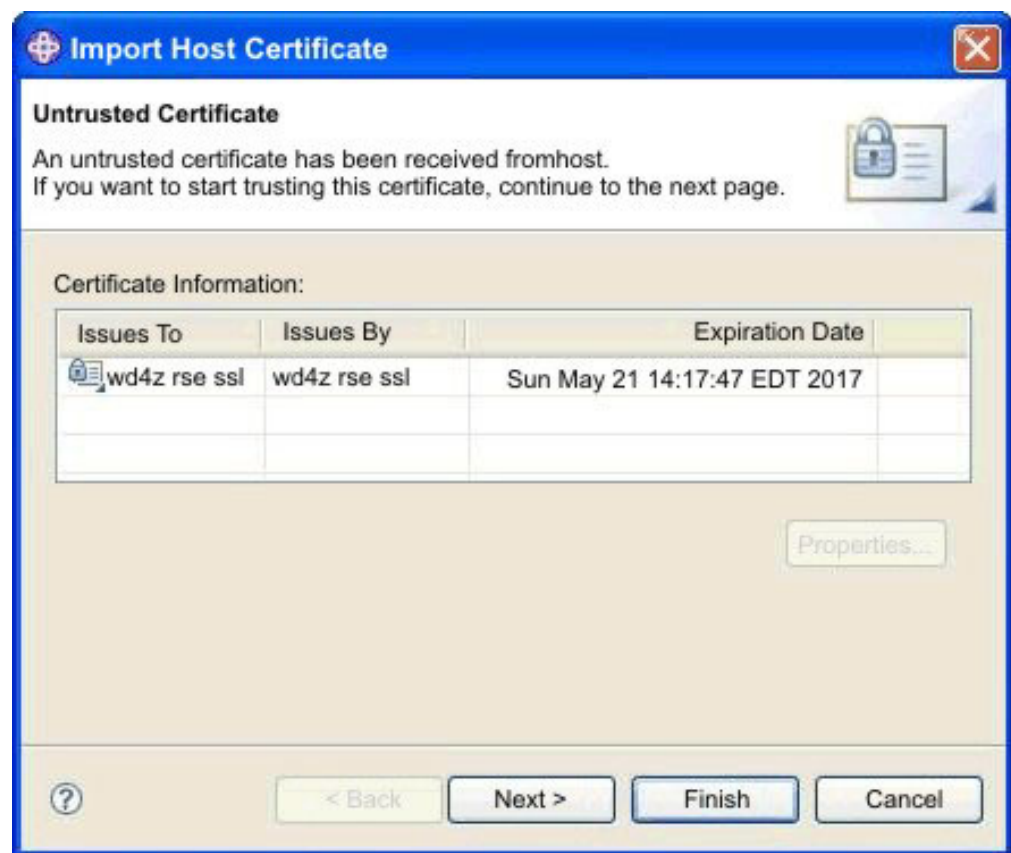


Abbildung 62. Dialog 'Hostzertifikat importieren'

Der Benutzer kann dieses Zertifikat als vertrauenswürdig akzeptieren, indem er auf die Schaltfläche 'Finish' klickt. Danach wird die Verbindungsinitialisierung fortgesetzt.

Anmerkung: Möglicherweise verwenden der RSE-Dämon und der RSE-Server zwei verschiedene Zertifikatspositionen. Daraus ergeben sich zwei verschiedene Zertifikate und somit auch zwei Bestätigungen.

Wenn der Client ein Zertifikat einmal anerkannt hat, wird dieser Dialog nicht mehr angezeigt. Die Liste vertrauenswürdiger Zertifikate kann verwaltet werden. Wählen Sie dazu **Fenster > Benutzervorgaben... > Ferne Systeme > SSL** aus, um den folgenden Dialog aufzurufen:

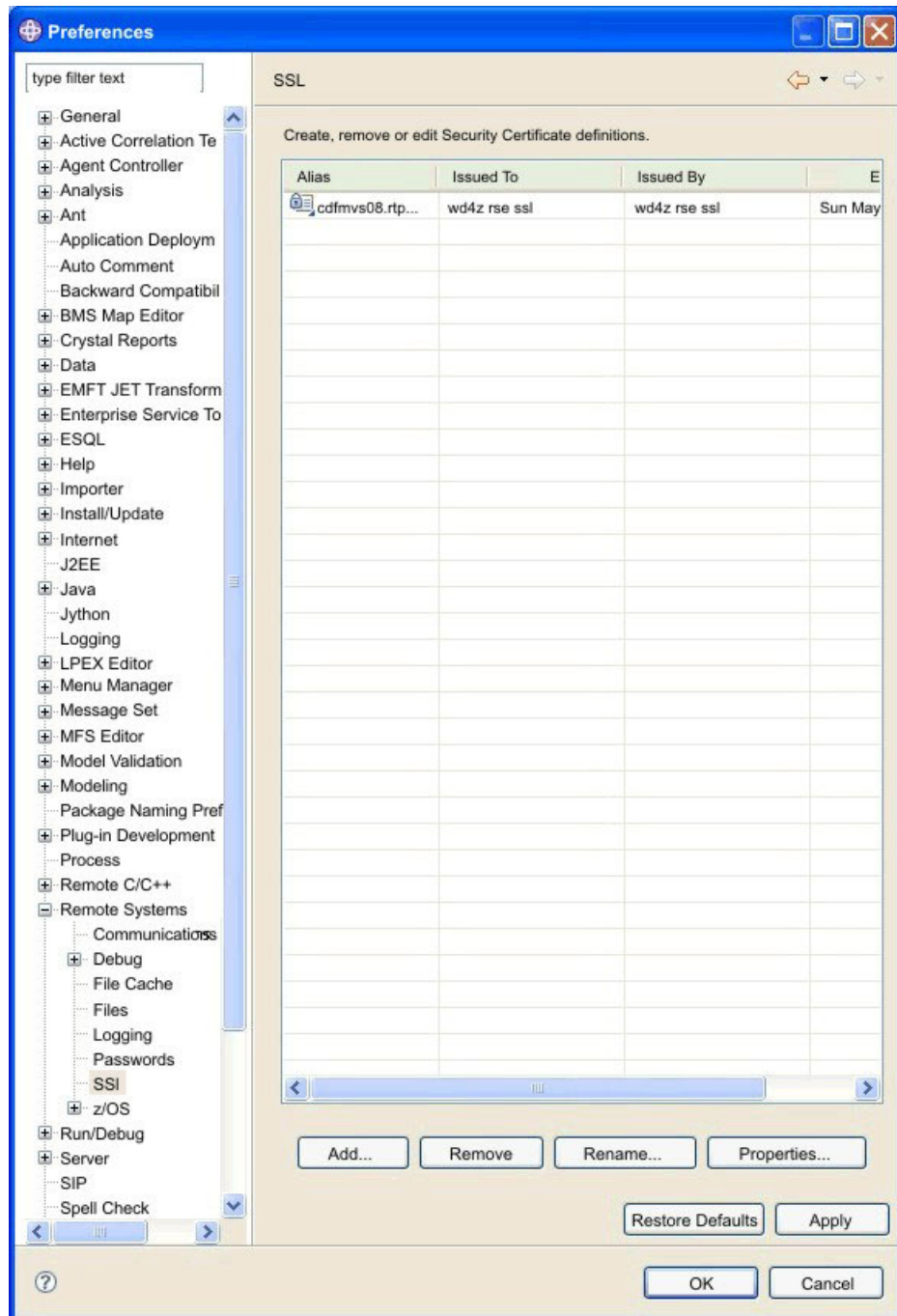


Abbildung 63. Vorgabendialog - SSL

Wenn die SSL-Kommunikation fehlschlägt, gibt der Client eine Fehlermeldung zurück. Weitere Informationen sind in den verschiedenen Server- und Benutzerprotokolldateien verfügbar. Lesen Sie die diesbezüglichen Beschreibungen in den Abschnitten „Protokollierung des RSE-Dämons und des Thread-Pools“ auf Seite 139 und „Protokollierung des RSE-Benutzers“ auf Seite 140.

Unterstützung der X.509-Clientauthentifizierung hinzufügen (optional)

Mit einem X.509-Zertifikat unterstützt der RSE-Dämon die eigene Authentifizierung der Benutzer. Voraussetzung hierfür ist die Verwendung der mit SSL verschlüsselten Kommunikation, da dies eine Erweiterung der Hostauthentifizierung mit einem in SSL verwendeten Zertifikat ist.

Es gibt mehrere Wege zur Zertifikatsauthentifizierung für einen Benutzer. Lesen Sie hierzu den Abschnitt „Clientauthentifizierung unter Verwendung von X.509-Zertifikaten“ auf Seite 171. Die folgenden Schritte dokumentieren die Konfiguration, die zur Unterstützung der Methode erforderlich ist, bei der Ihre Sicherheitssoftware das Zertifikat unter Verwendung der HostIdMappings-Zertifikatserweiterung authentifiziert.

1. Ändern Sie das Zertifikat, das die Zertifizierungsstelle (CA) identifiziert, die zum Signieren des Clientzertifikats verwendet wird, in ein sehr vertrauenswürdiges CA-Zertifikat. Auch wenn der Status TRUST für die Zertifikatsüberprüfung ausreichend ist, wird er durch den Status HIGHTRUST ersetzt, da dieser im Rahmen des Anmeldeprozesses zur Zertifikatsauthentifizierung verwendet wird.

```
RACDCERT CERTAUTH ALTER(LABEL('HighTrust CA')) HIGHTRUST
```

2. Fügen Sie der Schlüsseldatei rdzssl.racf das von der CA signierte Zertifikat hinzu, damit es zur Überprüfung der Clientzertifikate verfügbar ist.

```
RACDCERT ID(stcrse) CONNECT(CERTAUTH LABEL('HighTrust CA') +  
RING(rdzssl.racf))
```

Dies beendet die Konfiguration der Sicherheitssoftware für das CA-Zertifikat.

3. Definieren Sie in der Klasse SERVAUTH eine Ressource (Format IRR.HOST.hostname) für den Hostnamen CDFMVS08.RALEIGH.IBM.COM, der in der HostIdMappings-Erweiterung Ihres Clientzertifikats definiert ist.

```
RDEFINE SERVAUTH IRR.HOST.CDFMVS08.RALEIGH.IBM.COM UACC(NONE)
```

4. Gewähren Sie der Benutzer-ID der gestarteten RSE-Task STCRSE Zugriff mit LESEBERECHTIGUNG auf diese Ressource.

```
PERMIT IRR.HOST.CDFMVS08.RALEIGH.IBM.COM CLASS(SERVAUTH) +  
ACCESS(READ) ID(stcrse)
```

5. Aktivieren Sie die Änderungen in der SERVAUTH-Klasse. Verwenden Sie den ersten Befehl, wenn die SERVAUTH-Klasse noch nicht aktiviert ist. Verwenden Sie den zweiten Befehl, um eine aktive Konfiguration zu aktualisieren.

```
SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)  
oder  
SETROPTS RACLIST(SERVAUTH) REFRESH
```

Dies beendet die Konfiguration der Sicherheitssoftware für die HostIdMappings-Erweiterung.

6. Starten Sie die gestartete RSE-Task erneut, um von nun an Clientanmeldungen mit X.509-Zertifikaten zu akzeptieren.

Schlüsseldatenbank mit gskkyman erstellen (optional)

Führen Sie diesen Schritt nicht aus, wenn Sie eine SAF-kompatible Schlüsseldatei für die Schlüsseldatenbank des RSE-Dämons verwenden.

gskkyman ist ein shellbasiertes, menügeführtes z/OS UNIX-Programm, das eine z/OS UNIX-Datei erstellt, mit Daten füllt und verwaltet. Diese Datei enthält private Schlüssel, Zertifikatanforderungen und Zertifikate und wird als Schlüsseldatenbank bezeichnet.

Anmerkung: Die Umgebung für gskkyman muss möglicherweise mit den folgenden Anweisungen konfiguriert werden. Weitere Informationen hierzu enthält die Veröffentlichung *System SSL Programming* (IBM Form SC24-5901).

```
PATH=$PATH:/usr/lpp/gskssl/bin
export NLSPATH=/usr/lpp/gskssl/lib/nls/msg/En_US.IBM-1047/%N:$NLSPATH
export STEPLIB=$STEPLIB:SYS1.SIEALNKE
```

```
$ cd /etc/rdz/ssl
$ gskkyman          Database Menu
```

1 - Create new database

```
Enter option number: 1
Enter key database name (press ENTER to return to menu): rdzssl.kdb
Enter database password (press ENTER to return to menu): rsessl
Re-enter database password: rsessl
Enter password expiration in days (press ENTER for no expiration):
Enter database record length (press ENTER to use 2500):
```

Key database /etc/rdz/ssl/rdzssl.kdb created.

Press ENTER to continue.

Key Management Menu

6 - Create a self-signed certificate

Enter option number (press ENTER to return to previous menu): **6**

Certificate Type

5 - User or server certificate with 1024-bit RSA key

```
Select certificate type (press ENTER to return to menu): 5
Enter label (press ENTER to return to menu): rdzrse
Enter subject name for certificate
  Common name (required): rdz rse ssl
  Organizational unit (optional): rdz
Organization (required): IBM
  City/Locality (optional): Raleigh
  State/Province (optional): NC
  Country/Region (2 characters - required): US
Enter number of days certificate will be valid (default 365): 3650
```

Enter 1 to specify subject alternate names or 0 to continue: **0**

Please wait

Certificate created.

Press ENTER to continue.

Key Management Menu

0 - Exit program

Enter option number (press ENTER to return to previous menu): 0

\$ ls -l rdzssl.*

total 152

-rw----- 1 IBMUSER SYS1 35080 May 24 14:24 rdzssl.kdb

-rw----- 1 IBMUSER SYS1 80 May 24 14:24 rdzssl.rdb

\$ chmod 644 rdzssl.*

\$ ls -l rdzssl.*

-rw-r--r-- 1 IBMUSER SYS1 35080 May 24 14:24 rdzssl.kdb

-rw-r--r-- 1 IBMUSER SYS1 80 May 24 14:24 rdzssl.rdb

Das obige Beispiel beginnt mit der Erstellung einer Schlüsseldatenbank `rdzssl.kdb` mit dem Kennwort `rsessl`. Wenn die Datenbank vorhanden ist, wird sie mit Daten gefüllt. Dazu wird ein neues selbst signiertes Zertifikat erstellt, das für ca. 10 Jahre gültig ist (ohne Berücksichtigung des zusätzlichen Tages in Schaltjahren). Das Zertifikat wird unter der Bezeichnung `rdzrse` und mit dem bereits für die Schlüsseldatenbank verwendeten Kennwort (`rsessl`) gespeichert. (Dies ist eine RSE-Anforderung.)

`gskkyman` legt die Schlüsseldatenbank mit einer (sehr sicheren) Bitmaske (600 Berechtigungsbits) an, die nur dem Eigner Zugriff gewährt. Die Berechtigungen müssen weniger restriktiv gesetzt werden, sofern der Dämon nicht dieselbe Benutzer-ID wie der Ersteller der Schlüsseldatenbank verwendet. 644 (Eigner mit Lese-/Schreibzugriff; Lesezugriff für alle übrigen Benutzer) ist eine verwendbare Maske für den Befehl `chmod`.

Das Ergebnis können Sie wie folgt überprüfen, indem Sie im Untermenü **Manage keys and certificates** die Option **Show certificate information** auswählen:

\$ gskkyman

Database Menu

2 - Open database

Enter option number: 2

Enter key database name (press ENTER to return to menu): **rdzssl.kdb**

Enter database password (press ENTER to return to menu): **rsessl**

Key Management Menu

1 - Manage keys and certificates

Enter option number (press ENTER to return to previous menu): 1

Key and Certificate List

1 - rdzrse

Enter label number (ENTER to return to selection menu, p for previous list): 1

Key and Certificate Menu

1 - Show certificate information

Enter option number (press ENTER to return to previous menu): 1

Certificate Information

Label: rdzrse

Record ID: 14

Issuer Record ID: 14

Trusted: Yes

```

        Version: 3
    Serial number: 45356379000ac997
        Issuer name: rdz rse ssl
                    rdz
                    IBM
                    Raleigh
                    NC
                    US
    Subject name: rdz rse ssl
                rdz
                IBM
                Raleigh
                NC
                US
    Effective date: 2007/05/24
    Expiration date: 2017/05/21
    Public key algorithm: rsaEncryption
    Public key size: 1024
    Signature algorithm: sha1WithRsaEncryption
    Issuer unique ID: None
    Subject unique ID: None
    Number of extensions: 3

```

Enter 1 to display extensions, 0 to return to menu: 0

Key and Certificate Menu

0 - Exit program

Enter option number (press ENTER to return to previous menu): 0

Das folgende Beispiel für `ssl.properties` zeigt, dass die Anweisungen `daemon_*` sich von dem zuvor aufgeführten Beispiel für eine SAF-Schlüsseldatei unterscheiden.

```

$ oedit /etc/rdz/ssl/ssl.properties
-> ändern: enable_ssl=true
-> Kommentarzeichen entfernen und ändern: daemon_keydb_file=rdzssl.kdb
-> Kommentarzeichen entfernen und ändern: daemon_keydb_password=rsessl
-> Kommentarzeichen entfernen und ändern: daemon_key_label=rdzrse
-> Kommentarzeichen entfernen und ändern: server_keystore_file=rdzssl.racf
-> Kommentarzeichen entfernen und ändern: server_keystore_label=rdzrse
-> Kommentarzeichen entfernen und ändern: server_keystore_type=JCERACFKS

```

Die oben beschriebenen Änderungen aktivieren SSL und teilen dem RSE-Dämon mit, dass das Zertifikat in der Schlüsseldatei `rdzssl.kdb` unter der Bezeichnung `rdzrse` mit dem Kennwort `rsessl` gespeichert ist. Der RSE-Server verwendet weiterhin eine SAF-konforme Schlüsseldatei.

Keystore mit keytool erstellen (optional)

Führen Sie diesen Schritt nicht aus, wenn Sie eine SAF-kompatible Schlüsseldatei für den Keystore des RSE-Servers verwenden.

`keytool -genkey` generiert ein privates Schlüsselpaar und ein entsprechendes selbst signiertes Zertifikat, die als ein (mit einem Aliasnamen bezeichneter) Eintrag in einer (neuen) Keystoredatei gespeichert werden.

Anmerkung: Sie müssen Java in Ihre Suchverzeichnisse für Befehle aufnehmen. Für die Ausführung von `keytool` ist möglicherweise die folgende Anweisung notwendig (wobei `/usr/lpp/java/J5.0` hier für das Verzeichnis steht, in dem Java installiert ist): `PATH=$PATH:/usr/lpp/java/J5.0/bin`

Alle Informationen können als ein Parameter übergeben werden. Durch die Längenbeschränkung der Befehlszeile sind jedoch folgende Interaktionen erforderlich:

```
$ cd /etc/rdz/ssl
$ keytool -genkey -alias rdzrse -validity 3650 -keystore rdzssl.jks -storepass
rsessl -keypass rsessl
What is your first and last name?
[Unknown]: rdz rse ssl
What is the name of your organizational unit?
[Unknown]: rdz
What is the name of your organization?
[Unknown]: IBM
What is the name of your City or Locality?
[Unknown]: Raleigh
What is the name of your State or Province?
[Unknown]: NC
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=rdz rse ssl, OU=rdz, O=IBM, L=Raleigh, ST=NC, C=US correct? (type "yes"
or "no")
[no]: yes
$ ls -l rdzssl.*
-rw-r--r-- 1 IBMUSER SYS1          1224 May 24 14:17 rdzssl.jks
```

Das oben erstellte selbst signierte Zertifikat ist für ca. 10 Jahre gültig (ohne Berücksichtigung des zusätzlichen Tages in Schaltjahren). Es wird in `/etc/rdz/ssl/rdzssl.jks` mit dem Aliasnamen `rdzrse` gespeichert. Das Kennwort (`rsessl`) stimmt mit dem Keystore-Kennwort überein. Dies ist eine RSE-Anforderung.

Das Ergebnis können Sie wie folgt mit der Option `-list` überprüfen:

```
$ keytool -list -alias rdzrse -keystore rdzssl.jks -storepass rsessl -v
Alias name: rdzrse
Creation date: May 24, 2007
Entry type: keyEntry
Certificate chain length: 1
Certificate 1:
Owner: CN=rdz rse ssl, OU=rdz, O=IBM, L=Raleigh, ST=NC, C=US
Issuer: CN=rdz rse ssl, OU=rdz, O=IBM, L=Raleigh, ST=NC, C=US
Serial number: 46562b2b
Valid from: 5/24/07 2:17 PM until: 5/21/17 2:17 PM
Certificate fingerprints:
    MD5: 9D:6D:F1:97:1E:AD:5D:B1:F7:14:16:4D:9B:1D:28:80
    SHA1: B5:E2:31:F5:B0:E8:9D:01:AD:2D:E6:82:4A:E0:B1:5E:12:CB:10:1C
```

Das folgende Beispiel für `ssl.properties` zeigt, dass die Anweisungen `server_*` sich von dem zuvor aufgeführten Beispiel für eine SAF-Schlüsseldatei unterscheiden.

```
$ oedit /etc/rdz/ssl/ssl.properties
-> ändern: enable_ssl=true
-> Kommentarzeichen entfernen und ändern: daemon_keydb_file=rdzssl.racf
-> Kommentarzeichen entfernen und ändern: daemon_key_label=rdzrse
-> Kommentarzeichen entfernen und ändern: server_keystore_file=rdzssl.jks
-> Kommentarzeichen entfernen und ändern: server_keystore_password=rsessl
-> Kommentarzeichen entfernen und ändern: server_keystore_label=rdzrse
-> Kommentarzeichen entfernen und ändern (optional): server_keystore_type=JKS
```

Die oben beschriebenen Änderungen aktivieren SSL und teilen dem RSE-Server mit, dass das Zertifikat in der Schlüsseldatei `rdzssl.jks` unter der Bezeichnung `rdzrse` mit dem Kennwort `rsessl` gespeichert ist. Der RSE-Dämon verwendet immer noch eine SAF-kompatible Schlüsseldatei.

Anhang B. TCP/IP konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von TCP/IP oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten.

Zusätzliche Informationen zur TCP/IP-Konfiguration finden Sie im *Communications Server: IP Configuration Guide* (IBM Form SC31-8775) und in der Veröffentlichung *Communications Server: IP Configuration Reference* (IBM Form SC31-8776).

Abhängigkeit vom Hostnamen

Wenn Sie APPC für TSO Commands Service verwenden, ist Developer for System z bei der Initialisierung darauf angewiesen, dass TCP/IP mit dem richtigen Hostnamen konfiguriert ist. Dies impliziert, dass die verschiedenen TCP/IP- und Resolverkonfigurationsdateien ordnungsgemäß definiert sein müssen.

