

IBM Rational Developer for System z
Version 9.0.1

Hostkonfigurationsreferenz



IBM Rational Developer for System z
Version 9.0.1

Hostkonfigurationsreferenz



Hinweis

Vor Verwendung dieser Informationen sollten die allgemeinen Informationen unter „Bemerkungen zur Dokumentation für IBM Rational Developer for System z“ auf Seite 227 gelesen werden.

Sechste Edition (Dezember 2013)

Diese Ausgabe bezieht sich auf IBM Rational Developer for System z Version 9.0.1 (Programmnummer 5724-T07) und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

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

IBM freut sich über Anregungen. Wenn Sie Anmerkungen zu diesem Handbuch oder einer anderen Veröffentlichung haben, füllen Sie bitte das Antwortformular am Ende des Handbuchs aus.

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

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM Rational Developer for System z, Version 9.0.1, Host Configuration Reference Guide,
IBM Form SC14-7290-05,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2000, 2013

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

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

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
Dezember 2013

Inhalt

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

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

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

Zielgruppe.	xi
Zusammenfassung der Änderungen	xii
Beschreibung der Dokumentinhalte	xiv
Wissenswertes zu Developer for System z	xiv
Sicherheitsaspekte	xiv
Hinweise zu TCP/IP	xiv
Hinweise zu WLM.	xiv
Optimierungsaspekte	xiv
Leistungsaspekte	xiv
Push-to-Client-Aspekte	xv
CICSTS-Aspekte.	xv
Hinweise zu Benutzerexits	xv
Anpassung der TSO-Umgebung	xv
Ausführung mehrerer Instanzen	xv
Konfigurationsprobleme lösen	xv
SSL- und X.509-Authentifizierung konfigurieren	xv
TCP/IP konfigurieren.	xvi

Developer for System z Hostkonfigurationsreferenz	1
---	---

Kapitel 1. Wissenswertes zu Developer for System z.	3
---	---

Komponentenübersicht	4
RSE als Java-Anwendung	5
Taskeigner	7
Verbindungsflow	8
Integrated Debugger	10
CARMA	11
CARMA-Konfigurationsdateien.	12
CRASTART	12
Batchübergabe	12
Dateisperreneigner	13
Sperren aufheben	14
z/OS UNIX-Verzeichnisstruktur	15
Aktualisierungsberechtigungen für Benutzer ohne Systemadministratorrechte	17
Nützliche Befehle für Sicherheitsfunktion	17
Nützliche z/OS UNIX-Befehle	18
Beispielkonfiguration	18

Kapitel 2. Sicherheitsaspekte	19
---	----

Authentifizierungsmethoden.	20
Benutzer-ID und Kennwort	20
Benutzer-ID und Kennwort für einmaliges Anmelden	20
X.509-Zertifikat	20
Authentifizierung durch JES Job Monitor	20

Debug Manager-Authentifizierung.	21
Verbindungssicherheit	21
Externe Kommunikation auf angegebene Ports beschränken	22
Kommunikation mit SSL oder TLS verschlüsseln	22
Eingangsport überprüfen	23
PassTickets verwenden	23
Prüfprotokollierung.	24
Steuerung der Prüffunktion	24
Prüfprozesse	25
Prüfdaten	25
JES-Sicherheit.	26
Aktionen für Beschränkungen der Jobziele	26
Aktionen für Beschränkungen der Jobausführung	28
Zugriff auf Spooldateien	29
Mit SSL/TLS verschlüsselte Kommunikation	30
Mit Integrated Debugger verschlüsselte Kommunikation	31
Clientauthentifizierung unter Verwendung von X.509-Zertifikaten	32
Prüfung der Zertifizierungsstelle (CA)	33
Zertifikatswiderrufsliste (CRL) abfragen (optional)	33
Authentifizierung durch Ihre Sicherheitssoftware	34
Authentifizierung durch den RSE-Dämon	35
Eingangsport (POE) überprüfen	36
Clientfunktionen ändern	36
OFF.REMOTECOPY.MVS.	37
Push-to-Client-Entwicklergruppen	38
Debug-Sicherheit	39
CICSTS-Sicherheit	40
CRD-Repository	40
CICS-Transaktionen.	40
Mit SSL verschlüsselte Kommunikation	40
Sonstige Informationen	40
GATE-Überlastung (Thrashing).	40
Veraltetes ACEE	41
SCLM-Sicherheit.	41
Konfigurationsdateien für Developer for System z	41
JES Job Monitor - FEJCNFG.	41
RSE - rsed.envvars	42
RSE - ssl.properties.	43
RSE - pushtoclient.properties	43
Sicherheitsdefinitionen.	44
Anforderungen und Prüfliste	44
Sicherheitseinstellungen und -klassen aktivieren	46
OMVS-Segment für Benutzer von Developer for System z definieren.	47
Gestartete Tasks für Developer for System z definieren	47
RSE als sicheren z/OS UNIX-Server definieren	49
Programmgesteuerte MVS-Bibliotheken für RSE definieren	49
PassTicket-Unterstützung für RSE definieren	50
Anwendungsschutz für RSE definieren	51
JES-Befehlssicherheit definieren.	52