Sie können Ihre TCP/IP-Konfiguration mit dem Installationsprüfprogramm fekfivpt testen. Der Befehl sollte eine Ausgabe wie im folgenden Beispiel zurückgeben (\$ ist die z/OS UNIX-Eingabeaufforderung):

```
$ fekfivpt
```

```
Wed Jul  2 13:11:54 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
```

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```
res_init Resolver values:
```

```
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset  = /etc/resolv.conf
Translation Table      = Default
UserId/JobName         = USERID
Caller API             = LE C Sockets
Caller Mode            = EBCDIC
(L) DataSetPrefix     = TCPIP
(L) HostName          = CDFMVS08
(L) TcpIpJobName       = TCPIP
(L) DomainOrigin      = RALEIGH.IBM.COM
(L) NameServer         = 9.42.206.2
                      9.42.206.3
(L) NsPortAddr        = 53           (L) ResolverTimeout      = 10
(L) ResolveVia        = UDP          (L) ResolverUdpRetries   = 1
(*) Options NDots     = 1
(*) SockNoTestStor    =
(*) AlwaysWto         = NO           (L) MessageCase         = MIXED
(*) LookUp            = DNS LOCAL
```

```
res_init Succeeded
```

```
res_init Started: 2008/07/02 13:11:54.755363
```

```
res_init Ended: 2008/07/02 13:11:54.755371
```

```
*****
```

```
MVS TCP/IP NETSTAT CS V1R9      TCPIP Name: TCPIP      13:11:54
```

```
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled
```

```
-----
```

```
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match
```

Wissenswertes zu Resolvern

Der Resolver arbeitet für Programme als ein Client, der für die Auflösung von Namen in Adressen oder von Adressen in Namen auf Namensserver zugreift. Für die Anforderung eines aufrufenden Programms kann der Resolver auf verfügbare Namensserver zugreifen, lokale Definitionen verwenden (z. B. `/etc/resolv.conf`, `/etc/hosts`, `/etc/ipnodes`, `HOSTS.SITEINFO`, `HOSTS.ADDRINFO` oder `ETC.IPNODES`) oder eine Kombination aus beiden Möglichkeiten anwenden.

Beim Starten des Adressraums des Resolvers wird eine optionale Resolverkonfigurationsdatei gelesen, auf die die DD-Karte `SETUP` in der JCL-Prozedur des Resolvers zeigt. Wenn die Konfigurationsdaten nicht zur Verfügung stehen, greift der Resolver auf die anwendbare native MVS- oder z/OS UNIX-Suchreihenfolge ohne Angaben von `GLOBALTCPIPDATA`, `DEFAULTTCPIPDATA`, `GLOBALIPNODES`, `DEFAULTIPNODES` oder `COMMONSEARCH` zurück.

Wissenswertes zur Suchreihenfolge für Konfigurationsdaten

Es ist wichtig, dass Sie die von TCP/IP-Funktionen verwendete Suchreihenfolge für Konfigurationsdateien verstehen und wissen, wann Sie die Standardsuchreihenfolge mit Umgebungsvariablen, JCL oder anderen von Ihnen angegebenen Variablen außer Kraft setzen können. Ausgehend von diesen Kenntnissen können Sie Ihre Benennungsstandards für lokale Dateien und HFS-Dateien anpassen. Außerdem ist es bei der Fehlerdiagnose hilfreich zu wissen, welche Konfigurationsdatei oder HFS-Datei verwendet wird.

Ein anderer wichtiger Punkt ist, dass die Suche bei Anwendung einer Suchreihenfolge für Konfigurationsdateien bei der ersten gefundenen Datei beendet wird. Wenn Sie Konfigurationsdaten in eine Datei stellen, die nie gefunden wird, weil es in der Suchreihenfolge vorher eine andere Datei gibt oder die Datei nicht von der Suchreihenfolge, die die Anwendung gewählt hat, erfasst wird, kann es daher zu unerwarteten Ergebnissen kommen.

Bei der Suche nach Konfigurationsdateien können Sie TCP/IP mit DD-Anweisungen in den JCL-Prozeduren oder durch das Setzen von Umgebungsvariablen explizit mitteilen, wo sich die meisten Konfigurationsdateien befinden. Sie können TCP/IP die Position der Konfigurationsdateien aber auch dynamisch auf der Grundlage der Suchreihenfolgen ermitteln lassen. Diese Suchreihenfolgen sind im *Communications Server: IP Configuration Guide* (IBM Form SC31-8775) dokumentiert.

Während der Initialisierung des TCP/IP-Stacks verwendet die Konfigurationskomponente des TCP/IP-Stacks `TCPIP.DATA`, um den `HOSTNAME` des Stacks zu ermitteln. Zum Abrufen des Wertes wird die Suchreihenfolge für die z/OS UNIX-Umgebung verwendet.

Anmerkung: Mit dem Trace-Resolver können Sie bestimmen, welche `TCPIP.DATA`-Werte der Resolver verwendet und woher sie stammen. Informationen zum dynamischen Starten des Trace enthält der *Communications*

Server: IP Diagnosis Guide (IBM Form GC31-8782). Setzen Sie nach Aktivierung des Trace einen TSO-Befehl **NETSTAT HOME** und einen z/OS UNIX-Shellbefehl **netstat -h** ab, um die Werte anzuzeigen. Wenn Sie von TSO und von der z/OS UNIX-Shell ein PING für einen Hostnamen absetzen, werden auch die Aktivitäten in Richtung aller DNS-Server angezeigt, die möglicherweise konfiguriert sind.

Suchreihenfolgen in der z/OS UNIX-Umgebung

Die Datei oder Tabelle, nach der gesucht wird, kann eine MVS-Datei oder eine HFS-Datei sein. Dies hängt von den Einstellungen in der Resolverkonfiguration und dem Vorhandensein bestimmter Dateien im System ab.

Basiskonfigurationsdateien des Resolvers

Die Basiskonfigurationsdatei des Resolvers enthält TCPIP.DATA-Anweisungen. Diese Datei wird wegen der enthaltenen Resolveranweisungen referenziert, aber auch, weil sie unter anderem das Dateipräfix (Wert der Anweisung DATASETPREFIX) für den Zugriff auf die in diesem Abschnitt genannten Konfigurationsdateien enthält.

Für den Zugriff auf die Basiskonfigurationsdatei des Resolvers wird diese Suchreihenfolge verwendet:

1. **GLOBALTCPIPDATA**

Wenn der Wert der Konfigurationsanweisung GLOBALTCPIPDATA für den Resolver definiert ist, wird er verwendet. (Lesen Sie hierzu auch den Abschnitt „Wissenswertes zu Resolvern“ auf Seite 314.) Es wird weiter nach einer zusätzlichen Konfigurationsdatei gesucht. Die Suche endet mit der nächsten gefundenen Datei.

2. Wert der Umgebungsvariablen **RESOLVER_CONFIG**

Der Wert der Umgebungsvariablen wird verwendet. Die Suche scheitert, wenn die Datei nicht vorhanden ist oder anderweitig exklusiv zugeordnet ist.

3. **/etc/resolv.conf**

4. Karte **//SYSTCPD DD**

Die dem DD-Namen in SYSTCPD zugeordnete Datei wird verwendet. In der z/OS UNIX-Umgebung hat ein untergeordneter Prozess keinen Zugriff auf die DD-Anweisung SYSTCPD. Dies ist darauf zurückzuführen, dass bei fork()- oder exec-Funktionsaufrufen die SYSTCPD-Zuordnung nicht vom übergeordneten Prozess übernommen wird.

5. **USERID.TCPIP.DATA**

USERID ist die Benutzer-ID, die der aktuellen Sicherheitsumgebung (Adressraum, Task oder Thread) zugeordnet ist.

6. **JOBNAME.TCPIP.DATA**

JOBNAME ist der in der JCL-Anweisung JOB angegebene Name für Batch-Jobs oder der Prozedurname für eine gestartete Prozedur.

7. **SYS1.TCPPARMS(TCPDATA)**

8. **DEFAULTTCPIPDATA**

Wenn der Wert der Konfigurationsanweisung DEFAULTTCPIPDATA für den Resolver definiert ist, wird er verwendet. (Lesen Sie hierzu auch den Abschnitt „Wissenswertes zu Resolvern“ auf Seite 314.)

9. **TCPIP.TCPIP.DATA**

Umsetztabelle

Die Umsetztabelle (EBCDIC zu ASCII und ASCII zu EBCDIC) werden referenziert, um die zu verwendenden Umsetzungsdateien zu ermitteln. Für den Zugriff auf diese Konfigurationsdatei wird die folgende Suchreihenfolge verwendet: (Die Suche endet mit der ersten gefundenen Datei.)

1. Der Wert der Umgebungsvariablen **X_XLATE**
Dies ist der Name der mit dem TSO-Befehl CONVXLAT erzeugten Umsetztabelle.
2. **USERID.STANDARD.TCPXLBIN**
USERID ist die Benutzer-ID, die der aktuellen Sicherheitsumgebung (Adressraum oder Task/Thread) zugeordnet ist.
3. **JOBNAME.STANDARD.TCPXLBIN**
JOBNAME ist der in der JCL-Anweisung JOB angegebene Name für Batch-Jobs oder der Prozedurname für eine gestartete Prozedur.
4. **HLQ.STANDARD.TCPXLBIN**
HLQ repräsentiert den Wert der Anweisung DATASETPREFIX in der Basiskonfigurationsdatei des Resolvers, der verwendet wird, wenn er aufgefunden wird. Andernfalls wird der Standard-HLQ TCPIP verwendet.
5. Wenn keine Tabelle gefunden wird, verwendet der Resolver eine fest codierte Standardtabelle, die mit der im Dateimember SEZATCPX(STANDARD) aufgelisteten Tabelle identisch ist.

Lokale Hosttabellen

Standardmäßig versucht der Resolver zuerst, konfigurierte Domänennamensserver für Auflösungsanforderungen zu verwenden. Falls die Auflösungsanforderung nicht erfüllt werden kann, werden lokale Hosttabellen genutzt. Das Verhalten des Resolvers wird von TCPIP.DATA-Anweisungen gesteuert.

Die TCPIP.DATA-Resolveranweisungen definieren, ob und ggf. wie Domänennamensserver zu verwenden sind. Außerdem kann mit der Anweisung LOOKUP TCPIP.DATA gesteuert werden, wie Domänennamensserver und lokale Hosttabellen verwendet werden sollen. Weitere Informationen zu TCPIP.DATA-Anweisungen finden Sie in der Veröffentlichung *Communications Server: IP Configuration Reference* (IBM Form SC31-8776).

Der Resolver verwendet die spezifische Suchreihenfolge für Sitenamen von IPv4 uneingeschränkt für getnetbyname-API-Aufrufe. Die spezifische Suchreihenfolge für Sitenamen von IPv4 ist wie folgt. Die Suche endet mit der ersten gefundenen Datei:

1. Der Wert der Umgebungsvariablen **X_SITE**
Dies ist der Name der mit dem TSO-Befehl **MAKESITE** erstellten Informationsdatei HOSTS.SITEINFO.
2. Wert der Umgebungsvariablen **X_ADDR**
Dies ist der Name der mit dem TSO-Befehl **MAKESITE** erstellten Informationsdatei HOSTS.ADDRINFO.
3. **/etc/hosts**
4. **USERID.HOSTS.SITEINFO**
USERID ist die Benutzer-ID, die der aktuellen Sicherheitsumgebung (Adressraum oder Task/Thread) zugeordnet ist.

5. JOBNAME.HOSTS.SITEINFO

JOBNAME ist der in der JCL-Anweisung JOB angegebene Name für Batch-Jobs oder der Prozedurname für eine gestartete Prozedur.

6. HLQ.HOSTS.SITEINFO

HLQ repräsentiert den Wert der Anweisung DATASETPREFIX in der Basiskonfigurationsdatei des Resolvers, der verwendet wird, wenn er aufgefunden wird. Andernfalls wird der Standard-HLQ TCPIP verwendet.

Diese Konfigurationsinformationen in Developer for System z anwenden

Wie bereits erwähnt, ist Developer for System z bei der Initialisierung davon abhängig, dass TCP/IP mit dem richtigen Hostnamen konfiguriert ist, wenn Sie APPC verwenden. Dies impliziert, dass die verschiedenen TCP/IP- und Resolver-konfigurationsdateien ordnungsgemäß definiert sein müssen.

Im folgenden Beispiel geht es hauptsächlich um einige Konfigurationstasks für TCP/IP und den Resolver. Beachten Sie, dass es sich nicht um eine komplette Konfiguration für TCP/IP oder den Resolver handelt. Das Beispiel hebt nur einige wichtige Aspekte hervor, die auf Ihren Standort anwendbar sein könnten.

1. In der folgenden JCL sehen Sie, dass TCP/IP den Stack-Hostnamen mithilfe von SYS1.TCPPARMS(TCPDATA) bestimmt.

```
//TCPIP    PROC  PARMS='CTRACE(CTIEZB00)',PROF=TCPPROF,DATA=TCPDATA
//*
//* TCP/IP NETWORK
//*
//TCPIP    EXEC  PGM=EZBTCPIP,REGION=0M,TIME=1440,PARM=&PARMS
//PROFILE  DD   DISP=SHR,DSN=SYS1.TCPPARMS(&PROF)
//SYSTCPD  DD   DISP=SHR,DSN=SYS1.TCPPARMS(&DATA)
//SYSPRINT DD   SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//ALGPRINT DD   SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CFGPRINT DD   SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSOUT   DD   SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CEEDUMP  DD   SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSERROR DD   SYSOUT=*
```

2. Aus SYS1.TCPPARMS(TCPDATA) können wir entnehmen, dass der Systemname als Hostname verwendet werden soll und dass kein Domänen Namensserver (DNS) verwendet wird. Alle Namen werden durch eine Suche in der Standorttabelle aufgelöst.

```
; HOSTNAME gibt den TCP-Hostnamen dieses Systems an. Wenn kein
; Wert angegeben ist, wird für HOSTNAME standardmäßig der im PARMLIB-Member
; IEFSSNxx angegebene Knotenname verwendet.
;
; HOSTNAME
;
; DOMAINORIGIN gibt den Domänenursprung an, der an Hostnamen angehängt wird,
; die an den Resolver übergeben werden. Enthält ein Hostname
; Punkte, wird der Wert von DOMAINORIGIN nicht
; an den Hostnamen angehängt.
;
DOMAINORIGIN  RALEIGH.IBM.COM
;
; NSINTERADDR gibt die IP-Adresse des Namensservers an.
; LOOPBACK (14.0.0.0) gibt Ihren lokalen Namensserver an. Wenn kein
; Namensserver verwendet wird, codieren Sie keine Anweisung NSINTERADDR.
; (Setzen Sie die folgende Zeile NSINTERADDR auf Kommentar, wenn alle
; Namen durch eine Suche in der Standorttabelle aufgelöst werden sollen.)
;
; NSINTERADDR  14.0.0.0
```



```

;
; TRACE RESOLVER bewirkt, dass ein vollständiger Trace für alle Abfragen an den
; Namensserver oder an Standorttabellen und alle entsprechenden Antworten auf die
; Benutzerkonsole geschrieben werden. Der Befehl ist nur für Debugzwecke bestimmt.
;
; TRACE RESOLVER

```

3. In der Resolver-JCL sehen wir, dass die DD-Anweisung SETUP nicht verwendet wird. Wie Sie aus dem Abschnitt „Wissenswertes zu Resolvieren“ auf Seite 314 wissen, bedeutet dies, dass GLOBALTCIPDATA und andere Variablen nicht verwendet werden.

```

//RESOLVER PROC PARMS='CTRACE(CTIRES00)'
//*
/** RESOLVER FÜR IP-NAMEN – BEGINN MIT SUB=MSTR
/**
//RESOLVER EXEC PGM=EZBREINI,REGION=0M,TIME=1440,PARM=&PARMS
/**SETUP DD DISP=SHR,DSN=USER.PROCLIB(RESSETUP),FREE=CLOSE

```

4. Wenn wir davon ausgehen, dass die Umgebungsvariable RESOLVER_CONFIG nicht gesetzt ist, können wir aus Tabelle 51 auf Seite 319 entnehmen, dass der Resolver versuchen wird, /etc/resolv.conf als Basiskonfigurationsdatei zu verwenden.

```

TCPIPJOBNAME TCPIP
DomainOrigin RALEIGH.IBM.COM
HostName CDFMVS08

```

Wie im Abschnitt „Suchreihenfolgen in der z/OS UNIX-Umgebung“ auf Seite 315 erwähnt, enthält die Basiskonfigurationsdatei TCPIP.DATA-Anweisungen. Wenn der Systemname CDFMVS08 lautet, sehen wir, dass /etc/resolv.conf mit SYS1.TCPPARMS(TCPDATA) synchron ist. (TCPDATA gibt an, dass der Systemname als Hostname verwendet werden soll.) Es liegen keine DNS-Definitionen vor, sodass die Standorttabellen durchsucht werden.

5. Aus Tabelle 51 auf Seite 319 können wir außerdem entnehmen, dass in Ermangelung anderer Angaben standardmäßig die ASCII-EBCDIC-Umsetzungstabelle verwendet wird.
6. Unter der Voraussetzung, dass der TSO-Befehl **MAKESITE** nicht verwendet wird (um die Variablen X_SITE und X_ADDR zu erstellen), wird /etc/hosts als Standorttabelle für die Namenssuche verwendet.

```

# Resolver /etc/hosts-Datei cdfmvs08
9.42.112.75 cdfmvs08 # CDFMVS08 Host
9.42.112.75 cdfmvs08.raleigh.ibm.com # CDFMVS08 Host
127.0.0.1 localhost

```

Der minimale Inhalt dieser Datei bezieht sich auf das aktuelle System. Im obigen Beispiel sind cdfmvs08 und cdfmvs08.raleigh.ibm.com als gültige Namen für die IP-Adresse des z/OS-Systems definiert.

Wenn ein Domänennamensserver (DNS) verwendet werden würde, würde der DNS die /etc/hosts-Informationen enthalten und /etc/resolv.conf und SYS1.TCPPARMS(TCPDATA) würden Anweisungen enthalten, die den DNS für das System identifizieren.

Um Unklarheiten zu vermeiden, sollten die Konfigurationsdateien für TCP/IP und den Resolver synchron sein.

Tabelle 51. Für den Resolver verfügbare lokale Definitionen

Beschreibung des Dateityps	Betroffene APIs	Mögliche Dateien
Basiskonfigurationsdateien des Resolvers	Alle APIs	<ol style="list-style-type: none"> 1. GLOBALTCPIPDATA 2. Umgebungsvariable RESOLVER_CONFIG 3. /etc/resolv.conf 4. DD-Name in SYSTCPD 5. USERID.TCPIP.DATA 6. JOBNAME.TCPIP.DATA 7. SYS1.TCPPARMS(TCPDATA) 8. DEFAULTTCPIPDATA 9. TCPIP.TCPIP.DATA
Umsetztabelle	Alle APIs	<ol style="list-style-type: none"> 1. Umgebungsvariable X_XLATE 2. USERID.STANDARD.TCPXLBIN 3. JOBNAME.STANDARD.TCPXLBIN 4. HLQ.STANDARD.TCPXLBIN 5. Vom Resolver bereitgestellte Umsetztabelle (Member STANDARD in SEZATCPX)
Lokale Hosttabellen	endhostent endnetent getaddrinfo gethostbyaddr gethostbyname gethostent GetHostNumber GetHostResol GetHostString getnameinfo getnetbyaddr getnetbyname getnetent IsLocalHost Resolve sethostent setnetent	IPV4 <ol style="list-style-type: none"> 1. Umgebungsvariable X_SITE 2. Umgebungsvariable X_ADDR 3. /etc/hosts 4. USERID.HOSTS.xxxxINFO 5. JOBNAME.HOSTS.xxxxINFO 6. HLQ.HOSTS.xxxxINFO IPV6 <ol style="list-style-type: none"> 1. GLOBALIPNODES 2. Umgebungsvariable RESOLVER_IPNODES 3. USERID.ETC.IPNODES 4. JOBNAME.ETC.IPNODES 5. HLQ.ETC.IPNODES 6. DEFAULTIPNODES 7. /etc/ipnodes

Anmerkung: Tabelle 51 ist ein Auszug aus einer Tabelle im *Communications Server: IP Configuration Guide* (IBM Form SC31-8775). Die vollständige Tabelle können Sie sich im genannten Handbuch ansehen.

Nicht ordnungsgemäß aufgelöste Hostadresse

Wenn Sie feststellen, dass der TCP/IP-Resolver die Hostadresse nicht ordnungsgemäß auflösen kann, liegt dies höchstwahrscheinlich daran, dass eine Resolverkonfigurationsdatei fehlt oder unvollständig ist. Ein deutlicher Hinweis auf dieses Problem ist die folgende Nachricht in `lock.log`:

```
clientip(0.0.0.0) <> callerip(<Host-IP-Adresse>)
```

Führen Sie zur Überprüfung das TCP/IP-Installationsprüfprogramm `fekfivpt` wie in Kapitel 7, „Installationsprüfung“, auf Seite 109 beschrieben aus. Der Abschnitt der Ausgabe mit der Resolverkonfiguration sieht in etwa wie das folgende Beispiel aus:

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```
res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table     = Default
UserId/JobName        = USERID
Caller API            = LE C Sockets
Caller Mode           = EBCDIC
```

Vergewissern Sie sich, dass die Definitionen in der von „Local Tcp/Ip Dataset“ referenzierten Datei stimmen.

Wenn Sie für die IP-Resolver-Datei keinen Standardnamen verwenden, bleibt dieses Feld leer (bei Verwendung der z/OS UNIX-Suchreihenfolge). Fügen Sie in dem Fall die folgende Anweisung zu `rsed.envvars` hinzu. `<Resolver-Datei>` repräsentiert hier den Namen Ihrer IP-Resolver-Datei.

```
RESOLVER_CONFIG='<Resolver-Datei>'
```

Anhang C. INETD konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von INETD oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten. INETD wird von Developer for System z verwendet, um REXEC/SSH-Funktionalitäten bereitzustellen.

Der Dämon INETD führt das Service-Management für ein IP-Netz durch. Er verringert die Systembelastung, indem er andere Dämonen nur bei Bedarf aufruft und intern mehrere einfache Internetservices (wie echo) bereitstellt. INETD liest die Konfigurationsdatei `inetd.conf`, um zu bestimmen, welche zusätzlichen Services bereitgestellt werden müssen. ETC.SERVICES wird für die Verknüpfung von Services mit Ports verwendet.

`inetd.conf`

Die Services, die auf INETD zurückgreifen, sind in `inetd.conf` definiert. Diese Datei wird während der Startzeit von INETD gelesen. Die Standardposition und der Standardname für `inetd.conf` lauten `/etc/inetd.conf`. Ein Beispiel für eine Datei `inetd.conf` finden Sie unter `/samples/inetd.conf`.

Für Einträge in `inetd.conf` gelten die folgenden Syntaxregeln:

- Kommentare beginnen mit einem Nummernzeichen (#) oder Semikolon (;) und gehen bis zum Ende der Zeile.
- Bei Einträgen wird die Groß-/Kleinschreibung unterschieden.
- Die Einträge sind feldabhängig, jedoch nicht spaltenabhängig.
- Die Felder sind durch ein Leerzeichen oder Tabulatorzeichen voneinander getrennt.
- Einträge können mehrere Zeilen umfassen, wobei die folgenden zusätzlichen Syntaxregeln zu beachten sind:
 - Die Teilung muss zwischen zwei separaten Wörtern erfolgen (die durch ein Leerzeichen oder Tabulatorzeichen voneinander getrennt sind).
 - Die Folgezeile muss mit einem Leerzeichen oder Tabulatorzeichen beginnen.
 - In die Fortsetzung dürfen keine Kommentare eingebettet werden.

Jeder Eintrag umfasst sieben positionsgebundene Felder im folgenden Format:

Service**name** Socket**typ** Protokoll Option_wait Benutzer-ID Serverprogramm
Serverprogrammargumente

[IP-Adresse:]Service**name**

IP-Adresse ist eine lokale IP-Adresse, gefolgt von einem Doppelpunkt. Wenn die Adresse angegeben ist, wird sie anstelle von `INADDR_ANY` oder der aktuellen Standardadresse verwendet. Wenn Sie speziell `INADDR_ANY` anfordern möchten, verwenden Sie `.*.` Wenn IP-Adresse (oder ein Doppelpunkt) ohne weitere Einträge in der Zeile angegeben ist, wird diese Adresse zum Standard für die folgenden Zeilen, bis eine neue Standardadresse angegeben ist. Service**name** ist ein anerkannter oder benutzerdefinierter Service**name**. Der angegebene Name muss mit einem der in ETC.SERVICES definierten Servernamen übereinstimmen.

Sockettyp

Der Typ stream oder dgram gibt an, ob für den Service ein Datenstrom- oder Datagrammsocket verwendet wird.

Protokoll[,sndbuf=n[,rcvbuf=n]]

Für Protokoll kann der Wert tcp[4|6] oder udp[4|6] als weitere Qualifizierung des Servicenamens angegeben werden. Der Servicename und das Protokoll müssen mit einem Eintrag in ETC.SERVICES übereinstimmen, bis auf die "4" oder "6", die im Eintrag in ETC.SERVICES nicht enthalten sein sollte.

Die Werte sndbuf und rcvbuf geben die Größe des Sendepuffers und des Empfangspuffers an. Die von 'n' repräsentierte Größe kann in Bytes angegeben werden. Sie können aber auch ein 'k' oder 'm' hinzufügen, wenn Sie Kilobytes oder Megabytes angeben möchten. Die Reihenfolge, in der Sie sndbuf und rcvbuf angeben, ist beliebig.

Option_wait[.max]

Mit der Option wait oder nowait, wait wird angegeben, dass der Dämon ein Einzelthreaddämon ist und die nächste Anforderung erst bedienen kann, wenn die erste abgeschlossen ist. Wenn nowait angegeben ist, setzt INETD bei Empfang einer Verbindungsanforderung an einem Datenstromsocket ein 'accept' ab. Wenn 'wait' angegeben ist, muss der Server das 'accept' absetzen, sofern es sich um einen Datenstromsocket handelt.

Der Wert max gibt die maximal zulässige Anzahl von Benutzern an, die innerhalb eines Intervalls von 60 Sekunden einen Service anfordern dürfen. Der Standardwert liegt bei 40. Wird der Maximalwert überschritten, wird der Service-Port geschlossen.

Benutzer-ID[.Gruppe]

Benutzer-ID ist die Benutzer-ID, unter der der verzweigte Dämon ausgeführt werden muss. Diese Benutzer-ID kann von der INETD-Benutzer-ID abweichen. Welche Berechtigungen dieser Benutzer-ID erteilt werden, hängt von den Anforderungen des Service ab. Die INETD-Benutzer-ID benötigt die Berechtigung BPX.DAEMON, um den verzweigten Prozess auf diese Benutzer-ID umzuschalten.

Der optionale Wert Gruppe ist durch einen Punkt (.) von der Benutzer-ID getrennt und ermöglicht die Ausführung des Servers mit einer Gruppen-ID, die von der Standardgruppen-ID für diese Benutzer-ID abweicht.

Serverprogramm

Serverprogramm ist der vollständige Pfadname des Service. Beispiel:
/usr/sbin/rlogind ist der vollständige Pfadname für den Befehl rlogind.

Serverprogrammargumente

Es können maximal 20 Argumente angegeben werden. Das erste Argument ist der Servername.

ETC.SERVICES

INETD verwendet ETC.SERVICES, um den Services, die von INETD unterstützt werden müssen, Portnummern und Protokolle zuzuordnen. Diese Datei kann eine MVS-Datei oder eine z/OS UNIX-Datei sein. Ein Beispiel ist in SEZAINST(SERVICES) enthalten, das auch unter /usr/lpp/tcpip/samples/services verfügbar ist. Die Suchreihenfolge für ETC.SERVICES hängt davon ab, ob INETD mit einer z/OS UNIX-Methode oder einer nativen MVS-Methode gestartet wird.

Für die Spezifikation der Serviceinformationen gelten die folgenden Syntaxregeln:

- Wenn ETC.SERVICES eine MVS-Datei ist, muss es sich um eine Datei im Fixed-Format oder Fixed-Block-Format mit LRECL zwischen 56 und 256 handeln.
- Wenn ETC.SERVICES eine HFS-Datei ist, kann sie eine maximale Länge von 256 haben.
- Einträge in einer Zeile sind durch Leerzeichen oder Tabulatorzeichen voneinander getrennt.
- Jeder Service ist in einer gesonderten Zeile angegeben.
- Jeder Servicename muss an der ersten Position einer Zeile beginnen.
- Die maximale Länge für Servicenamen und Aliasnamen liegt bei 32 Zeichen.
- Es werden maximal 35 Aliasnamen erkannt.
- Bei Servicenamen und Aliasnamen wird die Groß-/Kleinschreibung unterschieden.
- Kommentare beginnen mit einem Nummernzeichen (#) oder Semikolon (;) und gehen bis zum Ende der Zeile.

Jeder Eintrag umfasst vier positionsgebundene Felder im folgenden Format:

ServiceName Portnummer/Protokoll Aliasnamen

ServiceName

Gibt einen anerkannten oder benutzerdefinierten Servicenamen an.

Portnummer

Gibt die für den Service verwendete Socket-Port-Nummer an.

Protokoll

Gibt das für den Service verwendete Transportprotokoll an. Gültige Werte sind tcp und udp.

Aliasnamen

Gibt eine Liste nicht offizieller Servicenamen an.

Suchreihenfolge in der z/OS UNIX-Umgebung

Für den Zugriff auf ETC.SERVICES unter z/OS UNIX wird diese Suchreihenfolge verwendet. Die Suche endet mit der ersten gefundenen Datei:

1. **/etc/services**
2. **USERID.ETC.SERVICES**
USERID ist die Benutzer-ID, die zum Starten von INETD verwendet wird..
3. **HLQ.ETC.SERVICES**
HLQ repräsentiert den Wert der Anweisung DATASETPREFIX in der Basiskonfigurationsdatei des Resolvers, der verwendet wird, wenn er aufgefunden wird. Andernfalls wird der Standard-HLQ TCP/IP verwendet.

Suchreihenfolge in der nativen MVS-Umgebung

Für den Zugriff auf ETC.SERVICES in der nativen MVS-Umgebung wird diese Suchreihenfolge verwendet. Die Suche endet mit der ersten gefundenen Datei:

1. **//DD-Karte SERVICES**
Die der DD-Anweisung SERVICES zugeordnete Datei wird verwendet.
2. **USERID.ETC.SERVICES**
USERID ist die Benutzer-ID, die zum Starten von INETD verwendet wird..
3. **JOBNAME.ETC.SERVICES**

JOBNAME ist der in der JCL-Anweisung JOB angegebene Name für Batch-Jobs oder der Prozedurname für eine gestartete Prozedur.

4. HLQ.ETC.SERVICES

HLQ repräsentiert den Wert der Anweisung DATASETPREFIX in der Basiskonfigurationsdatei des Resolvers, der verwendet wird, wenn er aufgefunden wird. Andernfalls wird der Standard-HLQ TCPIP verwendet.

Anmerkung: Wenn INETD mit BPXPATCH gestartet wird, wird nicht die native MVS-Suchreihenfolge verwendet, weil BPXBATCH den Startbefehl in der z/OS UNIX-Umgebung ausführt. Die native MVS-Suchreihenfolge wird nur verwendet, wenn ein MVS-Lademodul wie SEZALOAD(FTP) gestartet wird.

Portdefinitionen in PROFILE.TCPIP

Verwechseln Sie nicht die PORT-Definitionen (oder PORTRANGE-Definitionen) in PROFILE.TCPIP mit Ports, die in ETC.SERVICES definiert sind. Diese Ports werden jeweils für verschiedene Zwecke verwendet. Anhand der in PROFILE.TCPIP definierten Ports stellt TCPIP fest, ob ein Port für einen bestimmten Service reserviert ist. ETC.SERVICES wird von INETD verwendet, um einem Service einen Port zuzuordnen.

Wenn INETD eine Anforderung an einem überwachten Port empfängt, richtet er eine untergeordnete Prozessverzweigung (mit dem angeforderten Service) ein, die die Bezeichnung inetdx hat. Hier steht inetd für den Jobnamen für INETD (der von der Startmethode abhängig ist) und x für eine einstellige Zahl.

Dies kompliziert die Portreservierung. Wenn ein überwachter INETD-Port in PROFILE.TCPIP reserviert ist, sollte der Name der gestarteten JCL-Prozedur für den z/OS UNIX-Kernel-Adressraum verwendet werden, damit fast jeder Prozess an den Port gebunden werden kann. Dieser Name ist normalerweise OMVS, sofern im Parameter STARTUP_PROC des PARMLIB-Members BPXPRMxx nicht explizit ein anderer Name angegeben ist.

In der nachfolgenden Auflistung ist erläutert, wie der Jobname ausgehend von der Umgebung, in der die Anwendung ausgeführt wird, bestimmt werden kann:

- Im Batch ausgeführte Anwendungen verwenden den Batch-Job-Namen.
- Von der MVS-Bedienerkonsole aus gestartete Anwendungen verwenden den Namen der gestarteten Prozedur (STC) als Jobnamen.
- Mit einer TSO-Benutzer-ID ausgeführte Anwendungen verwenden die TSO-Benutzer-ID als Jobnamen.
- Anwendungen, die von der z/OS-Shell ausgeführt werden, haben normalerweise einen Jobnamen, der sich aus der Benutzer-ID des angemeldeten Benutzers und einem Suffix (ein Zeichen) zusammensetzt.
- Berechtigte Benutzer können Anwendungen von der z/OS-Shell ausführen und den Jobnamen mit der Umgebungsvariablen _BPX_JOBNAME setzen. In diesem Fall ist der für die Umgebungsvariable angegebene Wert der Jobname.
- Der Name der gestarteten JCL-Prozedur für den UNIX System Services-Kernel-Adressraum kann verwendet werden, wenn die Anbindung an den Port für fast alle Aufrufer der Socket-API bind() (mit Ausnahme von Benutzern der Pascal-API) möglich sein soll. Dieser Name ist normalerweise OMVS, sofern im Parameter STARTUP_PROC des PARMLIB-Members BPXPRMxx nicht explizit ein anderer Name angegeben ist.

- Von INETD gestartete z/OS UNIX-Anwendungen verwenden den Jobnamen des INETD-Servers.

Anmerkung: Die in ETC.SERVICES definierten Ports können sich von der in PROFILE.TCPIP für den Service reservierten Portnummer unterscheiden, obwohl dies nicht empfohlen wird.

/etc/inetd.pid

Der INETD-Prozess erstellt eine temporäre Datei /etc/inetd.pid, die die Prozess-ID (PID) des derzeit ausgeführten Dämons INETD enthält. Mithilfe dieses PID-Wertes werden syslog-Einträge identifiziert, die von dem INETD-Prozess stammen. Dieser PID-Wert wird außerdem für Befehle bereitgestellt, die einen solchen Wert benötigen, z. B. kill. Darüber hinaus wird die PID als Sperrmechanismus verwendet, um zu verhindern, dass mehr als ein INETD-Prozess aktiv ist.

Start

Die z/OS UNIX-Implementierung von INETD befindet sich standardmäßig in /usr/sbin/inetd und unterstützt zwei optionale, nicht positionsgebundene Startparameter:

```
/usr/sbin/inetd [-d] [inetd.conf]
```

-d Debugoption. Die Debugausgabe wird an stderr gesendet und kann dann vom Dämon syslogd an eine Datei weitergeleitet werden. Weitere Informationen zu syslogd finden Sie im *Communications Server: IP Configuration Guide* (IBM Form SC31-8775). Wenn INETD auf diese Weise gestartet wurde, wird keine Verzweigung für einen untergeordneten Prozess zum Starten eines Service erstellt.

inetd.conf

Konfigurationsdatei. Der Standardwert ist /etc/inetd.conf

Sie sollten INETD während des IPL starten. Am häufigsten geschieht dies von /etc/rc oder /etc/inittab aus (nur unter z/OS ab Version 1.8). Sie können den Dämon auch von einem Job oder einer gestarteten Task aus starten, indem Sie BPXBATCH verwenden, oder in einer Shellsitzung eines Benutzers mit entsprechender Berechtigung.

/etc/rc

Wenn INETD unter z/OS UNIX vom Initialisierungs-Shell-Skript /etc/rc gestartet wird, sucht der Dämon in der z/OS UNIX-Suchreihenfolge nach ETC.SERVICES. Eine Beispieldatei /etc/rc ist im Lieferumfang als /samples/rc enthalten. Zum Starten von INETD können die folgenden Beispielbefehle verwendet werden.

```
# Start INETD
_BPX_JOBNAME='INETD' /usr/sbin/inetd /etc/inetd.conf &
sleep 5
```

/etc/inittab

Unter z/OS ab Version 1.8 gibt es eine alternative Methode (/etc/inittab), um während der Initialisierung von z/OS UNIX Befehle abzusetzen. Bei Verwendung von /etc/inittab haben Sie die Möglichkeit, den Parameter respawn zu definieren, der den Prozess automatisch neu startet, wenn er beendet ist. (Für einen zweiten Neustart innerhalb von 15 Minuten wird ein WTOR an den Bediener gesendet.) Wenn INETD unter Verwendung von /etc/inittab gestartet wird, sucht der Dämon in der z/OS UNIX-Suchreihenfolge nach ETC.SERVICES. Eine Beispieldatei

/etc/inittab ist im Lieferumfang als /samples/inittab enthalten. Zum Starten von INETD kann der folgende Beispielbefehl verwendet werden.

```
# Start INETD
inetd::respfrk:/usr/sbin/inetd /etc/inetd.conf
```

Anmerkung: Beachten Sie, dass der im Beispiel verwendete Parameter respfrk das Attribut respawn an alle verzweigten Prozesse, einschließlich RSE, sendet. Wenn der Client die Verbindung schließt, wird sie von respawn neu gestartet. Der RSE-Server wird später wieder durch eine Zeitlimitüberschreitung beendet. Weitere Informationen zu inittab enthält die Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

BPXBATCH

Die Startmethode BPXBATCH funktioniert für gestartete Tasks und für Benutzerjobs. Vergessen Sie nicht, dass INETD ein Hintergrundprozess ist, sodass der BPXBATCH-Schritt, der INETD startet, innerhalb weniger Sekunden nach dem Start beendet ist. Wenn INETD von BPXBATCH gestartet wird, sucht der Dämon in der z/OS UNIX-Suchreihenfolge nach ETC.SERVICES. Die im folgenden Codebeispiel aufgelistete JCL ist eine Beispielprozedur zum Starten von INETD. (Der Schritt KILL entfernt einen ggf. vorhandenen aktiven INETD-Prozess.)

```
//INETD    PROC PRM=
//*
//KILL      EXEC PGM=BPXBATCH,REGION=0M,
//          PARM='SH ps -e | grep inetd | cut -c 1-10 | xargs -n 1 kill'
//*
//INETD     EXEC PGM=BPXBATCH,REGION=0M,
//          PARM='PGM /usr/sbin/inetd &PRM'
//STDERR    DD SYSOUT=*
//* STDIN, STDOUT und STDENV nehmen standardmäßig den Wert /dev/null an.
//*
```

Abbildung 64. Start-JCL für INETD

Anmerkung:

- STDIN, STDOUT und STDERR müssen beim Anlegen z/OS UNIX-Dateien sein. STDENV kann eine MVS-Datei oder eine z/OS UNIX-Datei sein. Ab z/OS 1.7 kann SYSOUT für STDOUT und STDERR zugeordnet werden. In der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802) erfahren Sie mehr über BPXBATCH.
- Wenn INETD von BPXBATCH gestartet wird, kann inetd.conf eine MVS-Datei oder ein MVS-Dateimember sein. Codieren Sie hierfür die Anweisung PARM wie im folgenden Beispiel. (Verwenden Sie nur einfache Hochkommata (').)

```
// PARM='PGM /usr/sbin/inetd //'SYS1.TCPPARMS(INETCONF)'' &PRM'
```

Shellsitzung

Wenn INETD von einer Shellsitzung aus gestartet wird, verwendet der Dämon die z/OS UNIX-Suche nach ETC.SERVICES. Die folgenden Beispielbefehle können (von einer Person mit ausreichender Berechtigung) zum Stoppen und Starten von INETD verwendet werden. (# ist die z/OS UNIX-Eingabeaufforderung.)

```
# ps -e | grep inetd
7 ?          0:00 /usr/sbin/inetd
# kill 7
# _BPX_JOBNAME='INETD' /usr/sbin/inetd &
```

Anmerkung: Diese Methode wird nicht für den Erststart empfohlen. Dafür ist /etc/rc oder /etc/inittab besser geeignet, da die Scripts während der Initialisierung von z/OS UNIX ausgeführt werden.

Sicherheit

INETD ist ein z/OS UNIX-Prozess und erfordert daher, dass die Sicherheitssoftware gültige OMVS-Definitionen für die INETD zugeordnete Benutzer-ID enthält. Für die Benutzer-ID müssen UID, HOME und PROGRAM gesetzt sein. Außerdem muss GID für die Standardgruppe des Benutzers gesetzt sein. Wenn INETD von /etc/rc oder /etc/inittab gestartet wird, wird die Benutzer-ID vom z/OS UNIX-Kernel übernommen. Standardmäßig lautet sie OMVSKERN.

```
ADDGROUP OMVSGRP OMVS(GID(1))
ADDUSER OMVSKERN DFLTGRP(OMVSGRP) NOPASSWORD +
        OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
```

INETD ist ein Dämon, der Zugriff auf Funktionen wie setuid() haben muss. Die Benutzer-ID, mit der INETD gestartet wird, benötigt daher für das Profil BPX.DAEMON in der Klasse FACILITY die Zugriffsberechtigung READ. Wenn das Profil nicht definiert ist, muss zwingend UID 0 verwendet werden.

```
PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(OMVSKERN)
```

Die INETD-Benutzer-ID benötigt außerdem die Zugriffsberechtigung EXECUTE für das Programm inetd (/usr/sbin/inetd), die Berechtigung READ für den Zugriff auf die Dateien inetd.conf und ETC.SERVICES sowie die Zugriffsberechtigung WRITE für /etc/inetd.pid. Wenn Sie INETD ohne UID 0 ausführen möchten, können Sie für das Profil SUPERUSER.FILESYS in der Klasse UNIXPRIV die Zugriffsberechtigung CONTROL einrichten, damit die erforderlichen Rechte für z/OS UNIX-Dateien vorhanden sind.

Programme, die eine Dämonberechtigung benötigen, müssen programmgesteuert sein, wenn BPX.DAEMON in der Klasse FACILITY definiert ist. Bei Verwendung des INETD-Standardprogramms (/usr/sbin/inetd) ist dies bereits der Fall. Für Kopien oder eine angepasste Version müssen Sie die Programmsteuerung jedoch manuell definieren. Mit dem Befehl **extattr +p** können Sie eine z/OS UNIX-Datei zu einer programmgesteuerten Datei machen. Mit der RACF-Klasse PROGRAM können Sie aus einem MVS-Lademodul ein programmgesteuertes Modul machen.

Systemprogrammierer, die INETD von ihrer Shellsitzung aus neu starten müssen, verwenden ihre Berechtigungen für den Start von INETD. Deshalb müssen sie dieselben Rechte wie die reguläre INETD-Benutzer-ID haben. Darüber hinaus benötigen sie die Berechtigung, den INETD-Prozess aufzulisten und zu stoppen. Diese Berechtigung kann auf verschiedenen Wegen erteilt werden.

- UID 0

Für Benutzer-IDs von Personen wird dies nicht empfohlen, weil es keine z/OS UNIX-bezogenen Einschränkungen gibt.

- Zugriffsberechtigung READ für das Profil BPX.SUPERUSER in der Klasse FACILITY
Über den Befehl **su** kann der Benutzer ein Benutzer mit der UID 0 werden. Dies ist die empfohlene Konfiguration.
- Zugriff auf Einzelprofile, die die erforderlichen Berechtigungen abdecken
 - Zugriffsberechtigung READ für SUPERUSER.PROCESS.GETPSENT in der Klasse UNIXPRIV (für den Befehl **ps**)
 - Zugriffsberechtigung READ für SUPERUSER.PROCESS.KILL in der Klasse UNIXPRIV (für den Befehl **kill**)

- Zugriffsberechtigung READ für BPX.JOBNAME in der Klasse FACILITY (für die Umgebungsvariable _BPX_JOBNAME)

In der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802) erfahren Sie mehr über die Befehle **extattr** und **su**. Weitere Informationen zur Klasse UNIXPRIV und zu BPX.*-Profilen in der Klasse FACILITY finden Sie in der Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800). Weitere Informationen zum Definieren von OMVS-Segmenten und zur Klasse PROGRAM enthält der *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

Anforderungen von Developer for System z

Developer for System z ist bei der Verwaltung von REXEC und/oder SSH auf INETD angewiesen. Das Produkt kann zusätzliche Anforderungen stellen, die über die oben beschriebenen Anforderungen an die INETD-Konfiguration hinausgehen.