Dateiprofile definieren	54
Programmgesteuerte z/OS UNIX-Dateien für RSE definieren	58
Sicherheitseinstellungen prüfen	59

Kapitel 3. Hinweise zu TCP/IP 61

TCP/IP-Ports	61
Externe Kommunikation	62
Interne Kommunikation	62
TCP/IP-Portreservierung	63
CARMA und TCP/IP-Ports	63
LDAP-Aspekte	63
TCP/IP-Standardverhalten überschreiben	64
Verzögertes ACK	64
Mehrfachstack (CINET)	64
CARMA und Stackaffinität	65
crastart*.conf	65
CRASUB*	65
Verteilte dynamische VIPA	66
Portauswahl beschränken	67
Beispielkonfiguration	70
System "SYS1" – TCP/IP-Profil	71
System "SYS2" – TCP/IP-Profil	71

Kapitel 4. Hinweise zu WLM. 73

Klassifikation für Verarbeitungsprozesse	73
Klassifikationsregeln	74
Ziele festlegen	75
Hinweise zur Zielauswahl	76
STC	77
OMVS	77
JES	79
ASCH	80
CICS	80

Kapitel 5. Optimierungsaspekte 83

Ressourcennutzung	83
Überblick	84
Anzahl der Adressräume	85
Anzahl der Prozesse	88
Anzahl der Threads	91
Temporäre Ressourcennutzung	94
Speicherbelegung	95
Begrenzung für die Größe des Java-Heapspeichers	95
Begrenzung für die Größe der Adressräume	95
Richtlinien für Größenschätzungen	96
Beispielanalyse der Speicherbelegung	97
Speicherbelegung im z/OS UNIX-Dateisystem	101
Definitionen von wichtigen Ressourcen	104
/etc/rdz/rsed.envvars	104
SYS1.PARMLIB(BPXPRMxx)	105
Definitionen von verschiedenen Ressourcen	108
EXEC-Karte in der Server-JCL	108
FEK.#CUST.PARMLIB(FEJJCNF)	108
SYS1.PARMLIB(IEASYSxx)	108
SYS1.PARMLIB(IVTPRMxx)	108
SYS1.PARMLIB(ASCHPMxx)	109
Überwachung	109
RSE überwachen	109

z/OS UNIX überwachen	110
Netz überwachen	113
z/OS UNIX-Dateisysteme überwachen	113
Beispielkonfiguration	113
Thread-Pool-Anzahl	114
Mindestbegrenzungen festlegen	114
Grenzwerte definieren	115
Ressourcennutzung überwachen	116

Kapitel 6. Leistungsaspekte 119

Dateisystem zFS verwenden	119
Verwendung von STEPLIB vermeiden	119
Verbesserung des Zugriffs auf Systembibliotheken	119
LE-Laufzeitbibliotheken (Language Environment)	120
Anwendungsentwicklung	120
Verbesserung des Durchsatzes von Sicherheitsprüfungen	121
Auslastungsverwaltung	121
Feste Java-Heapgröße	121
Java-Option '-Xquickstart'	122
Gemeinsame Klassennutzung durch mehrere JVMs	122
Gemeinsame Klassennutzung aktivieren	123
Cachegrößenbegrenzung	123
Cachesicherheit	123
SYS1.PARMLIB(BPXPRMxx)	123
Plattenspeicherplatz	124
Dienstprogramme für Cacheverwaltung	124

Kapitel 7. Push-to-Client-Aspekte 127

Einführung	127
Primäres System	128
Push-to-Client-Metadaten	129
Position von Metadaten	129
Sicherheit der Metadaten	129
Speicherbelegung durch Metadaten	130
Clientkonfigurationssteuerung	130
Clientversionssteuerung	131
Mehrere Entwicklergruppen	131
Aktivierung	131
Gruppenverkettungen	132
Arbeitsbereichsbindung	133
Position der Gruppen-Metadaten	133
Konfigurationsschritte	134
LDAP-basierte Gruppenauswahl	135
LDAP-Schema	136
LDAP-Serverauswahl	136
LDAP-Serverposition	137
Beispielkonfiguration	137
Push-to-Client-Back-End zu LDAP hinzufügen	138
Anfängliche LDAP-Gruppenkonfiguration	138
Entwickler zu LDAP-Gruppen hinzufügen	139
pushtoclient.properties	139
rsed.envvars	140
/var/rdz/pushtoclient/*install	140
SAF-basierte Gruppenauswahl	140
Beispielkonfiguration	141
Sicherheitsdefinition	141
pushtoclient.properties	142

rsed.envvars.	142
/var/rdz/pushtoclient/*install	142
Karenzzeit für die Zurückweisung von Änderungen	142
Hostbasierte Projekte	143
Kapitel 8. CICSTS-Aspekte.	145
RESTful oder Web-Service	146
Primäre und nicht primäre Verbindungsregionen	146
Installation von CICS-Ressourcen protokollieren	147
Application Deployment Manager, Sicherheit.	147
Sicherheit des CRD-Repositorys	147
Pipelinesicherheit	147
Transaktionssicherheit	147
Mit SSL verschlüsselte Kommunikation.	149
Ressourcensicherheit	149
Verwaltungsdienstprogramm	149
Migrationshinweise zum Verwaltungsdienstprogramm	154
Nachrichten des Verwaltungsdienstprogramms	154
CICS-Transaktionsdebugging	157
Kapitel 9. Hinweise zu Benutzerexits 159	
Merkmale von Benutzerexits	159
Aktivierung von Benutzerexits	159
Benutzerexitroutine schreiben	159
Konsolennachrichten	160
Ausführung mithilfe einer variablen Benutzer-ID	160
z/OS UNIX-Shell-Script	161
z/OS UNIX-REXX-Exec	161
Verfügbare Exitpunkte	162
audit.action	162
logon.action	163
Kapitel 10. Anpassung der TSO-Umgebung	165
TSO Commands Service.	165
Zugriffsmethoden	165
TSO/ISPF-Client-Gateway als Zugriffsmethode verwenden	166
ISPF.conf	166
Vorhandene ISPF-Profile verwenden.	166
Verwendung einer Zuordnungs-Exec	167
Mehrere Zuordnungs-Execs verwenden.	167
Mehrere 'ISPF.conf'-Dateien mit mehreren Developer for System z-Konfigurationen	168
Kapitel 11. Ausführung mehrerer Instanzen	169
Identische Konfiguration in einem Sysplex	169
Identische Softwareversionen mit unterschiedlichen Konfigurationsdateien	170
Automatisierte Synchronisierung	171
Alle anderen Situationen	172