REXEC (oder SSH) wird für die beiden folgenden Zwecke verwendet. Lesen Sie hierzu auch den Abschnitt „REXEC (oder SSH) verwenden (optional)“ auf Seite 102.

- Ferne (hostbasierte) Aktionen in z/OS UNIX-Unterprojekten
- Alternative Startmethode für den RSE-Server

Die fernen Aktionen in z/OS UNIX-Unterprojekten erfordern keine speziellen Einstellungen. Für die alternative Startmethode für den RSE-Server sind jedoch spezielle Einstellungen erforderlich.

INETD

Die Umgebungseinstellungen von INETD werden übergeben, wenn ein Prozess gestartet wird. Die Berechtigungen für die INETD-Benutzer-ID müssen ordnungsgemäß gesetzt sein, damit INETD den RSE-Server starten kann.

- Wenn INETD über JCL mit BPXBATCH gestartet wird, muss die Regionsgröße gleich null sein.
- Wird INETD in einer TSO/OMVS-Shellsession gestartet, muss die Regionsgröße bei mindestens 2096128 liegen.
- Wenn INETD von /etc/rc oder /etc/inittab gestartet wird, wird die Regionsgröße von SYS1.PROCLIB(BPX0INIT) verwendet, die standardmäßig bei 0 liegt.

REXEC (oder SSH)

Der REXEC-Dämon (oder SSH-Dämon), der von INETD gestartet wird, wenn ein Client eine Verbindung mit dem Port 512 (bzw. zum Port 22) herstellt, führt die Authentifizierung durch, startet den RSE-Server und gibt die Portnummer für die weitere Kommunikation an den Client zurück. Die dem RSE-Dämon (SSH-Dämon) in inetd.conf zugeordnete Benutzer-ID benötigt hierfür die folgenden Berechtigungen:

- Gültige OMVS-Definitionen in der Sicherheitssoftware; UID, HOME und PROGRAM müssen gesetzt sein. Außerdem muss GID für die Standardgruppe des Benutzers gesetzt sein.
- Zugriffsberechtigung READ für das Profil BPX.DAEMON in der Klasse FACILITY
- Zugriffsrechte READ und EXECUTE für das Installationsverzeichnis von Developer for System z (standardmäßig /usr/lpp/rdz/*)
- Zugriffsrechte READ und EXECUTE für die Konfigurationsverzeichnisse von Developer for System z (standardmäßig /etc/rdz/*)

Anhang D. APPC konfigurieren

Dieser Anhang soll Sie bei einigen allgemeinen Problemen unterstützen, die beim Konfigurieren von APPC (Advanced Program-to-Program Communication) oder beim Überprüfen oder Modifizieren einer vorhandenen Konfiguration auftreten könnten.

Zusätzliche Informationen zur APPC-Verwaltung und zu den nachfolgend beschriebenen PARMLIB-Membren enthalten die Veröffentlichungen *MVS Planning: APPC/MVS Management* (IBM Form SA22-7599) und *MVS Initialization and Tuning Reference* (IBM Form SA22-7592).

Beachten Sie, dass es sich nicht um eine komplette Konfiguration für APPC handelt. Das Beispiel hebt nur einige wichtige Aspekte hervor, die auf Ihren Standort anwendbar sein könnten.

Das Member SYS1.SAMPLIB(ATBALL) enthält eine Liste und Beschreibungen aller APPC-bezogenen Member (Beispielmember) in SYS1.SAMPLIB.

VSAM

APPC/MVS speichert die Konfigurationsdaten in den folgenden SYS1.PARMLIB-Membren und in zwei VSAM-Dateien:

- Die VSAM-Datei für Transaktionsprogramme (Standardname SYS1.APPCTP) enthält Zeitplanungs- und Sicherheitsinformationen für z/OS-Programme.
- Die VSAM-Datei für Nebeninformationen (Standardname SYS1.APPCSI) enthält die von lokalen z/OS-Transaktionsprogrammen und APPC/MVS-Servern verwendeten Umsetzungen der Namen symbolischer Bestimmungsorte.

Ein Transaktionsprogramm ist ein Anwendungsprogramm, das über APPC mit einem Transaktionsprogramm auf demselben oder einem anderen System kommuniziert, um auf Ressourcen zuzugreifen. Die APPC-Konfiguration für Developer for System z aktiviert ein neues Transaktionsprogramm mit dem Namen FEKFRSRV, das auch als TSO Commands Service bezeichnet wird.

Der folgende Job ist eine Verkettung der Beispielmember SYS1.SAMPLIB(ATBTPVSM) und SYS1.SAMPLIB(ATBSIVSM) und kann zum Definieren der APPC-VSAMs verwendet werden.

```

//APPCVSAM 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 APPC-VSAMs.
/*
//TP      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE CLUSTER (NAME(SYS1.APPCTP) -
                        VOLUME(*****)) -
                        INDEXED REUSE -
                        SHAREOPTIONS(3 3) -
                        RECORDSIZE(3824 7024) -
                        KEYS(112 0) -
                        RECORDS(300 150)) -
        DATA      (NAME(SYS1.APPCTP.DATA)) -
        INDEX      (NAME(SYS1.APPCTP.INDEX))
/*
//SI      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE CLUSTER (NAME(SYS1.APPCSI) -
                        VOLUME(*****)) -
                        INDEXED REUSE -
                        SHAREOPTIONS(3 3) -
                        RECORDSIZE(248 248) -
                        KEYS(112 0) -
                        RECORDS(50 25)) -
        DATA      (NAME(SYS1.APPCSI.DATA)) -
        INDEX      (NAME(SYS1.APPCSI.INDEX))
/*

```

Abbildung 65. JCL zur Erstellung von APPC-VSAM

VTAM

APPC ist eine Implementierung des LU-6.2-Protokolls der Systemnetzwerkarchitektur (SNA). Die SNA stellt Formate und Protokolle bereit, die verschiedene physische und logische SNA-Komponenten definieren, z. B. die logische Einheit (LU, Logical Unit). LU 6.2 ist ein Typ logischer Einheiten, der speziell für die Kommunikation zwischen Anwendungsprogrammen konzipiert ist.

Wenn Sie die SNA in MVS verwenden möchten, müssen Sie VTAM (Virtual Telecommunications Access Method) installieren und konfigurieren. Die APPC-Systemtasks können erst verwendet werden, wenn VTAM aktiv ist.

Der APPC-spezifische Teil der VTAM-Konfiguration umfasst drei Schritte:

1. Definieren Sie den verwendeten Modusnamen (z. B. APPCHOST) für VTAM, indem Sie SYS1.SAMPLIB(ATBLMODE) mit SYS1.SAMPLIB(ATBLJOB) assemblieren und eine Programmverknüpfung zu Ihrer SYS1.VTAMLIB herstellen. Weitere Details finden Sie in der Beschreibung zum Member SYS1.SAMPLIB(ATBLMODE).
2. Definieren Sie APPC/MVS als eine VTAM-Anwendung. Kopieren Sie dazu das Beispielmember SYS1.SAMPLIB(ATBAPPL) in eine Datei der SYS1.VTAMLST-Kette. Weitere Details finden Sie in der Beschreibung zum Member SYS1.SAMPLIB(ATBAPPL).

3. Führen Sie den Konsolbefehl **v net,act,id=atbappl** aus, um die neu definierte Anwendung zu aktivieren. (Hier steht **net** für den Namen Ihrer gestarteten VTAM-Task.) Mit dem Konsolbefehl **d net,appls** können Sie überprüfen, ob die Anwendung aktiv ist. Fügen Sie den Membernamen zu **SYS1.VTAMLST(ATCCONxx)** hinzu, wenn die Anwendung beim Start von VTAM aktiviert werden soll.

Der im Beispielmember **SYS1.SAMPLIB(ATBAPPL)** verwendete **ACBNAME (MVS-LU01)** kann an die Standards Ihres Standortes angepasst werden. Er muss jedoch mit den Definitionen im Member **SYS1.PARMLIB(APPCPMxx)** übereinstimmen.

```

MVS LU01 APPL  ACBNAME=MVS LU01,          C
                APPC=YES,                  C
                AUTOSES=0,                  C
                DDRAINL=NALLOW,             C
                DLOGMOD=APPCHOST,           C
                DMINWNL=5,                   C
                DMINWNR=5,                   C
                DRESPL=NALLOW,              C
                DSESLIM=10,                  C
                LMDENT=19,                   C
                MODETAB=LOGMODES,           C
                PARSESS=YES,                 C
                SECACPT=CONV,                C
                SRBEXIT=YES,                 C
                VPACING=1                    C

```

Abbildung 66. **SYS1.SAMPLIB(ATBAPPL)**

Weitere Informationen zur Konfiguration von VTAM enthält der *Communications Server IP SNA Network Implementation Guide* (IBM FormSC31-8777).

SYS1.PARMLIB(APPCPMxx)

Zur Aktivierung und Unterstützung des Datenaustauschs zwischen Systemen müssen an Standorten LUs (logische Einheiten) definiert werden, zwischen denen Sitzungen stattfinden können. Voraussetzung für eine APPC/MVS-Verarbeitung - auch auf einem einzelnen System - ist, dass an einem Standort mindestens eine LU definiert ist. LUs gehören zu den Elementen, die in **SYS1.PARMLIB(APPCPMxx)** definiert werden.

TSO Commands Service erfordert, dass APPC mit einer Basis-LU für die Bearbeitung ein- und abgehender Anforderungen konfiguriert ist.

Die LU-Definition muss dem Member **SYS1.PARMLIB(APPCPMxx)** hinzugefügt werden und die Parameter **BASE** und **SCHED(ASCH)** enthalten. Das Member **APPCPMxx** gibt auch an, welche VSAM-Dateien mit Transaktionsprofilen (TP) und Nebeninformationen (SI) verwendet werden sollen.

Das folgende Codebeispiel zeigt ein **SYS1.PARMLIB(APPCPMxx)**-Member, das für TSO Commands Service verwendet werden kann.

```

LUADD
  ACBNAME(MVS LU01)
  BASE
  SCHED(ASCH)
  TPDATA(SYS1.APPCTP)
  SIDEINFO DATASET(SYS1.APPCSI)

```

Abbildung 67. **SYS1.PARMLIB(APPCPMxx)**

Wenn ein System mehrere LU-Namen hat, müssen Sie ggf. Änderungen vornehmen, je nachdem, welche LU das System als Basis-LU auswählt. Die Basis-LU für das System wird wie folgt bestimmt:

1. Die Basis-LU des Systems wird von der letzten LUADD-Anweisung angegeben, die die beiden Parameter NOSCHED und BASE enthält. Mit diesem Basis-LU-Typ des Systems können abgehende Anforderungen verarbeitet werden, wenn keine Transaktionsscheduler aktiv sind.
2. Wenn es keine LUADD-Anweisung gibt, die sowohl NOSCHED als auch BASE enthält, wird die Basis-LU des Systems von der letzten LUADD-Anweisung repräsentiert, die den Parameter BASE enthält und ASCH als APPC/MVS-Transaktionsscheduler angibt. Dazu muss entweder SCHED(ASCH) codiert sein oder der Parameter SCHED muss gar nicht codiert sein (denn ASCH ist der Standardwert für SCHED).

Anmerkung: Der Bedienerbefehl **D APPC,LU,ALL** zeigt alle aktiven LU-Definitionen an und markiert die Basis-LU.

Falls es auf Ihrem System eine LU mit den Parametern BASE und NOSCHED gibt, wird diese LU als Basis-LU verwendet werden, TSO Commands Service würde jedoch nicht funktionieren, weil diese LU keinen Transaktionsscheduler für die Bearbeitung von Anforderungen an die Transaktion FEKFRSRV hat. Wenn Sie den Parameter NOSCHED dieser LU nicht entfernen können, setzen Sie die Umgebungsvariable `_FEKFSCMD_PARTNER_LU` in `rsed.envvars` auf die LU, für die die Parameter BASE und SCHED(ASCH) definiert sind. Beispiel:

```
_FEKFSCMD_PARTNER_LU=MVSLU01
```

Weitere Informationen zu `rsed.envvars` enthält der Abschnitt „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 33.

SYS1.PARMLIB(ASCHPMxx)

Der APPC/MVS-Transaktionsscheduler (mit dem Standardnamen ASCH) leitet bei eingehenden Dialoganforderungen Transaktionsprogramme ein und steuert den zeitlichen Ablauf dieser Programme. Das Member `SYS1.PARMLIB(ASCHPMxx)` steuert seine Funktionen unter anderem mit Transaktionsklassendefinitionen.

Die für TSO Commands Service verwendete APPC-Transaktionsklasse muss genug APPC-Initiatoren haben, damit für jeden Benutzer von Developer for System z ein Initiator verfügbar ist.

TSO Commands Service erfordert außerdem die Angabe der Standardspezifikationen in den Abschnitten `OPTIONS` und `TPDEFAULT`.

Das folgende Codebeispiel zeigt das Member `SYS1.PARMLIB(ASCHPMxx)`, das für TSO Commands Service verwendet werden kann.

```
CLASSADD  
  CLASSNAME(A)  
  MAX(20)  
  MIN(1)  
  MSGLIMIT(200)
```

```
OPTIONS  
  DEFAULT(A)
```

```
TPDEFAULT  
  REGION(2M)  
  TIME(5)  
  MSGLEVEL(1,1)  
  OUTCLASS(X)
```

Abbildung 68. *SYS1.PARMLIB(ASCHPMxx)*

Anmerkung:

- Das IBM Support Center kann Sie bitten, für das Debugging den Wert von MSGLIMIT zu erhöhen, damit mehr Ausgaben in die Protokolldatei geschrieben werden.
- Der Bedienerbefehl **D ASCH,ALL** zeigt alle aktiven APPC-Transaktionsschedulerklassen an.

APPC-Änderungen aktivieren

Die in den obigen Schritten dokumentierten Konfigurationsänderungen können jetzt aktiviert werden. Hierfür gibt es - je nach vorliegender Situation - verschiedene Möglichkeiten:

- Wenn APPC noch nicht aktiv ist, geben Sie die folgenden Konsolbefehle ein, um APPC/MVS zu starten. (Hier steht xx für die beiden letzten Zeichen der zugehörigen SYS1.PARMLIB-Member.)
 1. S APPC,SUB=MSTR,APPC=xx
 2. S ASCH,SUB=MSTR,ASCH=xxFügen Sie diese Befehle zu SYS1.PARMLIB(COMMNDxx) hinzu, damit sie beim Systemstart gestartet werden.
- APPC ist bereits aktiv. APPC kann die SYS1.PARMLIB-Member mit dem folgenden Konsolbefehl **SET** dynamisch neu laden. (Wobei xx für die beiden letzten Zeichen der zugehörigen SYS1.PARMLIB-Member steht):
 1. SET APPC=xx
 2. SET ASCH=xx

Mit den Konsolbefehlen **D APPC** und **D ASCH** kann die APPC-Konfiguration überprüft werden. Weitere Informationen zu den genannten Konsolbefehlen enthält die Veröffentlichung *MVS System Commands* (IBM Form GC28-1781).

Transaktion für TSO Commands Service definieren

Wenn APPC/MVS aktiv ist, kann TSO Commands Service von Developer for System z definiert werden. Lesen Sie hierzu die Beschreibung im Abschnitt „APPC-Transaktion für TSO Commands Service (optional)“ auf Seite 104.

Für eine dokumentierte Definition der APPC-Transaktion müssen Sie FEK.#CUST.JCL(FEKAPPC) anpassen und übergeben.

Die APPC-Transaktion kann auch interaktiv über die APPC-ISPF-Schnittstelle definiert werden. Diese Schnittstelle ist in einem White Paper dokumentiert. In diesem White Paper ist auch beschrieben, wie die APPC-Transaktion für die Erfassung benutzerspezifischer Accountinformationen konfiguriert werden kann.

Das White Paper *APPC and WebSphere Developer für System z* (IBM Form SC23-5885-00) ist in der Internetbibliothek für Developer for System z unter <http://www-306.ibm.com/software/awdtools/rdz/library/> verfügbar.

Anmerkung: In Developer for System z Version 7.1 hat sich die von APPC zum Starten von TSO Commands Service verwendete Transaktionsprogramm-JCL geändert. Wenn Sie den im White Paper beschriebenen Anweisungen für das Definieren des Transaktionsprogramms folgen, müssen Sie das Schlüsselwort NESTMACS zur PARM-Zeile hinzufügen. Beispiel:

```
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
```

Alternative Konfigurationsoptionen (optional)

Developer for System z unterstützt alternative Optionen für die APPC- und VTAM-Konfiguration, von denen nachfolgend einige dokumentiert sind.

Alternativer Transaktionsname

Der Standardtransaktionsname für TSO Commands Service ist "FEKFRSRV". Lesen Sie hierzu die Informationen unter „APPC-Transaktion für TSO Commands Service (optional)“ auf Seite 104. Im gleichen Abschnitt ist unter anderem angegeben, dass dieser Name geändert werden kann, wenn die Transaktion für APPC definiert wird.

Falls Sie den Transaktionsnamen in APPC ändern, muss der neue Name `_FEKFSCMD_TP_NAME_` in `rsed.envvars` zugeordnet werden. Lesen Sie hierzu die Beschreibung im Abschnitt „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 33.

Mehrere LUs

APPC ist ein Kommunikationsprotokoll, über das ein Programm (der Partnerknoten) mit einem Programm auf dem Host (dem lokalen Knoten) interagieren kann. Bei Developer for System z sind der Partnerknoten (TSO Commands-Server) und der lokale Knoten (RSE-Server) auf demselben z/OS-System aktiv. Standardmäßig verwenden beide für die Kommunikation untereinander dieselbe Basis-LU-Definition.

In `rsed.envvars` können Sie in der Anweisung `_FEKFSCMD_PARTNER_LU_` einen alternativen Partner-LU-Namen für TSO Commands Service angeben. Lesen Sie hierzu die Beschreibung im Abschnitt „RSE-Konfigurationsdatei `rsed.envvars`“ auf Seite 33. Die lokale LU kann nicht geändert werden und muss immer eine gültige Basis-LU (mit den Schlüsselwörtern BASE und SCHED) sein.

LU-Sicherheit

VTAM unterstützt eine sichere APPC-Konfiguration, bei der die Kommunikation zwischen der Partner-LU und der lokalen LU für die Sicherheitssoftware definiert werden muss.

Sie können diese Konfiguration aktivieren, indem Sie zur VTAM-Definition der lokalen LU (Basis-LU) VERIFY=REQUIRED hinzufügen. Die Sicherheitsdefinitionen müssen in der Klasse APPCLU erfolgen. Lesen Sie die diesbezüglichen Informationen in der Veröffentlichung *MVS Planning: APPC/MVS Management* (IBM Form SA22-7599).

Wenn diese Konfiguration in VTAM aktiv ist und die Konfiguration in Ihrer Sicherheitssoftware noch nicht abgeschlossen ist, kann die Kommunikation mit TSO Commands Service nicht initialisiert werden. Das Systemprotokoll enthält in dem Fall keine Nachricht, dass VTAM den Aufbau der Verbindung zurückgewiesen hat. Der APPC-IVP-Test (fekfi vpa) scheitert mit der Nachricht "Return code 1 - Allocate Failure no retry".

Anhang E. Voraussetzungen

In diesem Anhang werden die Hostvoraussetzungen und die zusätzlich erforderliche Software für diese Version von Developer for System z aufgelistet.

Eine aktuelle Liste der erforderlichen und optionalen Voraussetzungen enthält die Veröffentlichung *Rational Developer for System z Prerequisites* (IBM Form SC23-7659) in der Onlinebibliothek für Developer for System z unter <http://www-01.ibm.com/software/awdtools/rdz/library/>.

Die in diesem Abschnitt aufgelisteten Produkte sind alle zum Zeitpunkt der Veröffentlichung dieses Handbuchs verfügbar. Rufen Sie die Website "IBM Software Support Lifecycle" unter <http://www.ibm.com/software/support/lifecycle/> auf, um zu prüfen, ob ein ausgewähltes Produkt zu dem Zeitpunkt, an dem Sie die zugehörige Funktion in Developer for System z verwenden möchten, immer noch verfügbar ist.

Hostvoraussetzungen für z/OS

Für die Verwendung von Developer for System z ist die folgende Umgebung mit den entsprechenden Voraussetzungen erforderlich:

z/OS

Auf dem Host muss eine der folgenden Versionen installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5694-A01	z/OS Version 1.11	ISPF: <ul style="list-style-type: none">• APAR OA29489 (für das TSO/ISPF-Client-Gateway) PTF UA51713 TCP/IP: <ul style="list-style-type: none">• Kein PTF oder Service-Level erforderlich
5694-A01	z/OS Version 1.10	ISPF: <ul style="list-style-type: none">• APAR OA29489 (für das TSO/ISPF-Client-Gateway) PTF UA51712 TCP/IP: <ul style="list-style-type: none">• APAR PK74282 (Zuwachs im festgelegten Speicher in CSM) PTF UK41810
5694-A01	z/OS Version 1.9	ISPF: <ul style="list-style-type: none">• APAR OA29489 (für das TSO/ISPF-Client-Gateway) PTF UA51687 TCP/IP: <ul style="list-style-type: none">• APAR PK74282 (Zuwachs im festgelegten Speicher in CSM) PTF UK41812

Programm-nummer	Produktname	Erforderliche PTFs oder Service-Level
5694-A01	z/OS Version 1.8	ISPF: <ul style="list-style-type: none"> • APAR OA20345 (Ausgabe verschachtelter Befehle) PTF UA33575 • APAR OA20449 (NESTMACS-Unterstützung aufnehmen) PTF UA34052 • APAR OA29489 (für das TSO/ISPF-Client-Gateway) PTF UA51686 TCP/IP: <ul style="list-style-type: none"> • APAR PK74282 (Zuwachs im festgelegten Speicher in CSM) PTF UK41811

Die zugehörige Produktwebsite ist:

<http://www-03.ibm.com/systems/z/os/zos/>

Anmerkungen:

1. Ferne (hostbasierte) Aktionen für z/OS UNIX-Unterprojekte erfordern, dass auf dem Host die z/OS UNIX-Version von REXEC oder SSH aktiv ist.
2. Zu z/OS gehören die folgenden Komponenten, die installiert, konfiguriert und zur Ausführung bereit sein müssen:
 - Interactive System Productivity Facility (ISPF)
 - <http://www-01.ibm.com/software/awdtools/ispf/>
 - Language Environment
 - <http://www-03.ibm.com/servers/eserver/zseries/zos/le/>
 - RACF oder ein funktional entsprechendes Sicherheitsprodukt
 - <http://www-03.ibm.com/servers/eserver/zseries/zos/racf/>
 - VTAM-Komponente von IBM Communications Server
 - <http://www-01.ibm.com/software/network/commserver/zos/>
 - IP-Servicekomponente von IBM Communications Server
 - <http://www-01.ibm.com/software/network/commserver/zos/>
 - Binder
 - APPC (optional)

Anmerkung:

- APPC ist eine obligatorische Voraussetzung, wenn der Service für ISPF-APAR OA29489 auf Ihrem Hostsystem nicht verfügbar ist.
- Sie können APPC durch die Client-Gateway-Funktionalität ersetzen, die mit ISPF für z/OS 1.10 bereitgestellt wird und in ISPF für z/OS 1.8 und 1.9 verfügbar ist, wenn die entsprechenden Programmkorrekturen (PTFs) angewendet wurden.

SMP/E

Um Developer for System z installieren zu können, muss eine der folgenden Versionen installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5655-G44	IBM System Modification Program Extended (SMP/E) für z/OS Version 3.5	Kein PTF oder Service-Level erforderlich
5655-G44	IBM System Modification Program Extended (SMP/E) für z/OS Version 3.4	Kein PTF oder Service-Level erforderlich

Die zugehörige Produktwebsite ist:

<http://www-03.ibm.com/systems/z/os/zos/smpe/>

SDK für z/OS Java 2 Technology Edition

Für eine Unterstützung von Anwendungen, die Remote Systems Explorer (RSE) verwenden, muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5655-R32	IBM 64-Bit SDK for z/OS, Java 2 Technology Edition Version 6.0	Service-Release 7
5655-R31	IBM 31-Bit SDK for z/OS, Java 2 Technology Edition Version 6.0	Service-Release 7
5655-N99	IBM 64-Bit SDK for z/OS, Java 2 Technology Edition Version 5.0	Service-Release 11
5655-N98	IBM 31-Bit SDK for z/OS, Java 2 Technology Edition Version 5.0	Service-Release 11

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/servers/eserver/zseries/software/java/>

Anmerkung: Bei Verwendung einer 64-Bit-Version von Java muss die vorläufige Programmkorrektur für Developer for System z "APAR PM07305" angewendet werden. Die vorläufige Programmkorrektur ist über die Seite für empfohlene Services für Developer for System z" verfügbar: <http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335>.

Zusätzliche Hostvoraussetzungen für z/OS

Die in diesem Abschnitt aufgelisteten Produkte und angegebenen Softwareprogramme sind zur Unterstützung bestimmter Features von Developer for System z erforderlich. Der Workstation-Client von Developer for System z kann ohne diese vorausgesetzten Produkte fehlerfrei installiert werden. Zur Laufzeit müssen die als erforderlich angegebenen Produkte jedoch installiert und betriebsbereit sein, damit das entsprechende Feature ordnungsgemäß funktionieren kann.

z/OS

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5694-A01	z/OS Version 1.11	HLASM Kein PTF oder Service-Level erforderlich XL C/C++ Kein PTF oder Service-Level erforderlich SCLM Kein PTF oder Service-Level erforderlich LE (PL/I) Kein PTF oder Service-Level erforderlich TN3270 Kein PTF oder Service-Level erforderlich
5694-A01	z/OS Version 1.10	HLASM Kein PTF oder Service-Level erforderlich XL C/C++ Kein PTF oder Service-Level erforderlich SCLM Kein PTF oder Service-Level erforderlich LE (PL/I) Kein PTF oder Service-Level erforderlich TN3270 Kein PTF oder Service-Level erforderlich
5694-A01	z/OS Version 1.9	HLASM Kein PTF oder Service-Level erforderlich XL C/C++ Kein PTF oder Service-Level erforderlich SCLM • APAR OA27379 (SCLM-Suche) PTF UA46330 + UA46331, UA46332, UA46333, UA46334 (sprachabhängig) LE (PL/I) Kein PTF oder Service-Level erforderlich TN3270 Kein PTF oder Service-Level erforderlich

Programm-nummer	Produktname	Erforderliche PTFs oder Service-Level
5694-A01	z/OS Version 1.8	HLASM Kein PTF oder Service-Level erforderlich XL C/C++ Kein PTF oder Service-Level erforderlich SCLM <ul style="list-style-type: none"> • APAR OA21104 (Build im Informationsmodus) PTF UA35046 + UA35056, UA35057, UA35058 oder UA35059 (sprachabhängig) • APAR OA16924 (Erweiterung von SCLMINFO) PTF UA29772 + UA29922, UA29923, UA29924 oder UA29925 (sprachabhängig) • APAR OA16804 (Unterstützung für Ersatzbenutzer-ID aufnehmen) PTF UA33524 + UA33533, UA33534, UA33535 oder UA33536 (sprachabhängig) LE (PL/I) <ul style="list-style-type: none"> • APAR PK41552 (neue PL/I-Nachrichten für Developer for System z) PTF UK24482 (Englisch) oder UK24483 (Japanisch) TN3270 Kein PTF oder Service-Level erforderlich

Die zugehörige Produktwebsite ist:

<http://www-03.ibm.com/systems/z/os/zos/>

Anmerkung:

1. JES3 Version 1.10 oder höher ist eine zusätzliche Voraussetzung für JES3-Benutzer, die die Unterstützung von Job Monitor für das Anzeigen der Ausgabe aktiver Jobs verwenden.
2. High Level Assembler (HLASM) muss auf dem Host mit dem angegebenen Service-Level installiert sein, um in Developer for System z entwickelte oder bearbeitete Assemblerprogramme kompilieren zu können.

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/hlasm/>

3. Der XL C/C++-Compiler muss auf dem Host mit dem angegebenen Service-Level installiert sein, um in Developer for System z entwickelte oder bearbeitete C/C++-Programme kompilieren zu können.

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/czos/>

4. SCLM muss auf dem Host mit dem angegebenen Service-Level installiert sein, damit SCLM Developer Toolkit unterstützt wird.

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/scmsuite/sclm/>

Anmerkung:

- Der APAR OA16804 ist nur für eine geschützte Builderstellung und Umstufung sowie ein geschütztes Deployment erforderlich.
- Der APAR OA26997 ist nur für die Unterstützung der Membersicherheit erforderlich.
- Der APAR OA27379 ist nur für die Unterstützung der Membersicherheit oder der SCLM-Suchfunktionalität erforderlich.

5. Language Environment muss auf dem Host mit dem angegebenen Service-Level installiert sein, damit Enterprise Service Tools für PL/I unterstützt wird.

Die zugehörige Produktwebsite ist:

<http://www-03.ibm.com/servers/eserver/zseries/zos/le/>

6. TN3270 muss auf dem Host mit dem angegebenen Service-Level installiert sein, damit der Host-Connect-Emulator unterstützt wird. TN3270 ist ein Teil der IP-Servicekomponente von IBM Communications Server.

Die zugehörige Produktwebsite ist:

<http://www-01.ibm.com/software/network/commserver/zos/>

COBOL-Compiler

Um in Developer for System z entwickelte oder bearbeitete COBOL-Programme kompilieren zu können, muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5655-S71	IBM Enterprise COBOL für z/OS Version 4.2	Kein PTF oder Service-Level erforderlich
5655-S71	IBM Enterprise COBOL für z/OS Version 4.1	Kein PTF oder Service-Level erforderlich
5535-G53	IBM Enterprise COBOL für z/OS Version 3.4	Kein PTF oder Service-Level erforderlich

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/cobol/zos/>

Anmerkung: Für die compilergestützte XML-Konvertierung unter Verwendung der auf XMLSS basierenden Funktion XML PARSE erfordert Enterprise Service Tools IBM Enterprise COBOL für z/OS Version 4.1.

PL/I-Compiler

Um in Developer for System z entwickelte oder bearbeitete PL/I-Programme kompilieren zu können, muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5655-H31	IBM Enterprise PL/I für z/OS Version 3.9	Kein PTF oder Service-Level erforderlich
5655-H31	IBM Enterprise PL/I für z/OS Version 3.8	Kein PTF oder Service-Level erforderlich
5655-H31	IBM Enterprise PL/I für z/OS Version 3.7	Kein PTF oder Service-Level erforderlich
5655-H31	IBM Enterprise PL/I für z/OS Version 3.6	Kein PTF oder Service-Level erforderlich

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/pli/plizos/>

Debug Tool for z/OS

Für eine Unterstützung von fernem Debugging aus Developer for System z heraus muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Programmiersprache	Erforderliche APARs, PTFs oder Service-Level
5655-V50	IBM Debug Tool for z/OS Version 10.1	COBOL, PL/I, C/C++, Assembler und zusätzliche Features	Alle verfügbaren Korrekturen und Service-Level
5655-U27	IBM Debug Tool for z/OS Version 9.1	COBOL, PL/I, C/C++, Assembler und zusätzliche Features	Alle verfügbaren Korrekturen und Service-Level
5655-S16	IBM Debug Tool Utilities and Advanced Functions for z/OS Version 8.1.0	COBOL, PL/I, C/C++, Assembler und zusätzliche Features	Alle verfügbaren Korrekturen und Service-Level
5655-S17	IBM Debug Tool for z/OS Version 8.1.0	COBOL, PL/I, Assembler, C/C++	Alle verfügbaren Korrekturen und Service-Level

Anmerkung: Für die Unterstützung von CICS-Debugkonfigurationen in IBM Rational Developer for System z Version 7.6.1 oder höher ist IBM Debug Tool Version 10.1 oder Version 9.1 (PTF-Nummer: UK52904) erforderlich.

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/debugtool/>

Anmerkung: Debug Tool Utilities and Advanced Functions (DTU & AF) ist ein übergeordnetes Produkt zu Debug Tool.

Ab Version 9 werden Debug Tool for z/OS sowie Debug Tool Utilities and Advanced Functions in einem Angebot zusammengefasst.

CICS Transaction Server

Für eine Unterstützung von Anwendungen mit eingebetteten CICS-Anweisungen muss eine der folgenden Versionen installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5655-S97	IBM CICS Transaction Server für z/OS Version 4.1	Kein PTF oder Service-Level erforderlich
5697-E93	IBM CICS Transaction Server für z/OS Version 3.2	UK34221
5697-E93	IBM CICS Transaction Server für z/OS Version 3.1	UK15767, UK15764, UK11782, UK11294, UK12233, UK12521, UK15261, UK15271, UK34221, UK34078

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/http/cics/tserver/>

Anmerkung:

- Für CICS Transaction Server sind zusätzliche Konfigurationsschritte erforderlich, um die Zusammenarbeit des Servers mit Debug Tool zu gewährleisten.
- Für eine Unterstützung von Application Deployment Manager, Service Component Architecture und Features von Enterprise Service Tools, die in IBM Rational Developer for System z Version 7.6 oder höher neu implementiert wurden, ist die RESTful-Schnittstelle erforderlich, die in CICS Transaction Server Version 4.1 oder höher verfügbar ist.
- Für die Unterstützung vieler Features von Enterprise Service Tools ist CICS Transaction Server Version 3.2 oder höher erforderlich.
Eine vollständige Liste der Spezifikationen für die Laufzeitanforderungen finden Sie in der Dokumentation zu Enterprise Service Tools im Information Center für IBM Rational Developer for System z unter <http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/>.
- Für Application Deployment Manager wird mindestens CICS Transaction Server Version 3.1 mit dem Service UK34221 vorausgesetzt.

IMS

Für eine Unterstützung von Anwendungen, die IMS Database und Data Communications verwenden, muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5635-A02	IBM IMS Version 11.1	Kein PTF oder Service-Level erforderlich
5635-A01	IBM IMS Version 10.1	Kein PTF oder Service-Level erforderlich
5655-J38	IBM IMS Version 9.1	Kein PTF oder Service-Level erforderlich

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/data/ims/ims/>

Anmerkung:

- Für IMS sind zusätzliche Konfigurationsschritte erforderlich, um die Zusammenarbeit des Produkts mit Debug Tool zu gewährleisten.
- Für Enterprise Service Tools ist die Version 10.1 oder höher von IMS, IMS Connect und IMS SOAP Gateway erforderlich.

DB2 for z/OS

Für eine Unterstützung von DB2 muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5635-DB2	IBM DB2 für z/OS Version 9.1	Kein PTF oder Service-Level erforderlich
5625-DB2	IBM DB2 Universal Database für z/OS Version 8.1	Kein PTF oder Service-Level erforderlich

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/data/db2/zos/>

Rational Team Concert for System z

Für eine Jazz-basierte Quellcodeverwaltung mithilfe von fernen Projekten in Developer for System z muss die folgende Version installiert sein.

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5724-V82	Rational Team Concert for System z - Server Version 2.0	<p>FMID HAHA200 – Team Server</p> <ul style="list-style-type: none">• UK54064• UK54071• UK54073• UK54095• UK54098 <p>FMID HAHB200 – Toolkit</p> <ul style="list-style-type: none">• UK54065• UK54066• UK54099 <p>FMID HAHC200 – Job Monitor</p> <ul style="list-style-type: none">• Kein PTF oder Service-Level erforderlich <p>FMID HAHD200 – BuildForge Agent</p> <ul style="list-style-type: none">• UK54097

Die zugehörige Produktwebsite ist:

<http://www-01.ibm.com/software/awdtools/rtc/>

Anmerkung: Rational Team Concert - Server Version 1.0 oder Rational Team Concert for System z - Server Version 1.0.1 bieten teilweise Unterstützung für einige Funktionen von Developer for System z. Dazu gehören beispielsweise lokale Projekte. Es wird Rational Team Concert for System z - Server Version 2.0.0.2 empfohlen, da es besser integriert ist und die Features voll unterstützt.

File Manager

Für eine Unterstützung der Integration von File Manager muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5655-U29	IBM File Manager for z/OS Version 10.1	• UK54428

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/filemanager/>

Fault Analyzer

Für eine Unterstützung der Integration von Fault Analyzer muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5655-V51	IBM Fault Analyzer Version 10.1	Kein PTF oder Service-Level erforderlich
5655-U28	IBM Fault Analyzer Version 9.1	Kein PTF oder Service-Level erforderlich
5655-S15	IBM Fault Analyzer Version 8.1	Kein PTF oder Service-Level erforderlich

Die zugehörige Produktwebsite ist:

<http://www.ibm.com/software/awdtools/faultanalyzer/>

REXX

Um SCLM Developer Toolkit verwenden zu können, muss eine der folgenden Versionen auf dem Host installiert sein:

Programmnummer	Produktname	Erforderliche PTFs oder Service-Level
5695-014	IBM Library for REXX on zSeries Version 1.4	Kein PTF oder Service-Level erforderlich
5695-014	IBM Library for REXX on zSeries Alternate Library Version 1.4.0 (FMIDs HWJ9143, JWJ9144)	Kein PTF oder Service-Level erforderlich

Auf der Produktwebsite steht eine Version von REXX/370 Alternate Library zur Verfügung:

<http://www.ibm.com/software/awdtools/rexx/rexxzseries/>

Ported Tools

Wenn Sie für ein sicheres Deployment in SCLM Developer Toolkit sftp oder scp verwenden möchten, müssen die IBM Ported Tools für z/OS (unter z/OS UNIX) installiert sein.

Auf der Produktwebsite steht eine Version von IBM Ported Tools for z/OS zur Verfügung:

http://www-03.ibm.com/servers/eserver/zseries/zos/unix/port_tools.html

Ant

Für JAVA/J2EE-Builds mit SCLM Developer Toolkit muss Apache Ant (unter z/OS UNIX) installiert sein.

Apache Ant ist ein Java-basiertes Open-Source-Build-Tool, das Sie von der Produktwebsite herunterladen können:

<http://ant.apache.org/>

Endevor®

CA Endevor® Software Change Manager Release 12 muss installiert sein, um die Schnittstelle für CA Endevor® SCM in Developer for System z verwenden zu können.

CA Endevor® SCM ist ein Produkt von CA. Die zugehörige Produktwebsite ist:

<http://www.ca.com/us/products/product.aspx?ID=259>

Literaturübersicht

Referenzierte Veröffentlichungen

In diesem Dokument werden die folgenden Veröffentlichungen referenziert:

Tabelle 52. Referenzierte Veröffentlichungen

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
Java Diagnostic Guide	SC34-6650	Java 5.0	http://www.ibm.com/developerworks/java/jdk/diagnosis/
Java SDK and Runtime Environment User Guide	/	Java 5.0	http://www-03.ibm.com/servers/eserver/zseries/software/java/
Program Directory for IBM Rational Developer for System z	GI11-8298	Developer for System z	http://www-306.ibm.com/software/awdtools/rdz/library/
Rational Developer for System z Common Access Repository Manager Developer's Guide	SC23-7660	Developer for System z	http://www-306.ibm.com/software/awdtools/rdz/library/
Rational Developer for System z Voraussetzungen	SC23-7659	Developer for System z	http://www-306.ibm.com/software/awdtools/rdz/library/
Rational Developer für System z Handbuch für den Schnelleinstieg in die Hostkonfiguration	GI11-3191	Developer for System z	http://www-306.ibm.com/software/awdtools/rdz/library/
Rational Developer für System z Hostplanung	GI11-3123	Developer for System z	http://www-306.ibm.com/software/awdtools/rdz/library/
SCLM Developer Toolkit Administrator's Guide	SC23-9801	Developer for System z	http://www-306.ibm.com/software/awdtools/rdz/library/
APPC and WebSphere Developer for System z	SC23-5885	White Paper	http://www-306.ibm.com/software/awdtools/rdz/library/
Communications Server IP Configuration Guide	SC31-8775	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Configuration Reference	SC31-8776	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Diagnosis Guide	GC31-8782	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP System Administrator's Commands	SC31-8781	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA Network Implementation Guide	SC31-8777	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA Operations	SC31-8779	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Cryptographic Services System SSL Programming	SC24-5901	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
DFSMS Macro Instructions for Data Sets	SC26-7408	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

Tabelle 52. Referenzierte Veröffentlichungen (Forts.)

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
DFSMS Using data sets	SC26-7410	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Language Environment Customization	SA22-7564	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Language Environment Debugging Guide	GA22-7560	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL Reference	SA22-7597	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning APPC/MVS Management	SA22-7599	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning Workload Management	SA22-7602	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS System Commands	SA22-7627	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E Customization	SA22-7783	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E REXX Reference	SA22-7790	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Command Reference	SA22-7802	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Planning	GA22-7800	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services User's Guide	SA22-7801	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Using REXX and z/OS UNIX System Services	SA22-7806	z/OS 1.9	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Resource Definition Guide	SC34-6430	CICS TS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
Resource Definition Guide	SC34-6815	CICSTS 3.2	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
Resource Definition Guide	SC34-7000	CICSTS 4.1	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
RACF Security Guide	SC34-6454	CICS TS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
RACF Security Guide	SC34-6835	CICSTS 3.2	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html

Tabelle 52. Referenzierte Veröffentlichungen (Forts.)

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
RACF Security Guide	SC34-7003	CICSTS 4.1	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
Language Reference	SC27-1408	Enterprise COBOL für z/OS	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html

In diesem Dokument werden die folgenden Websites referenziert:

Tabelle 53. Referenzierte Websites

Beschreibung	Referenzwebsite
Developer für System z - Information Center	http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp
Developer für System z - Support	http://www-306.ibm.com/software/awdtools/rdz/support/
Developer für System z - Library	http://www-306.ibm.com/software/awdtools/rdz/library/
Homepage von Developer for System z	http://www-306.ibm.com/software/awdtools/rdz/
Empfohlene Services für Developer for System z	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335
Verbesserungsvorschlag für Developer for System z	https://www.ibm.com/developerworks/support/rational/rfe/
z/OS-Internetbibliothek	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Information Center für CICSTS	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp
Download von Apache Ant	http://ant.apache.org/
Java-Keytool-Dokumentation	http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html
Homepage der CA-Unterstützung	https://support.ca.com/

Veröffentlichungen mit weiteren Informationen

Die folgenden Veröffentlichungen können Antworten auf Fragen enthalten, die vielleicht bei der Konfiguration erforderlicher Hostkomponenten auftauchen.

Tabelle 54. Veröffentlichungen mit weiteren Informationen

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	http://www.redbooks.ibm.com/
System Programmer's Guide to: Workload Manager	SG24-6472	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	http://www.redbooks.ibm.com/
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	http://www.redbooks.ibm.com/

Glossar

A

Aktions-ID. Eine numerische Kennung zwischen 0 und 999 für eine Aktion.

Antwortdatei.

1. Eine Datei, die vordefinierte Antworten auf Fragen enthält, die ein Programm stellt. Die Antworten werden verwendet, sodass diese Werte nicht einzeln eingegeben werden müssen.
2. Eine ASCII-Datei, die mit Installations- und Konfigurationsdaten angepasst werden kann, die eine Installation automatisieren. Die Installations- und Konfigurationsdaten müssen während einer interaktiven Installation eingegeben werden, aber mit einer Antwortdatei kann die Installation ohne jeglichen Benutzereingriff durchgeführt werden.

Anwendungsserver.

1. Ein Programm, das alle Anwendungsoperationen zwischen browserbasierten Computern und den Back-End-Geschäftsanwendungen oder -Datenbanken einer Organisation bearbeitet. Es gibt eine spezielle Klasse von Java-basierten Anwendungsservern, die dem Standard J2EE entsprechen. J2EE-Code kann ohne großen Aufwand zwischen diesen Anwendungsservern portiert werden. Diese Anwendungsserver können JSPs und Servlets für dynamischen Webinhalt und EJBs für Transaktionen und Datenbankzugriffe unterstützen.
2. Das Ziel einer Anforderung, die von einer fernen Anwendung stammt. In der DB2-Umgebung wird die Anwendungsserverfunktion von der Distributed Data Facility bereitgestellt und für den Zugriff auf DB2-Daten in fernen Anwendungen verwendet.
3. Ein Serverprogramm in einem verteilten Netz, das die Ausführungsumgebung für ein Anwendungsprogramm bereitstellt.
4. Das Ziel einer Anforderung, die von einem Anwendungsrequester stammt. Das Datenbankverwaltungssystem (DBMS) auf der Anwendungsserversite stellt die angeforderten Daten bereit.
5. Software, die die Kommunikation mit dem Client, der ein Asset anfordert, und Abfragen von Content Manager bearbeitet.

Ausgabesicht. Zeigt Nachrichten, Parameter und Ergebnisse an, die sich auf die von Ihnen bearbeiteten Objekte beziehen

Ausgabesicht der Konsole. Zeigt die Ausgabe eines Prozesses an und ermöglicht Ihnen, über die Tastatur Eingaben an einen Prozess zu senden

B

Bidirektional (BIDI). Bezeichnung für Scripts in Sprachen wie Arabisch und Hebräisch, die im Allgemeinen von rechts nach links geschrieben werden. Ausnahmen sind Zahlen, die von links nach rechts geschrieben werden. Diese Definition stammt aus dem LISA-Glossar (Localization Industry Standards Association).