Kapitel 12. Konfigurationsprobleme lösen.	175
Protokoll- und Konfigurationsanalyse mit FEK-LOGS	175
Protokolldateien	176
JES Job Monitor, Protokollierung	177
Protokollierung des RSE-Dämons und des Thread-Pools	177
RSE-Benutzer, Protokollierung.	178
SCLM Developer Toolkit, Protokollierung	180
CARMA-Protokollierung	180
Protokollierung des IVP-Tests "fekfivpc"	180
fekfivpi, Protokollierung des IVP-Tests	180
Protokollierung des IVP-Tests fekfvps	181
Protokollierung der Codeüberprüfung	181
Protokollierung der Codeabdeckung.	181
Speicherauszugsdateien	181
MVS-Speicherauszüge	181
Java-Speicherauszüge.	182
Positionen für z/OS UNIX-Speicherauszüge	183
Traceerstellung	184
JES Job Monitor, Traceerstellung	184
RSE, Traceerstellung	184
CARMA, Traceerstellung	185
Fehlerrückmeldungen, Trace	185
z/OS UNIX-Berechtigungsbits.	186
SETUID, Dateisystemattribut	186
Programmsteuerung autorisieren	187
APF-Autorisierung	188
Sticky Bit.	189
Reservierte TCP/IP-Ports	190
Adressraum, Größe	191
Anforderungen an die Start-JCL	191
In SYS1.PARMLIB(BPXPRMxx) festgelegte Begrenzungen	191
Im Sicherheitsprofil gespeicherte Begrenzungen	192
Von Systemexits erzwungene Begrenzungen	192
Begrenzungen für die 64-Bit-Adressierung.	192
Sonstige Informationen	193
Fehlerrückmeldung B37 - Abbruch aufgrund fehlenden Speicherplatzes	193
Systemgrenzwerte	193
Verbindung verweigert	193
OutOfMemoryError	194
Host-Connect-Emulator	194
Kapitel 13. SSL- und X.509-Authentifizierung konfigurieren	195
Auswahl von SSL oder TLS als Verschlüsselungsverfahren.	196
Speicherpositionen für private Schlüssel und Zertifikate festlegen	196
Schlüsseldatei mit RACF erstellen	197
Signiertes Zertifikat verwenden (optional).	198
Vorhandene RSE-Konfiguration klonen	199
Koexistenz durch Aktualisieren von rsed.envvars aktivieren	199
Aktualisierung von ssl.properties durchführen, um SSL zu aktivieren	200
Neuen RSE-Dämon erstellen, um SSL zu aktivieren	200

Verbindung testen	201
Unterstützung der X.509-Clientauthentifizierung hinzufügen (optional)	204
Schlüsseldatenbank mit gskkyman erstellen (optional).	204
Keystore mit keytool erstellen (optional)	207

Kapitel 14. TCP/IP konfigurieren 209

Hostnamen, Abhängigkeit	209
Wissenswertes zu Resolvern	210
Wissenswertes zur Suchreihenfolge für Konfigurationsdaten	210
Suchreihenfolgen in der z/OS UNIX-Umgebung	211
Basiskonfigurationsdateien des Resolvers	211
Umsetztabelle	212
Lokale Hosttabellen	212

Diese Konfigurationsinformationen in Developer for System z anwenden	213
Nicht ordnungsgemäß aufgelöste Hostadresse	216

Literaturübersicht 217

Referenzierte Veröffentlichungen	217
Veröffentlichungen mit weiteren Informationen	220