Bidirektionales Attribut. Texttyp, Textausrichtung, numerische Ersetzung und symmetrische Ersetzung

Buildanforderung. Eine Anforderung eines Clients zum Ausführen einer Buildtransaktion

Buildtransaktion. Ein unter MVS gestarteter Job, der Builds erstellt, wenn vom Client eine Buildanforderung empfangen wird

C

Container.

1. In CoOperative Development Environment/400 ein Systemobjekt, das Quellendateien enthält und organisiert. Beispiele für einen Container sind eine i5/OS-Bibliothek und eine partitionierte MVS-Datei.
2. In J2EE eine Entität, die Komponenten Sicherheits-, Deployment- und Laufzeitservices sowie Services für die Verwaltung des Lebenszyklus bereitstellt. (Sun) Jeder Containertyp (EJB, Web, JSP, Servlet, Applet und Anwendungsclient) stellt außerdem komponentenspezifische Services bereit.
3. In Backup Recovery and Media Services das physische Objekt, das zum Lagern und Umlagern von Datenträgern verwendet wird, wie z. B. Boxen, Schachteln oder Regale
4. In einem Virtual Tape Server (VTS) ein Behälter, in dem exportierte logische Datenträger gespeichert werden können. Ein Stapeldatenträger mit einem oder mehreren logischen Datenträger(n), der sich außerhalb einer VTS-Bibliothek befindet, wird als Container für diese Datenträger betrachtet.
5. Eine physische Speicherposition der Daten, z. B. eine Datei, ein Verzeichnis oder eine Einheit
6. Eine Spalte oder Zeile, die verwendet wird, um das Layout eines Portlets oder anderer Container auf einer Seite zu gestalten
7. Ein Element der Benutzerschnittstelle, das Objekte enthält. Im Ordnermanager ein Objekt, das andere Ordner oder Dokumente enthalten kann

D

Datei. Die Haupteinheit für das Speichern und Abrufen von Daten, die sich aus einer Sammlung von Daten in einer von mehreren vorgegebenen Zusammenstellungen zusammensetzt und durch Steuerinformationen beschrieben wird, auf die das System Zugriff hat

Datenbank. Eine Sammlung von in Wechselbeziehung zueinander stehenden oder unabhängigen Datenelementen, die zur Bereitstellung für eine oder mehrere Anwendung(en) zusammen gespeichert werden

Datendefinitionssicht. Enthält eine lokale Darstellung von Datenbanken und ihren Objekten und stellt Features für die Bearbeitung dieser Objekte und deren Export in eine ferne Datenbank bereit

Debug. Fehler in Programmen finden, diagnostizieren und beheben

Debugsitzung. Die Debugaktivitäten, die in dem Zeitraum zwischen dem Starten eines Debuggers durch den Entwickler und dem Beenden des Debuggers stattfinden

F

Fehlerpuffer. Ein Teil des Speichers, in dem Fehlernachrichten vorübergehend gespeichert werden

Fernes Dateisystem. Ein Dateisystem, das sich auf einem anderen Server oder Betriebssystem befindet

Fernes System. Jedes andere System im Netz, mit dem Ihr System kommunizieren kann

G

Gateway.

1. Eine Middlewarekomponente, die eine Brücke zwischen Internet und Intranetumgebungen während Web-Service-Aufrufen bildet
2. Software, die Services zwischen Endpunkten und dem Rest der Tivoli-Umgebung bereitstellt
3. Eine Komponente eines Voice over Internet Protocol, die eine Brücke zwischen VoIP und Umgebungen mit Wählverbindungen darstellt
4. Eine Einheit oder ein Programm, mit der bzw. dem Netze oder Systeme mit unterschiedlichen Netzarchitekturen miteinander verbunden werden können. Die Systeme können unterschiedliche Eigenschaften haben, z. B. unterschiedliche Kommunikationsprotokolle, unterschiedliche Netzarchitekturen oder unterschiedliche Sicherheitsrichtlinien. In diesem Fall übernimmt das Gateway sowohl eine Umsetzungs- als auch eine Verbindungsrolle.

I

Interactive System Productivity Facility (ISPF). Ein IBM Lizenzprogramm, das als Gesamtanzeigeditor und Dialogmanager eingesetzt wird. Das Programm wird für das Schreiben von Anwendungsprogrammen verwendet und ermöglicht dem Benutzer, Standardanzeigen und Dialoge zwischen dem Anwendungsprogrammierer und dem Endbenutzer zu generieren. ISPF setzt sich aus vier Hauptkomponenten zusammen: DM, PDF, SCLM und C/S. Die Komponente DM ist Dialog Manager, das die Services für Dialoge und Endbenutzer bereitstellt. Die Komponente PDF ist Program Development Facility, das Services für die Unterstützung von Dialog- und Anwendungsentwicklern bereitstellt. Die Komponente SCLM ist Software Configuration Library Manager, das Anwendungsentwicklern Services für die Verwaltung Ihrer Anwendungsentwicklungsbibliotheken bereitstellt. Die Komponente C/S ist die Client/Serverkomponente, die es Ihnen ermöglicht, ISPF auf programmierbaren Workstations auszuführen, um die Anzeigen mit der Anzeigefunktion des Workstationbetriebssystems anzuzeigen und Workstation-Tools und -daten in Host-Tools und -daten zu integrieren.

Interpreter. Ein Programm, das jede Instruktion einer höheren Programmiersprache übersetzt und ausführt, bevor es die nächste Instruktion übersetzt und ausführt

Isomorph. Jedes zusammengesetzte Element (in anderen Worten jedes Element, das weitere Elemente enthält) des XML-Instanzdokuments hat ausgehend vom Stammverzeichnis genau ein entsprechendes COBOL-Gruppenelement, dessen Verschachtelungstiefe mit der Verschachtelungstiefe seines XML-Äquivalents identisch ist. Jedes nicht zusammengesetzte Element (in anderen Worten jedes Element, das keine weiteren Elemente enthält) im XML-Instanzdokument hat ausgehend vom Stamm genau ein entsprechendes Datenelement, dessen Verschachtelungstiefe mit der Verschachtelungstiefe seines XML-Äquivalents identisch ist und dessen Speicheradresse zur Laufzeit eindeutig identifiziert werden kann.

K

Kompilieren.

1. In ILE-Sprachen (Integrated Language Environment) das Umsetzen von Quellenanweisungen in Module, die anschließend in Programme oder Serviceprogramme eingebunden werden können
2. Das Umsetzen eines vollständigen Programms oder von Teilen eines Programms, das in einer höheren Programmiersprache geschrieben ist, in ein Computerprogramm in IL, Assemblersprache oder Maschinensprache

L

Ladebibliothek. Eine Bibliothek mit Lademodulen

LINKAGE SECTION. Der Abschnitt im Datenteil einer aktivierten Einheit (einem aufgerufenen Programm oder einer aufgerufenen Methode), der Datenelemente beschreibt, die von der aktivierten Einheit (Programm oder Methode) zur Verfügung gestellt werden. Die aktivierte Einheit und die aktivierende Einheit können auf diese Datenelemente verweisen.

N

Navigatoransicht. Eine hierarchische Sicht der Ressourcen in der Workbench

Nicht isomorph. Eine einfache Zuordnung von COBOL-Elementen und XML-Elementen, die zu XML-Dokumenten und COBOL-Gruppen gehören, die keine identische Form haben (nicht isomorph sind). Eine nicht isomorphe Zuordnung kann auch zwischen nicht isomorphen Elementen isomorpher Strukturen erstellt werden.

P

Perspektive. Eine Gruppe von Sichten, die verschiedene Aspekte der Ressourcen in der Workbench zeigen. Der Workbench-Benutzer kann - je nach auszuführender Task - die Perspektive wechseln und auch das Layout der Sichten und Editoren innerhalb einer Perspektive anpassen.

Perspektive für ferne Systeme. Eine Schnittstelle für die Verwaltung ferner Systeme unter Einhaltung von Konventionen, die ISPF ähnlich sind

R

RAM. Repository Access Manager

Repository.

1. Ein Speicherbereich für Daten. Jedes Repository hat einen Namen und einen zugehörigen Geschäftselementtyp. Standardmäßig ist der Repositoryname identisch mit dem Namen des Geschäftselements. Beispielsweise hat ein Repository für Rechnungen den Namen 'Rechnungen'. Es gibt zwei Typen von Informationsrepositories: lokale (prozessspezifische) und globale (wiederverwendbare) Repositories.
2. Eine VSAM-Datei, in der die Status von BTS-Prozessen gespeichert werden. Wenn ein Prozess nicht unter der Steuerung von BTS ausgeführt wird, werden der Prozessstatus (und die Status der zugehörigen Aktivitäten) erhalten, indem sie in eine Repository-Datei geschrieben werden. Die Status aller Prozesse eines bestimmten Prozesstyps (und der zugehörigen Aktivitätsinstanzen) werden in derselben Repository-

ry-Datei gespeichert. Es können Datensätze für mehrere Prozesstypen in dasselbe Repository geschrieben werden.

3. Ein permanenter Speicherbereich für Quellcode und andere Anwendungsressourcen. In einer Teamprogrammierungsumgebung ermöglicht ein gemeinsam benutztes Repository den Zugriff mehrerer Benutzer auf Anwendungsressourcen.
4. Eine Sammlung von Informationen über die Warteschlangenmanager, die zu einem Cluster gehören. Zu diesen Informationen gehören die Namen der Warteschlangenmanager, ihre Positionen, ihre Channel, die zugehörigen Warteschlangen usw.

Repositoryinstanz. Ein Projekt oder eine Komponente, das bzw. die in einem SCM-System vorhanden ist

Repositorysicht. Zeigt die CVS-Repository-Positionen an, die Ihrer Workbench hinzugefügt wurden

S

Serveransicht. Zeigt eine Liste mit allen Servern und den zugehörigen Konfigurationen an

Shell. Eine Softwareschnittstelle zwischen Benutzern und dem Betriebssystem, die Befehle und Benutzerinteraktionen interpretiert und diese an das Betriebssystem übermittelt. Ein Computer kann mehrere Shellebenen für unterschiedliche Ebenen von Benutzerinteraktionen haben.

Shellname. Der Name der Shellschnittstelle

Shell-Script. Eine Datei mit Befehlen, die von der Shell interpretiert werden können. Der Benutzer gibt den Namen der Scriptdatei an der Shelleingabeaufforderung ein und veranlasst die Shell damit, die Scriptbefehle auszuführen.

Sidedeck. Eine Bibliothek, in der die Funktionen eines DLL-Programms veröffentlicht werden. Die Eintrags- und Modulnamen werden nach der Kompilierung des Quellcodes in der Bibliothek gespeichert.

Sperraktion. Sperrt ein Member

T

Taskliste. Eine Liste mit Prozeduren, die in einem Steuerungsablauf ausgeführt werden können

U

Unbeaufsichtigte Deinstallation. Ein Deinstallationsprozess, bei dem keine Nachrichten an die Konsole gesendet werden, sondern Nachrichten und Fehler nach dem Aufruf des Deinstallationsbefehls in Protokolldateien gespeichert werden

Unbeaufsichtigte Installation. Eine Installation, bei der keine Nachrichten an die Konsole gesendet, sondern Nachrichten und Fehler in Protokolldateien gespeichert werden. Bei einer unbeaufsichtigten Installation können Antwortdateien für die Dateneingabe verwendet werden.

URL. Uniform Resource Locator

Dokumentationshinweise für IBM Rational Developer for System z

© Copyright IBM Corporation - 2010

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekannt gegeben. 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 dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

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

Intellectual Property Dept. for Rational Software
IBM Corporation
3039 Cornwallis Road, PO Box 12195
Research Triangle Park, NC 27709
USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen 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 Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

Copyrightlizenz

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben wurden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. IBM kann deshalb nicht garantieren, dass die Zuverlässigkeit, Wartungsfreundlichkeit und Funktion dieser Programme gegeben ist. Die Beispielprogramme werden auf der Grundlage des gegenwärtigen Zustands (auf "as-is"-Basis) und ohne 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 sind Marken oder eingetragene Marken der International Business Machines Corp. Weitere Produkt- und Servicenamen können Marken von IBM oder von anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie im Web unter www.ibm.com/legal/copytrade.shtml.

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

Intel[®], das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium[®] sind Marken oder eingetragene Marken der Intel Corporation oder deren Tochtergesellschaften in den USA oder anderen Ländern.

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

Java und alle Java-basierten 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.

Index

Sonderzeichen

.dstoreMemLogging 138
.dstoreTrace 138
/etc/inittab, z/OS UNIX-Initialisierung,
INETD 325
/etc/rc, z/OS UNIX, INETD-Start 325
_RSE_CMDSERV_OPTS, Definieren zu-
sätzlicher Java-Startparameter 47
_RSE_JVAOPTS, Definieren zusätzlicher
Java-Startparameter 42
_RSE_PORTRANGE 161

A

Abfrage einer Zertifikatswiderrufsliste
(CRL)
 CRL-Umgebungsvariablen 173
 rsed.envvars 173
Abhängigkeit vom Hostnamen 313
ADM anpassen 73
Administrator, SCLMDT 87
ADNCSDAR, nicht primäre RESTful- und
Web-Service-Schnittstelle 76
ADNCSDRS, primäre RESTful-Schnittstel-
le 75
ADNCSDTX, RESTful-Änderungs-ID 76
ADNCSDWS, primärer Web-Service 78
ADNJSPAU, CICS-Verwaltungsdienstpro-
gramm 75
ADNJSPAU, Verwaltungsdienstpro-
gramm 261
ADNMSGHC, Pipelinenachrichtenhand-
ler 77
ADNMSGHS, Pipelinenachrichtenhand-
ler 77
ADNTXNC, RESTful-Änderungs-ID 76
ADNVCRD, CRD-Repository 74
ADNVMFST, Manifestrepository 79
Adressraum, Größe 153
Adressräume, Begrenzung für die Grö-
ße 227
Aktionen für Beschränkungen der Jobaus-
führung 168
Aktivieren der Beispiel-RAM (Repository
Access Managers) 62
Aktivieren des PDS-Repository-Access-
Managers 63
Aktivieren des SCLM-Repository-Access-
Managers 63
Aktivieren des Skeleton-RAM (Repository
Access Manager) 63
Aktivieren von Common Access Reposi-
tory Manager 51
Aktualisierungsberechtigungen für Benut-
zer ohne Systemadministratorrech-
te 203
Alternative Konfigurationsoptionen,
APPC und VTAM 334
Alternative RSE-Verbindungsmetho-
de 103
Alternativer CARMA-Serverstart 60
Alternativer Name für Partner-LU, TSO
Commands Service 334
Änderungen an DB2 90
Änderungen an PROCLIB 90
Änderungen an Workload Manager 89
Änderungen zwischen Version 7.0 und
Version 7.1 295
Anforderungen an die Start-JCL 153
Anpassen von SCLM Developer Tool-
kit 81
Anpassung, Basis 15
Anpassung der CRD-Servertransaktions-
IDs 76
Anpassung der TSO-Umgebung 269
Anpassung von Application Deployment
Manager 257
Anpassung von CRASERV.properties 60
Anpassung von ISPF.conf 61, 270
Anpassungskonfiguration 15
Anpassungstasks, optional 89
Ant
 zusätzlich erforderliche Produkte 347
Ant installieren und anpassen 86
Anwendungsentwicklung 250
Anwendungsschutz für RSE definie-
ren 188
Anzahl der Adressräume 217
Anzahl der Prozesse 220
Anzahl der Threads 223
Apache Ant
 zusätzlich erforderliche Produkte 347
APF-Autorisierung 150
 FEK.SFEKAUTH 181
APPC, alternative Konfigurationsoptio-
nen 334
APPC, Hinweise zur Verwendung 107
APPC als Zugriffsmethode verwen-
den 273
APPC-Änderungen aktivieren 333
APPC konfigurieren 329
APPC-Transaktion
 Fehlerbehebung für APPC 155
APPC-Transaktion, Implementie-
rung 106
APPC-Transaktion, Vorbereitungen 105
APPC-Transaktion für TSO Commands
Service 104
APPC-Transaktion und TSO Commands
Service 155
APPC-Transaktionen mit mehreren Konfi-
gurationen von Developer for System
z 274
APPC-Transaktions-JCL, Basisanpas-
sung 273
APPC-Transaktions-JCL, Zuordnungs-
Exec verwenden 274
APPC-Transaktions-JCL für vorhandene
ISPF-Profile verwenden 273
APPC-Transaktionsprotokollierung 143
APPCPMxx 331

Application Deployment Manager
(ADM) 193
Application Deployment Manager, CICS
Resource Definition-Editor 257
Application Deployment Manager, CICS
Resource Definition-Server 257
Application Deployment Manager, Sicher-
heit 259
Application Deployment Manager anpas-
sen 73, 257
ASCHPMxx 332
 MAX 239
Aspekte der Leistung 249
Aspekte der Sicherheit 159
ASSIZEMAX 184
audit.log 139
Ausführung mehrerer Instanzen 277
Auslastungsverwaltung 251
Authentifizierung durch JES Job Moni-
tor 160
Authentifizierung durch RSE-Dä-
mon 174
Authentifizierung durch Sicherheitssoft-
ware 173
Authentifizierung konfigurieren, SSL und
X.509 299
Authentifizierungsmethoden 159

B

Basisanpassung 15
Basiskonfigurationsdateien des Resol-
vers 315
Batchübergabe, CA Endevor® SCM-RAM-
Start 65
Batchübergabe, CARMA-Server star-
ten 56
Bedienerbefehle 125
Bedingte Aktionen für Jobs 166
Bedingter Zugriff auf Spooldateien 170
Befehl 'Modify', JES Job Monitor 127
Befehl 'Modify', RSE-Dämon 128
Befehl 'Start', JES Job Monitor 125
Befehl 'Start', Sperrdämon 126
Befehl Start, RSE-Dämon 126
Befehle, Modify (F) 127
Befehle, Sperrdämon Modify 131
Befehle, Start (S) 125
Befehle, Stop (P) 131
Befehle des Bedieners 125
Befehlssicherheit definieren, JES 185
Begrenzung für die Größe der Adressräu-
me 227
Begrenzung für die Größe des Java-Heap-
speichers 226
Beispiel-RAM (Repository Access Mana-
gers) aktivieren 62
Beispielanalyse der Speicherbele-
gung 228
Beispielkonfiguration 244
 Anzahl der Thread-Pools 245

- Beispielkonfiguration (*Forts.*)
 - Grenzwerte definieren 246
 - minimale Grenzwerte bestimmen 245
- Benutzer, RSE-Protokollierung 140
- Benutzer-ID, Hinweise 9
- Benutzer-ID und Kennwort 160
- Benutzer-ID und Kennwort für einmaliges Anmelden 160
- Benutzer ohne Systemadministratorrechte, Aktualisierungsberechtigungen 203
- Berechtigungsbits, z/OS UNIX 149
- Bereits konfigurierte Dateien, Sicherung
 - Version 7.0 283
 - Version 7.1 283
 - Version 7.5 283
- Beschränkung der externen Kommunikation auf angegebene Ports 161
- Beschränkungen der Ausführung, Aktionen für Jobs 168
- Bidirektionale Sprachen, Unterstützung 92
- BPXBATCH, INETD-Start 326
- BPXPRMxx 246
 - INADDRANYCOUNT 238
 - MAXASSIZE 154, 184, 237
 - MAXFILEPROC 237
 - MAXMMAPAREA 237
 - MAXPROCSYS 156, 236
 - MAXPROCUSER 156, 236
 - MAXSOCKETS 238
 - MAXTHREADS 236
 - MAXTHREADTASKS 236
 - MAXUIDS 157, 237
- BPXPRMxx, Grenzwerte festlegen
 - MAXASSIZE 16
 - MAXPROCUSER 16
 - MAXTHREADS 16
 - MAXTHREADTASKS 16

C

- CA Endeavor® SCM-RAM 64
- CA Endeavor® SCM-RAM anpassen 70
- CA Endeavor® SCM-RAM-Start, Batchübergabe 65
- CA Endeavor® SCM-RAM-Start, CRAFT 68
- CA Endeavor® Software Configuration Manager-Repository-Access-Manager aktivieren 64
- Cachegrößenbegrenzung, Java Virtual Machines (JVMs) 253
- Cachesicherheit, Java Virtual Machines (JVMs) 253
- CARMA, FMID HCMA710 296
- CARMA, RSE-Schnittstelle 53
- CARMA, Traceerstellung 147
- CARMA aktivieren 51
- CARMA-Protokollierung
 - rsecomm.log 142
- CARMA-Serverstart, alternativ 52, 57, 60
- CARMA-Start, Batchübergabe 56
- CARMA und TCP/IP-Ports 164
- CEE.SCEELPA
 - SYS1.PARMLIB(LPALSTxx) 250

- CICS, nicht primäre Verbindungsregion 79
- CICS, nicht primäre Verbindungsregionen 76
- CICS, primäre Verbindungsregion 75, 78
- CICS, Unterstützung bidirektionaler Sprachen 92
- CICS Resource Definition-Editor (CRD-Editor), Application Deployment Manager 257
- CICS Resource Definition-Server (CRD-Server), Application Deployment Manager 257
- CICS-Ressourcendefinitionen, Administrator 257
- CICS-Ressourcendefinitionen, Entwickler 257
- CICS-Ressourceninstallation protokollieren 259
- CICS Transaction Server
 - zusätzlich erforderliche Produkte 344
- CICS-Transaktionen 176
- CICS-Verbindungsregionen, nicht primäre 76
- CICS-Verwaltungsdienstprogramm 75
- CICSplex SM Business Application Services (BAS) 258
- CICSTS-Aspekte 257
- CICSTS-Sicherheit 176
- CLASSPATH 279
- Client, Prüfliste 13
- Clientauthentifizierung unter Verwendung von X.509-Zertifikaten 171
- COBOL
 - ferne Prüfung 148
- COBOL-Compiler
 - zusätzlich erforderliche Produkte 342
- COMMNDxx, gestartete Tasks hinzufügen 18
- Common Access Repository Manager (CARMA), FMID HCMA710 296
- Common Access Repository Manager, Protokollierung 142
- Common Access Repository Manager aktivieren 51
- CRA#ASLM, Beispiel-SCLM-RAM 63
- CRA#CIRX, IRXJCL oder CRAXJCL 72
- CRA#CRAM, Beispiel-Skeleton-RAM 63
- CRA#UADD, CARMA-Komponenten 52, 65
- CRA#UQRY, CARMA-Komponenten 52, 65
- CRA#VCAD, Endeavor SCM-RAM 65
- CRA#VCAS, Endeavor SCM-RAM 65
- CRA\$VDEF, CARMA-Komponenten 52
- CRA\$VMSG, CARMA-Komponenten 52, 65
- CRA#VPDS, Beispiel-PDS-RAM 63
- CRA#VSLM, Beispiel-SCLM-RAM 63
- CRA\$VSTR, CARMA-Komponenten 52
- CRAISPRX, ISPF für alternativen CARMA-Start 61
- CRAISPRX, ISPF-Konfigurationsdatei 48
- CRANDVRA anpassen 69
- CRASERV.properties anpassen 60
- CRASRV.properties
 - clist.dsname 54

- CRASRV.properties (*Forts.*)
 - crastart.configuration.file 54
 - crastart.steplib 54
 - crastart.stub 54
 - crastart.syslog 54
 - crastart.tasklib 54
 - crastart.timeout 54
 - port.range 54
 - port.start 54
 - startup.script.name 54
- CRASRV.properties anpassen 56, 58, 66, 68
- CRASTART, alternativer CARMA-Start 57
- CRASTART, CA Endeavor® SCM-RAM-Start 68
- crastart.conf, CARMA-Start mit CRAFT 58
- crastart.conf anpassen 58
- crastart.endeavor.conf anpassen 68
- CRASUBCA anpassen 66
- CRASUBMT, CARMA-Batchstart 56
- CRAXJCL, IRXJCL oder CRAXJCL 72
- CRD-Repository 74, 176
- CRD-Server mit der RESTful-Schnittstelle 75
- CRD-Servertransaktions-IDs anpassen 76
- cron, WORKAREA-Verzeichnisbereinigung 108

D

- Dämon, RSE 23
- Dämon, Sperren 24
- Dateien können nicht geöffnet werden, MVS 157
- Dateiprofile definieren 181
- Dateisystem-Speicherbelegung, z/OS UNIX 232
- Dateisystemattribut SETUID 149
- Dateisysteme, zFS 249
- DB2
 - zusätzlich erforderliche Produkte 345
- DB2, gespeicherte Prozedur 89
- DB2-Änderungen 90
- Debug Tool
 - zusätzlich erforderliche Produkte 343
- Debug Tool (IBM), zusätzlich erforderliche Produkte 20
- Definitionen, Sicherheit 28, 178
- Definitionen für den Resolver 319
- Definitionen von verschiedenen Ressourcen 238
 - EXEC-Karte, Server-JCL 238
 - FEJJCNFG 239
 - SYS1.PARMLIB(ASCHPMxx) 239
 - SYS1.PARMLIB(IEASYSxx) 239
 - SYS1.PARMLIB(IVTPRMxx) 239
- Deploymentvorbereitungen, Hinweise 12
- Developer for System z, gestartete Tasks definieren 184
- Developer for System z, Komponentenübersicht
 - grafische Darstellung 193

Developer for System z, Wissenswertes 193
Dienstprogramme für Cacheverwaltung, Java Virtual Machines (JVMs) 254
Durchsatz der Sicherheitsprüfung verbessern 251

E

Eigenschaftsgruppe, hostbasiert 98
Eingangsport überprüfen 162, 175
Einstellungen und Klassen für Sicherheit aktivieren 180
ELAXF*-Prozeduren, Beispiel
 ELAXFADT 26
 ELAXFASM 26
 ELAXFBMS 26
 ELAXFCOC 26
 ELAXFCOP 26
 ELAXFCOT 26
 ELAXFCP1 26
 ELAXFCPC 26
 ELAXFCPP 26
 ELAXFDCL 26
 ELAXFGO 26
 ELAXFLNK 26
 ELAXFMFS 26
 ELAXFPL1 26
 ELAXFPLP 26
 ELAXFPLT 26
 ELAXFPP1 26
 ELAXFTSO 26
 ELAXFUOP 26
ELAXF*-Prozeduren für ferne Builderstellung 26
ELAXMJCL, DB2-Änderungen 90
ELAXMSAM 90
Emulator, Host-Connect 158
Enterprise Service Tools-Unterstützung 91
Entfernen alter Dateien aus WORKAREA 88
Entwicklung von Anwendungen 250
Erforderliche Ressourcen 5
Erstellung von LSTRANS.FILE 84
EST-Unterstützung 91
ETC.SERVICES
 Aliasnamen 322
 Portnummer 322
 Protokoll 322
 Servicename 322
Externe Kommunikation 163
Externe Kommunikation auf angegebene Ports beschränken 161

F

fa.log 138
Fault Analyzer
 zusätzlich erforderliche Produkte 346
Fault Analyzer-Integration, Protokollierung
 fa.log 141
 rsecomm.log 141
Fehler beim Öffnen der MVS-Dateien 157

Fehlernachrichten, IRZ-Diagnose 93
Fehlerrückmeldungen, Trace 148
FEJJCNF 163, 246, 280
 CONSOLE_NAME 168
 MAX_THREADS 239
FEJJCNF, JES Job Monitor 177
FEJJCNF (Konfigurationsdatei für JES Job Monitor) 29
FEJJJCL, PROCLIB-Änderungen, JES Job Monitor 21
FEJTSO 29
FEKAPPC, APPC-Implementierung 106
FEKAPPCL, APPC-Implementierung 106
FEKAPPCX, APPC-Implementierung 106
FEKAPPL 160
fekfivpi, Protokollierung des IVP-Tests
 fekfivpi.log 143
fekfivpi.log 138
fekfivpi.log, Protokollierung des IVP-Tests 143
fekfivps.log, Protokollierung des IVP-Tests 143
FEKFRSRV 107
FEKLOCKD 24
FEKLOGS, Protokoll- und Konfigurationsanalyse 137
FEKRACE, Sicherheitsdefinitionen 28, 178
fekrivp 150
FEKRSED 23
FEKSETUP 15, 76, 82, 106
Ferne hostbasierte Aktionen, z/OS UNIX-Unterprojekte 103
Feste Java-Heapgröße 251
ffs.log 138
ffsget.log 138
ffsput.log 138
File Manager
 zusätzlich erforderliche Produkte 346
File Manager-Integration 100
File Manager-Integration, Protokollierung
 rsecomm.log 142
FMID HCMA710 296
FMID HHOP710 295
FMID HHOP750 292
FMID HHOP760 286
FMIEXT.properties
 enabled 101
 fmlistenport 101
Fragmente, Syntax 135
Fremdanbieter und X.509-Zertifikat 160
Für den Resolver verfügbare lokale Definitionen 319

G

Gemeinsame Klassennutzung, in Java Virtual Machines (JVMs) aktivieren 253
Gemeinsame Klassennutzung aktivieren, Java Virtual Machines (JVMs) 253
Gemeinsame Klassennutzung durch mehrere Java Virtual Machines (JVMs) 252
Gespeicherte DB2-Prozedur 89
Gestartete RSE-Task, RSED 23

Gestartete Task von JES Job Monitor, JMON 21
Gestartete Tasks, Definieren für Developer for System z
 JMON, gestartete Tasks 184
 LOCKD, gestartete Tasks 184
 RSED, gestartete Tasks 184
Gestartete Tasks, Services prüfen 111
Gestartete Tasks prüfen 109
Grenzwerte des Systems 156
Größe des Adressraums 153
Größenschätzungen, Richtlinien 227
gskkyman, Schlüsseldatenbank erstellen 308

H

Handler für Pipelinenachrichten 77
HCMA710 296
Heapspeicher, Begrenzung für die Größe, Java 226
HHOP710 295
HHOP750 292
HHOP760 286
Hinweise zu den Deploymentvorbereitungen 12
Hinweise zu den Installationsvorbereitungen 4
Hinweise zu den Konfigurationsvorbereitungen 9
Hinweise zu WLM xvii, 205
Hinweise zum Server 10
Hinweise zur Benutzer-ID 9
Hinweise zur Verwendung, APPC 107
Host
 Voraussetzungen 337
Host, zusätzlich erforderliche Produkte 340
Host-Adresse, nicht aufgelöst durch TCP/IP-Resolver
 lock.log 320
Host-Connect-Emulator 158
Hostbasierte Eigenschaftsgruppe 98
Hostbasierte Projekte 99
Hostkomponenten installieren und konfigurieren 3
Hostnamen, Abhängigkeit 313
Hostnamen in Developer for System z anwenden 317
Hosttabellen, lokal 316
Hostvoraussetzungen 337
Hostvoraussetzungen für z/OS 337

I

IBM Debug Tool, zusätzlich erforderliche Produkte 20
Identische Konfiguration in einem System 277
Identische Softwareversionen mit unterschiedlichen Konfigurationsdateien 278
IEASYSxx 246
 MAXUSER 157, 239
IMS
 zusätzlich erforderliche Produkte 344

INETD, Anforderungen von Developer for System z 328

INETD, temporäre Datei "/etc/inetd.pid" 325

inetd.conf 321

- Benutzer-ID 321
- Option_wait 321
- Protokoll 321
- Serverprogramm 321
- Serverprogrammargumente 321
- Servicename 321
- Sockettyp 321

INETD konfigurieren 321

INETD-Sicherheit 327

INETD-Umgebungseinstellungen 328

Initialisierung, IVP 112

Installation protokollieren, CICS-Ressourcen 259

Installation und Konfiguration von Hostkomponenten 3

Installationsprüfprogramme 111

Installationsprüfung 109

Installationsvorbereitungen, Hinweise 4

Integration, File Manager 100

Interne Kommunikation 163

IRXJCL, IRXJCL oder CRAXJCL 72

IRZ-Diagnosefehlnachrichten 93

ISP.SISPLoad

- TSO/ISPF-Client-Gateway von ISPF 187

ISPF, mehrere Zuordnungs-Execs verwenden 271

ISPF, Verbindung mit TSO/ISPF-Client-Gateway überprüfen 116

ISPF-Befehle 48, 61

ISPF.conf 48

- allocjob 48
- ISPF_timeout 48
- isplib 48
- ispmplib 48
- isplib 48
- ispslib 48
- isptlib 48
- sysproc 48

ISPF.conf, Aktualisierungen für SCLMDT 82

ISPF.conf, Basisanpassung 270

ISPF.conf anpassen 61

ISPF.conf-Dateien mit mehreren Konfigurationen 272

ISPF-Profil (JCL für APPC-Transaktion) 273

IVP

- fekfivpa 118
- fekfivpd 115
- fekfivpi 116
- fekfivpj 115
- fekfivpl 116
- fekfivpr 120
- fekfivps 118
- fekfivpz 121

IVP, grundlegende und Zusatzservices

- fekfivpa 111
- fekfivpd 111
- fekfivpi 111
- fekfivpj 111
- fekfivpl 111

IVP, grundlegende und Zusatzservices (Forts.)

- fekfivpr 111
- fekfivps 111
- fekfivpt 111
- fekfivpz 111
- IVP-Scripts 111

IVP-Initialisierung

- ivpinit 112

IVP-Test, Protokollierung

- fekfivpi.log 143
- fekfivps.log 143

IVTPRMxx

- ECSA MAX 239
- FIXED MAX 239

J

J2EE 81, 86

Java 81, 86

Java 2 Technology Edition

- zusätzlich erforderliche Produkte 339

JAVA_DUMP_TDUMP_PATTERN 144

Java-Heapgröße, fest 251

Java-Heapspeicher, Begrenzung für die Größe 226

Java-Option 'Xquickstart' 252

Java-Speicherauszüge 144

Java-Startparameter mit _RSE_CMDSERV_OPTS definieren 47

Java-Startparameter mit _RSE_JAVAOPTS definieren 42

Java Virtual Machines (JVMs), gemeinsame Klassennutzung 252

JES-Befehlssicherheit definieren 185

JES JMON

- GEN_CONSOLE_NAME 169

JES JMON, FEJJC�FG

- _BPXK_SETIBMOPT
- _TRANSPORT 29
- APPLID 29
- AUTHMETHOD 29
- CODEPAGE 29
- CONCHAR 29
- CONSOLE_NAME 29
- GEN_CONSOLE_NAME 29
- HOST_CODEPAGE 29
- LIMIT_COMMANDS 29
- LIMIT_VIEW 29
- LISTEN_QUEUE_LENGTH 29
- MAX_DATASETS 29
- MAX_THREADS 29
- SERV_PORT 29
- SUBMITMETHOD 29
- TIMEOUT 29
- TIMEOUT_INTERVAL 29
- TSO_TEMPLATE 29
- TZ 29

JES Job Monitor, Authentifizierung 160

JES Job Monitor, Befehl 'Modify' 127

JES Job Monitor, Befehl 'Start' 125

JES Job Monitor, FEJJC�FG 177

JES Job Monitor, JMON 109

JES Job Monitor (JMON) 193

JES Job Monitor, Konfigurationsdatei FEJJC�FG 29

JES Job Monitor, Protokollierung 139

JES Job Monitor, Traceerstellung 146

JES Job Monitor-Konfiguration

- GEN_CONSOLE_NAME 169

JES Job Monitor-Server 21

JES Job Monitor-Verbindung 115

JES-Sicherheit 166

JMON 21, 185, 280

- fekfivpj 115

JMON, JES Job Monitor 109

Job Monitor-Server, JES 21

Jobs, bedingte Aktionen 166

JVMs, gemeinsame Klassennutzung 252

K

Kennwort für einmaliges Anmelden und Benutzer-ID 160

Kennwort und Benutzer-ID 160

Keystore mit keytool erstellen 311

keytool, Keystore erstellen 311

Klassifikation für Verarbeitungsprozesse, WLM 205

Klassifikationsregeln, WLM 206

Klonen der vorhandenen RSE-Konfiguration 303

Koexistenz, rsed.envvars zum Aktivieren der Koexistenz aktualisieren 303

Kommunikation, extern 163

Kommunikation, intern 163

Kommunikation, mit SSL verschlüsselt 170, 261

Kommunikation mit SSL verschlüsseln 161

Komponenten, optionale 49

Komponenten installieren und konfigurieren 3

Komponentenübersicht, Developer for System z

- grafische Darstellung 193

Konfiguration, Anpassung 15

Konfiguration, identisch in einem Systemplex 277

Konfiguration für vorausgesetzte Produkte und Software 9

Konfiguration von Hostkomponenten 3

Konfigurationsdatei für TSO/ISPF-Client-Gateway 48

Konfigurationsdateiänderungen, andere Produkte 283

Konfigurationsdateien, Developer for System z 177

Konfigurationsdateien, unterschiedliche in identischen Softwareversionen 278

Konfigurationsdateien des Resolvers 315

Konfigurationsdaten, Suchreihenfolge 314

Konfigurationsprobleme lösen 137

Konfigurationsvorbereitung, Hinweise 9

Konfigurierbare Dateien 288, 296

Konsolnachrichten 132

- JES Job Monitor 132
- RSE-Dämon 132
- RSE-Thread-Pool-Server 132
- Sperrdämon 132

L

Lange/kurze Namen umsetzen, SCLM 84
Language Environment, Laufzeitbibliotheken 250
Laufzeitbibliotheken, Language Environment 250
Leerzeichen im Syntaxdiagramm 135
Leistungsaspekte 249
Lesen eines Syntaxdiagramms 134
LIMIT_COMMANDS 167
LIMIT_VIEW 170
LINKLIST, Definitionen für andere Produkte 21
LINKLIST-Definitionen, vorausgesetzte 20
lock.log 138
LOCKD 10
LOCKD, Sperrdämon 109
Lokale Hosttabellen 316
LPA-Definitionen, vorausgesetzte 20
LPALSTxx 250
LPALSTxx, LPA-Definitionen 18
LSTRANS.FILE erstellen 84
LU-Sicherheit, VTAM 334

M

Manifestrepository 79
Mehrere APPC-Transaktionen 274
Mehrere Instanzen ausführen 277
Mehrere ISPF.conf-Dateien 272
Mehrere Konfigurationen für Developer for System z durch Verwendung mehrerer ISPF.conf-Dateien 272
Mehrere Konfigurationen von Developer for System z für APPC-Transaktionen verwenden 274
Mehrere LUs 334
Mehrere Zuordnungs-Execs, TSO/ISPF 271
Methoden zur Authentifizierung 159
Migration 283
Migration, Hinweise 3
Migration, Version 7.5 auf Version 7.6 286
Migrationshinweise 283
Migrationshinweise, Verwaltungsdienstprogramm 266
Mit SSL verschlüsselte Kommunikation 170, 176, 261
Modify (F), Befehl 127
MVS-Bibliotheken für RSE definieren 187
MVS-Dateien können nicht geöffnet werden 157
MVS-Speicherauszüge 144

N

Nachrichten, Konsole 132
Nachrichten des Verwaltungsdienstprogramms 266
Nachrichtenhandler für Pipelinenachrichten 77
Namensumsetzung, SCLM 84

netstat 152
netstat, TCP/IP konfigurieren 114
Netz überwachen 243
Nicht alphanumerische Zeichen im Syntaxdiagramm 135
Nicht editierbare Zeichen, Anpassung für Bearbeitung 101
Nicht editierbare Zeichen, uchars.settings 101
Nicht primäre Verbindungsregionen, CICS 76

O

OMVS-Segment definieren 181
Operanden im Syntaxdiagramm 134
Operanden in einem Syntaxdiagramm auswählen 135
Optimierungsaspekte 215
Optionale Anpassungstasks 89
Optionale Komponenten 49
Optionale Tasks für Anpassung 89

P

PARM-Variable, JCL-Einschränkungen 24
PARMLIB, Änderungen 16
PassTicket-Unterstützung für RSE definieren 188
PassTickets verwenden 164
PDS-Repository-Access-Manager aktivieren 63
Pipelinenachrichten, Handler 77
Pipelinesicherheit 259
PL/I-Compiler
zusätzlich erforderliche Produkte 342
Plattenspeicherplatz, Java Virtual Machines (JVMs) 254
POE-Überprüfung 162, 175
Portdefinitionen, PROFILE.TCPIP 324
Ported Tools
zusätzlich erforderliche Produkte 347
PORTRANGE 42, 152
PORTRANGE, verfügbaren für RSE definieren 41
Ports, externe Kommunikation auf angegebene Ports beschränken 161
Ports, REXEC 104
Ports, TCP/IP 162
Ports, TCP/IP und CARMA 164
Portverfügbarkeit 113
Primäre und nicht primäre Verbindungsregionen 258
Primäre Verbindungsregion, CICS 75
Private Schlüssel und Zertifikate, Speicherpositionen festlegen 300
PROCLIB-Änderungen 21, 90
Produkte, vorausgesetzte 5
Produkte und Software (vorausgesetzte) konfigurieren 9
Profile für Datei definieren 181
PROFILE.TCPIP, Portdefinitionen 324
Programmgesteuerte MVS-Bibliotheken für RSE definieren 187

Programmgesteuerte UNIX-Dateien für RSE definieren 190
Programmgesteuerte z/OS UNIX-Dateien für RSE definieren 190
Programmsteuerung autorisieren 149
PROGxx, APF-Berechtigungen 18
PROGxx, LINKLIST-Definitionen 19
projectcfg.properties 99
PROJECT-HOME 100
WSED-VERSION 100
Projekte, hostbasiert 99
propertiescfg.properties 98
DEFAULT-VALUES 99
ENABLED 99
PROPERTY-GROUP 99
RDZ-VERSION 99
Protokoll- und Konfigurationsanalyse mit FEKLOGS 137
Protokolldateien
.dstoreMemLogging 138
.dstoreTrace 138
audit.log 138
fa.log 138
fekfivpi.log 138
fekfivps.log 138
ffs.log 138
ffsget.log 138
ffsput.log 138
lock.log 138
rmt_class_loader.cache.jar 138
rsecomm.log 138
rsedaemon.log 138
rseserver.log 138
serverlogs.count 138
stderr.log 138
stdout.log 138
Protokollierung, Konfigurationsdatei rsecomm.properties 96
Protokollierung, SCLM Developer Toolkit 142
Protokollierung der APPC-Transaktionen 143
Protokollierung der Fault Analyzer-Integration 141
Protokollierung der File Manager-Integration 142
Protokollierung des IVP-Tests fekfivpi 143
Protokollierung des RSE-Benutzers 140
Protokollierung des RSE-Dämons 139
Protokollierung des Sperrdämons 139
Protokollierung des Thread-Pools 139
Protokollierung von CARMA 142
Protokollierung von JES Job Monitor 139
Protokollierungssteuerung
_RSE_HOST_CODEPAGE 165
audit.*-Optionen 165
daemon.log 165
enable.audit.log 165
Prozeduren für ferne Builderstellung, ELAXF* 26
Prüfdaten
protokollierte Aktionen 166
Prüfliste, SCLM-Administrator 87
Prüfliste, Voraussetzungen 15

- Prüfliste der übergeordneten Qualifikationsmerkmale in ELAXF* 27
- Prüfliste für APPC-Transaktionen 105
- Prüfliste für den Client 13
- Prüfliste für den SCLM-Administrator 87
- Prüfprotokollierung, vom RSE-Dämon verwaltet 165
- Prüfung der Zertifizierungsstelle
 - gskkyman 172
 - SAF-Schlüsseldatei 172
 - TRUST, HIGHTRUST 172
- Prüfung des REXEC/SSH-Shell-Scripts 121

Q

- Qualifikationsmerkmal-Prüfliste in ELAXF* 27

R

- RACF
 - Berechtigungen 182
- RACF, Schlüsseldatei erstellen 301
- Rational Team Concert for System z
 - zusätzlich erforderliche Produkte 345
- Referenzierte Veröffentlichungen 349
- Remote Execution verwenden 102
- Remote Systems Explorer 11
- Repository, CRD 74
- Repository, Manifest 79
- Repository Access Manager 64
- Repository Access Manager aktivieren, PDS 63
- Repository Access Manager aktivieren, SCLM 63
- Repository Access Manager aktivieren, Skeleton 63
- Repository Access Manager für CA Endevor® SCM aktivieren 64
- Repositorysicherheit, CRD 259
- Reservierte Ports, TCP/IP 152
- Reservierte TCP/IP-Ports 152
- Resolver, verfügbare lokale Definitionen 319
- Resolver, Wissenswertes 314
- Ressourcen, erforderliche 5
- Ressourcendefinitionen, verschiedene 238
- Ressourceninstallation protokollieren, CICS 259
- Ressourcennutzung, Überblick 216
- Ressourcennutzung optimieren 215
- Ressourcensicherheit 261
- RESTful-Schnittstelle 258
 - ADMI 76
 - ADMR 76
 - ADMS 76
- RESTful-Schnittstelle, CRD-Server 75
- RESTful-Schnittstelle oder Web-Service-Schnittstelle 75, 258
- REXEC-Dämon, Benutzer-ID-Berechtigungen bei Start durch INETD 328
- REXEC konfigurieren 104
- REXEC/SSH-Shell-Script 121

- REXEC-Verbindung überprüfen 120
- REXEC verwenden 102
- REXX
 - zusätzlich erforderliche Produkte 346
- rmt_class_loader_cache.jar 138
- RSE 11
- RSE, Anwendungsschutz definieren 188
- RSE, PassTicket-Unterstützung definieren 188
- RSE, PORTRANGE definieren 41
- RSE, programmgesteuerte MVS-Bibliotheken definieren 187
- RSE, programmgesteuerte z/OS UNIX-Dateien definieren 190
- RSE, rsed.envvars
 - _RSE_JAVAOPTS 177
- RSE, ssl.properties 178
- RSE, Tracerstellung 146
- RSE, Überprüfung des Eingangsports definieren 175
- RSE als Java-Anwendung
 - grafische Darstellung 195
- RSE als sicheren z/OS UNIX-Server definieren 187
- RSE-Benutzer, Protokollierung
 - .dstoreMemLogging 140
 - .dstoreTrace 140
 - ffs.log 140
 - ffsget.log 140
 - ffsput.log 140
 - lock.log 140
 - rmt_class_loader.cache.jar 140
 - rsecomm.log 140
 - stderr.log 140
 - stdout.log 140
- RSE-Dämon 10, 23, 163
 - Konsolnachrichten 132
- RSE-Dämon, Authentifizierung 174
- RSE-Dämon, Befehl 'Modify' 128
- RSE-Dämon, Befehl Start 126
- RSE-Dämon, Protokolldateien
 - audit.log 140
 - rsedaemon.log 140
 - rseserver.log 140
 - serverlogs.count 140
 - stderr.*.log 140
 - stdout.*.log 140
- RSE-Dämon, Protokollierung 139
- RSE-Dämon, RSED 109
- RSE-Dämon (RSED) 193
- RSE-Dämon und Prüfprotokollierung 165
- RSE-Dämonverbindung 115
- RSE-Konfiguration klonen 303
- RSE-Konfigurationsdatei, rsed.envvars 33
- RSE-Schnittstelle zu CARMA 53
- RSE-Server 163
- RSE-Server als sicheren z/OS UNIX-Server definieren 187
- RSE-SSL-Verschlüsselung, ssl.properties
 - Dämmeigenschaften 93
 - Serveigenschaften 93
- RSE-Thread-Pool, Protokolldateien
 - audit.log 140
 - rsedaemon.log 140
 - rseserver.log 140

- RSE-Thread-Pool, Protokolldateien (Forts.)
 - serverlogs.count 140
 - stderr.*.log 140
 - stdout.*.log 140
- RSE-Thread-Pool-Server
 - Konsolnachrichten 132
- RSE-Tracekonfiguration, rsecomm.properties 96
- RSE überwachen 240
- RSE-Verbindungsmethode, alternativ 103
- rsecomm.log 138
 - File Manager-Integration, Protokollierung 142
 - SCLM Developer Toolkit, Protokollierung 142
- rsecomm.properties 96, 147
 - debug_level 96
 - log_location 96
 - server.version 96
- RSED 23
- RSED, RSE-Dämon 109
- rsed.envvars 128, 234, 279
 - _BPX_SHAREAS 40
 - _BPX_SPAWN_SCRIPT 40
 - _BPXK_SETIBMOPT
 - _TRANSPORT 38
 - _CEE_DMPTARG 36
 - _CEE_RUNOPTS 40
 - _CMDSERV_BASE_HOME 37
 - _CMDSERV_CONF_HOME 37, 272
 - _CMDSERV_WORK_HOME 37
 - _FEKFSCMD_PARTNER_LU_ 38
 - _FEKFSCMD_TP_NAME_ 38
 - &ISPROF=&SYSUID..ISPROF= 48
 - _RSE_CMDSERV_OPTS 40
 - _RSE_DAEMON_CLASS 40
 - _RSE_HOST_CODEPAGE 36
 - _RSE_JAVAOPTS 40, 144, 269
 - _RSE_LOCKD_CLASS 40
 - _RSE_LOCKD_PORT 36
 - _RSE_POOL_SERVER_CLASS 40
 - _RSE_PORTRANGE 38, 161
 - _RSE_SERVER_CLASS 40
 - _RSE_SERVER_TIMEOUT 40
 - _SCLMDT_TRANSTABLE 38
 - ANT_HOME 38
 - CGI_DTWORK 40
 - CLASSPATH 40
 - DAPPLID 46
 - Daudit.cycle 44
 - Daudit.retention.period 44
 - Ddaemon.log 42
 - Ddeny.nonzero.port 46
 - DDENY_PASSWORD 46
 - DDSTORE_IDLE_SHUTDOWN_TIMEOUT 46
 - DDSTORE_LOG_DIRECTORY 42
 - DDSTORE_MEMLOGGING_ON 46
 - DDSTORE_TRACING_ON 46
 - Denable.audit.log 44
 - Denable.automount 44
 - Denable.certificate.mapping 44
 - Denable.port.of.entry 44
 - Denable.standard.log 44
 - DHIDE_ZOS_UNIX 46

rsed.envvars (Forts.)
 Dipv6 44
 Dkeep.last.log 44
 Dmaximum.clients 42, 235
 Dmaximum.threadpool.process 44, 235
 Dmaximum.threads 42, 235
 Dminimum.threadpool.process 42, 235
 Dprocess.cleanup.interval 46
 Dsingle.logon 46
 DSTORE_LOG_DIRECTORY 142, 146
 DTSO_SERVER 46
 Duser.log 42
 GSK_CRL_SECURITY_LEVEL 38
 GSK_LDAP_PASSWORD 38
 GSK_LDAP_PORT 38
 GSK_LDAP_SERVER 38
 GSK_LDAP_USER 38
 JAVA_HOME 36
 JAVA_PROPAGATE 40
 LANG 36
 LIBPATH 40
 PATH 36, 40
 RSE_HOME 36
 RSE_JAVAOPTS 37
 RSE_LIB 40
 RSE_SAF_CLASS 37
 SCLMDT_BASE_HOME 40
 SCLMDT_CONF_HOME 38
 SCLMDT_WORK_HOME 40
 STEPLIB 36, 37, 38, 171
 TZ 36
 Xms 42, 235
 Xmx 42, 235
 rsed.envvars, Aktualisierung für die Umsetzung langer/kurzer Namen 86
 rsed.envvars, Aktualisierungen für SCLMDT 83
 rsed.envvars, RSE-Konfigurationsdatei 33
 rsed.envvars, zum Aktivieren der Koexistenz aktualisieren 303
 rsedaemon.log 138, 139
 rserver.log 138, 139

S

Schlüsseldatei mit RACF erstellen 301
 Schlüsseldatenbank mit gskkyman erstellen 308
 Schnellstart, Java-Option (-Xquickstart) 252
 Schnittstelle, RESTful oder Web-Service 75
 SCLM, Umsetzung langer/kurzer Namen 84
 SCLM-Administrator 86
 SCLM-Aktualisierungen für SCLMDT 87
 SCLM Developer Toolkit 188
 SCLM Developer Toolkit, Protokollierung rsecomm.log 142
 SCLM Developer Toolkit (SCLMDT) 193
 SCLM Developer Toolkit anpassen 81
 SCLM Developer Toolkit-Service 19

SCLM Developer Toolkit-Verbindung überprüfen
 SCLMDT-Prüfungen 118
 SCLM-Repository-Access-Manager aktivieren 63
 SCLM-Sicherheit 176
 SCLMDT, Aktualisierungen in ISPF-.conf 82
 SCLMDT, Aktualisierungen in rsed.envvars 83
 SCLMDT-Administrator 87
 SCLMDT-Verbindung überprüfen 118
 SDSF 82
 Secure Shell verwenden 102
 Secure Sockets Layer, Verbindung in der Hostkonfiguration testen 305
 Secure Sockets Layer konfigurieren 299
 Secure Sockets Layer zur Verschlüsselung der Kommunikation 161
 Segment definieren, OMVS 181
 Server, Hinweise 10
 serverlogs.count 138
 SETUID, Dateisystemattribut 149
 Shell-Script 'ivpinit', RSE-Umgebungsvariablen festlegen 112
 Shell-Script, REXEC/SSH 121
 Shellsitzung, INETD-Start 326
 Sichere APPC-Konfiguration, VTAM 334
 Sicherer z/OS UNIX-Server, RSE definieren 187
 Sicherheit, CICSTS 176
 Sicherheit, INETD 327
 Sicherheit, JES 166
 Sicherheit, Pipeline 259
 Sicherheit, Ressourcen 261
 Sicherheit, SCLM 176
 Sicherheit, Transaktionen 259
 Sicherheit des CRD-Repositorys 259
 Sicherheit für JES-Befehle definieren 185
 Sicherheit für Verbindungen 161
 Sicherheit von Application Deployment Manager (ADM) 259
 Sicherheitsaspekte 159
 Sicherheitsdefinitionen 28, 178
 Sicherheitsdefinitionen, Prüfliste 179
 Sicherheitseinstellungen prüfen 190
 Sicherheitseinstellungen und -klassen aktivieren 180
 Sicherheitsprofil mit gespeicherten Begrenzungen 154
 Sicherheitsprüfung, Durchsatz verbessern 251
 Sicherheitssoftware, Authentifizierung 173
 Sichern von konfigurierten Dateien
 Version 7.0 283
 Version 7.1 283
 Version 7.5 283
 Signiertes Zertifikat, selbstsigniert oder durch Zertifizierungsstelle signiert 302
 Skeleton-RAM (Repository Access Manager) aktivieren 63
 skulker, WORKAREA-Verzeichnisbereinigung 108
 SMP/E
 Voraussetzungen 339
 SMP/E-Installation, Sticky Bit 151

Software, vorausgesetzte Produkte konfigurieren 9
 Software Configuration Manager 64
 Softwareversionen, identische mit unterschiedlichen Konfigurationsdateien 278
 Speicherauszüge, Java 144
 Speicherauszüge, MVS 144
 Speicherauszüge, Position in z/OS UNIX 146
 Speicherauszugsdateien 144
 Speicherbelegung 226
 Speicherbelegung, Beispielanalyse 228
 Speicherbelegung, z/OS UNIX-Dateisystem 232
 Speicherpositionen für private Schlüssel und Zertifikate 300
 Sperrdämon 24, 200
 Konsolnachrichten 132
 Sperrdämon, Befehl 'Modify' 131
 Sperrdämon, Befehl 'Start' 126
 Sperrdämon, LOCKD 109
 Sperrdämon (LOCKD) 193
 Sperrdämon, Protokollierung 139
 Sperrdämon, Traceerstellung 147
 Sperrdämonflow
 grafische Darstellung 200
 Sperrdämonverbindung 116
 Sperren aufheben
 RSE, Befehl 'modify cancel' 201
 Spooldateien, bedingter Zugriff 170
 SSH-Dämon, Benutzer-ID-Berechtigungen bei Start durch INETD 328
 SSH konfigurieren 104
 SSH-Shell-Script 121
 SSH verwenden 102
 SSL, Verbindung in der Hostkonfiguration testen 305
 SSL konfigurieren 299
 ssl.properties 93
 daemon_key_label 95
 daemon_keydb_file 95
 daemon_keydb_password 95
 enable_ssl 95
 server_keystore_file 95
 server_keystore_label 95
 server_keystore_password 95
 server_keystore_type 95
 ssl.properties, SSL durch Erstellung eines neuen RSE-Dämons aktivieren 304
 ssl.properties für SSL-Aktualisierung aktivieren 303
 SSL-Verschlüsselung der Kommunikation 161
 Start, alternative Methode für CARMA-Server 60
 Start (S), Befehl 125
 Start, z/OS UNIX, INETD 325
 Start-JCL, Anforderungen 153
 Startparameter mit _RSE_CMDSERV_OPTS, Java zusätzlich definieren 47
 Startparameter mit _RSE_JAVAOPTS, Java zusätzlich definieren 42
 stderr.log 138
 stderr.log 138
 stdout.log 138
 stdout.log 138

STEPLIB, Verwendung 21
 STEPLIB, Verwendung vermeiden 249
 Sticky Bit, Verfügbarkeit des MVS-Lademoduls für z/OS UNIX 151
 Stop (P), Befehl 131
 Subsystemtypen
 ASCH 206
 CICS 206
 JES 206
 OMVS 206
 STC 206
 Suchreihenfolge, native MVS-Umgebung 323
 Suchreihenfolge, z/OS UNIX-Umgebung 323
 Suchreihenfolge für Konfigurationsdaten 314
 Suchreihenfolge in der z/OS-UNIX-Umgebung 315
 Symbole im Syntaxdiagramm 134
 Syntaxbeispiel 135
 Syntaxdiagramm, Leerzeichen 135
 Syntaxdiagramm, mehrere Operanden auswählen 135
 Syntaxdiagramm, nicht alphanumerische Zeichen und Leerzeichen 135
 Syntaxdiagramm, Operanden 134
 Syntaxdiagramm, Symbole 134
 Syntaxdiagramm lesen 134
 Syntaxdiagramm mit mehreren Zeichen 135
 Syntaxfragmente 135
 SYS1.PARMLIB(APPCPMxx) 331
 SYS1.PARMLIB(ASCHPMxx) 332
 SYS1.PARMLIB(BPXPRMxx) 246
 MAXASSIZE 154, 184
 MAXPROCSYS 156
 MAXPROCUSER 156
 MAXUIDS 157
 SYS1.PARMLIB(BPXPRMxx), Java Virtual Machines (JVMs) 254
 SYS1.PARMLIB(BPXPRMxx) mit festgelegten Begrenzungen 154
 SYS1.PARMLIB(IEASYsxx) 246
 MAXUSER 157
 SYSEXEC 48
 Sysplex, identische Konfiguration 277
 SYSPROC 48
 Systembibliotheken, Zugriff verbessern 249
 Systemexits, erzwungene Begrenzungen 154
 Systemgrenzwerte 156

T

Tabellen für Umsetzung 316
 Taskeigner 196
 TCP/IP, für den Resolver verfügbare lokale Definitionen 319
 TCP/IP in Developer for System z anwenden 317
 TCP/IP konfigurieren 313
 TCP/IP konfigurieren, netstat 114
 TCP/IP-Ports 162
 TCP/IP-Ports, grafische Darstellung 162
 TCP/IP-Ports, reservierte 152

TCP/IP-Resolver, Host-Adresse nicht aufgelöst
 lock.log 320
 Test, Protokollierung von fekvipi 143
 Testen der SSL-Verbindung in der Hostkonfiguration 305
 Thread-Pool, Protokollierung 139
 Threadsicherheit im RSE-Server
 PassTickets 164
 TN3270 342
 Trace für Fehlerrückmeldungen 148
 Traceerstellung 146
 Traceerstellung für CARMA 147
 Traceerstellung für JES Job Monitor 146
 Traceerstellung für RSE 96, 146
 Traceerstellung für Sperrdämon 147
 Tracekonfiguration, rsecomm.properties 96
 Transaktionsname, alternativ zu "FEK-FRSRV" für TSO Commands Service 334
 Transaktionsprüfliste, APPC 105
 Transaktionssicherheit 259
 Transaktionsspeicherauszug, Strukturvariablen 144
 TSO-Befehle 61
 TSO Commands Service 155, 193, 269
 TSO Commands Service, APPC-Transaktion 104
 TSO Commands Service, Protokollierung 143
 TSO Commands Service, Transaktion definieren 333
 TSO Commands Service, Verbindung mit APPC 118
 TSO/ISPF, mehrere Zuordnungs-Execs verwenden 271
 TSO/ISPF, vorhandene ISPF-Profile verwenden 270
 TSO/ISPF, Zuordnungs-Exec verwenden 271
 TSO/ISPF-Anpassung, ISPF.conf 270
 TSO/ISPF-Client-Gateway, Konfigurationsdatei 48
 TSO/ISPF-Client-Gateway als Zugriffsmethode verwenden 270
 TSO/ISPF-Client-Gateway von ISPF ISPSISPLoad 187
 TSO/ISPF mit mehreren Konfigurationen verwenden 272
 TSO-Umgebung anpassen 269
 TSO-Zugriffsmethoden 269

U

Überprüfung des Eingangsports für RSE definieren 175
 Überwachung, Netz 243
 Überwachung, RSE 240
 Überwachung, z/OS UNIX 241
 Überwachung, z/OS UNIX-Dateisysteme 244
 uchars.settings, nicht editierbare Zeichen 101
 Umgebungseinstellungen von IN-ETD 328
 Umsetztabelle 316

Umsetzung langer/kurzer Namen, Aktualisierung von rsed.envvars 86
 UNIX-Server, RSE definieren 187
 UNIX-Speicherauszüge, Position 146
 UNIX-Umgebung, Suchreihenfolge 315
 Unterschiedliche Konfigurationsdateien in identischen Softwareversionen 278
 Unterstützte Anwendungen 4
 Unterstützte Betriebssysteme 3
 Unterstützte Plattformen 3
 Unterstützte Subsysteme 4
 Unterstützung, Enterprise Service Tools 91
 Unterstützung, EST 91
 Unterstützung der Clientauthentifizierung hinzufügen, X.509 307
 Unterstützung für RSE, PassTicket definieren 188

V

Verbesserung des Durchsatzes von Sicherheitsprüfungen 251
 Verbesserung des Zugriffs auf Systembibliotheken 249
 Verbindung, JES Job Monitor 115
 Verbindung, REXEC 120
 Verbindung, RSE-Dämon 115
 Verbindung, Sperrdämon 116
 Verbindung mit TSO/ISPF-Client-Gateway von ISPF überprüfen 116
 Verbindung verweigert 156
 Verbindung zu TSO Commands Service mit APPC 118
 Verbindungsflow 198
 grafische Darstellung 198
 Verbindungsregionen, primäre und nicht primäre 258
 Verbindungssicherheit 161
 Verfügbarkeit von Ports 113
 Veröffentlichungen, referenzierte 349
 Verschlüsselte Kommunikation, mit SSL 170, 176, 261
 Verschlüsselung, ssl.properties 93
 Verschlüsselung der Kommunikation mit SSL 161
 Version 7.0 und Version 7.1, Änderungen 295
 Verwaltung der Auslastung 251
 Verwaltungsdienstprogramm, Migrationshinweise 266
 Verwaltungsdienstprogramm, Nachrichten 266
 Verwaltungsdienstprogramm für CICS-Administratoren
 bereitgestellte Funktionen 261
 Verweigte Verbindung 156
 Verwendung der Batchübergabe für CARMA-Serverstart 56
 Verwendung einer Zuordnungs-Exec 271
 Verwendung von PassTickets 164
 Verwendung von STEPLIB vermeiden 249
 Verwendung vorhandener ISPF-Profile 270

- Verzeichnisbereinigung, WORKAREA skulker 108
- Verzeichnisstruktur, z/OS UNIX grafische Darstellung 202
- Virtual Telecommunications Access Method 330
- Vorausgesetzte Definitionen, LINKLIST und LPA 20
- Vorausgesetzte Produkte 5
- Vorausgesetzte Produkte und Software konfigurieren 9
- Voraussetzungen
 - Host 337
 - SMP/E 339
- Voraussetzungen, bekannte Probleme 157
- Voraussetzungen, Developer for System z 337
- Voraussetzungen, Prüfliste 15
- Vorhandene ISPF-Profile verwenden 270
- Vorhandene ISPF-Profile verwenden (JCL für APPC-Transaktion) 273
- VSAM 329
- VTAM 330
- VTAM, alternative Konfigurationsoptionen 334

W

- Web-Service-Schnittstelle 258
 - ADMI 77
 - ADMR 77
 - ADMS 77
- Web-Service-Schnittstelle, CRD-Server 77
- Web-Service-Schnittstelle oder RESTful-Schnittstelle 75
- Webverwaltungsregion 258
- Wichtige Ressourcen, Definitionen 235
 - rsed.envvars 235
 - SYS1.PARMLIB(BPXPRMxx) 236
- Wissenswertes zu Developer for System z 193
- WLM-Klassifikationsregeln 206
- WORKAREA, alte Dateien entfernen 88
- WORKAREA-Verzeichnisbereinigung skulker 108
- Workload Manager 205
- Workload Manager-Änderungen 89

X

- X.509, Hinzufügen der Clientauthentifizierungsunterstützung 307
- X.509-Authentifizierung konfigurieren 299
- X.509-Zertifikat 160
- X.509-Zertifikate, Clientauthentifizierung 171
- Xquickstart, Java-Option 252

Z

- z/OS
 - zusätzlich erforderliche Produkte 340

- z/OS-Host
 - Voraussetzungen 337
- z/OS-Projektperspektive 99
- z/OS UNIX, Positionen für Speicherauszüge 146
- z/OS UNIX-Berechtigungsbits 149
- z/OS UNIX-Dateisystem, Speicherbelegung 232
- z/OS UNIX-Dateisysteme überwachen 244
- z/OS-UNIX-Grenzwerte in BPX-PRMxx 16
- z/OS UNIX-Server, RSE definieren 187
- z/OS UNIX überwachen 241
- z/OS-UNIX-Umgebung, Suchreihenfolge 315
- z/OS UNIX-Unterprojekte, ferne hostbasierte Aktionen 103
- z/OS UNIX-Verzeichnisstruktur
 - grafische Darstellung 202
- Zeichen im Syntaxdiagramm, nicht alphanumerische 135
- Zertifikat, X.509 160
- Zertifikate, Clientauthentifizierung unter Verwendung von X.509 171
- Zertifikatswiderrufsliste (CRL) abfragen
 - CRL-Umgebungsvariablen 173
 - rsed.envvars 173
- zFS-Dateisysteme verwenden 249
- Ziele festlegen, WLM 207
- Ziele in WLM festlegen 207
- Zugriff auf Spooldateien, bedingt 170
- Zugriff auf Systembibliotheken verbessern 249
- Zugriffsmethode, TSO/ISPF-Client-Gateway verwenden 270
- Zugriffsmethoden, TSO 269
- Zugriffsvoraussetzungen 9
- Zuordnungs-Exec verwenden 271
- Zuordnungs-Exec verwenden (JCL für APPC-Transaktion) 274
- Zusätzlich erforderliche Produkte
 - Ant 347
 - Apache Ant 347
 - CICS Transaction Server 344
 - COBOL-Compiler 342
 - DB2 345
 - Debug Tool 343
 - Fault Analyzer 346
 - File Manager 346
 - Host 340
 - IMS 344
 - Java 2 Technology Edition 339
 - PL/I-Compiler 342
 - Ported Tools 347
 - Rational Team Concert for System z 345
 - REXX 346
 - SDK für z/OS 339
 - z/OS 340
- Zusätzlich erforderliche Software, Developer for System z 337
- Zusätzliche Voraussetzungen für z/OS
 - C/C++ 341
 - High Level Assembler 341
 - Language Environment 342
 - SCLM 341

Antwort

IBM Rational Developer for System z
Hostkonfiguration
Version 7.6.1

IBM Form SC12-4062-04

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 0180 3 313233) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: ibmterm@de.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

IBM Deutschland GmbH
SW TSC Germany

71083 Herrenberg



Programmnummer: 5724-T07

SC12-4062-04