Glossar 221

Bemerkungen zur Dokumentation für IBM Rational Developer for System z . 227

Index 231

Abbildungen

1.	Komponentenübersicht	4
2.	RSE als Java-Anwendung	5
3.	Taskeigner	7
4.	Verbindungsflow	8
5.	Integrated Debugger	10
6.	CARMA-Flow.	11
7.	Ablauf bei der Enqueue-Bestimmung für Dateien	13
8.	z/OS UNIX-Verzeichnisstruktur.	15
9.	AT-TLS-Richtlinie für Debug Manager	32
10.	TCP/IP-Ports	61
11.	update.sh - DDVIPA-Setup mit einer Firewall unterstützen	68
12.	Beispiel für eine verteilte dynamische VIPA	70
13.	WLM-Klassifikation.	73
14.	Maximale Anzahl von Adressräumen	86
15.	Anzahl der Adressräume pro Client	87
16.	Maximale Anzahl von Prozessen	89
17.	Anzahl von Prozessen für STCRSE.	90
18.	Anzahl von Prozessen pro Client	90
19.	Maximale Anzahl von Threads in einem RSE-Thread-Pool	93
20.	Maximale Anzahl von Threads in einem RSE-Thread-Pool	93
21.	Ressourcennutzung mit 5 Anmeldungen	98
22.	Ressourcennutzung mit 5 Anmeldungen (Fortsetzung).	99
23.	Ressourcennutzung beim Bearbeiten eines Members der untergliederten Datei	100
24.	Speicherbelegung im z/OS UNIX-Dateisystem	102
25.	Ressourcennutzung der Beispielkonfiguration	117
26.	Musterdefinition für das LDAP-Schema	136
27.	ADNJSPAU - CICSTS-Verwaltungsdienstprogramm.	151
28.	ADNJSPAU - CICSTS-Verwaltungsdienstprogramm (Teil 2 von 3)	152
29.	ADNJSPAU - CICSTS-Verwaltungsdienstprogramm (Teil 3 von 3)	153
30.	RSEDSSL - RSE-Dämonbenutzerjob für SSL	201
31.	Dialog 'Hostzertifikat importieren'	202
32.	Vorgabendialog - SSL	203

Tabellen

1.	JES Job Monitor, Konsolbefehle	26	23.	Benutzerspezifische Ressourcennutzung	85
2.	Matrix der Befehlsberechtigungen für LIMIT- _COMMANDS	27	24.	Anzahl der Adressräume	85
3.	Erweiterte JESSPOOL-Profilen	27	25.	Begrenzungen für Adressräume	88
4.	LIMIT_CONSOLE (Berechtigungsmatrix für Konsole)	28	26.	Anzahl der Prozesse	88
5.	Berechtigungsmatrix zum Durchsuchen für LI- MIT_VIEW.	29	27.	Begrenzungen für Prozesse	90
6.	Mechanismen für den SSL-Zertifikatsspeicher	30	28.	Anzahl der Threads	91
7.	SAF-Informationen zur Änderung von Client- funktionen	37	29.	Begrenzungen für Threads	94
8.	Push-to-Client-relevante SAF-Informationen	38	30.	Referenzeinstellungen für die Speicherbele- gung.	97
9.	SAF-Informationen für Debugfunktionen	39	31.	Anweisungen für die Protokollausgabe	103
10.	Variablen für die Sicherheitskonfiguration	44	32.	Anweisungen für temporäre Ausgabe	104
11.	Bedienerbefehle von JES2 Job Monitor	53	33.	Matrix zur Unterstützung von Push-to-Client- Gruppen für '*.enabled'	131
12.	Bedienerbefehle von JES3 Job Monitor	53	34.	Matrix zur Unterstützung von Push-to-Client- Gruppen für 'reject.*.updates'	132
13.	WLM-Einstiegspunkt-Subsysteme	74	35.	Push-to-Client-Gruppenverkettungen	132
14.	WLM-Qualifikationsmerkmale für Arbeitsvor- gänge	75	36.	Push-to-Client-relevante LDAP-Informationen	135
15.	WLM-Verarbeitungsprozesse	76	37.	Push-to-Client-relevante SAF-Informationen	140
16.	WLM-Verarbeitungsprozesse - STC.	77	38.	Variablen für JAVA_DUMP_TDUMP_PAT- TERN	182
17.	WLM-Verarbeitungsprozesse - OMVS	78	39.	Mechanismen für den SSL-Zertifikatsspeicher	196
18.	WLM-Verarbeitungsprozesse - JES	79	40.	Für den Resolver verfügbare lokale Definitio- nen	215
19.	WLM-Verarbeitungsprozesse - ASCH	80	41.	Referenzierte Veröffentlichungen	217
20.	WLM-Verarbeitungsprozesse - CICS	80	42.	Referenzierte Websites	219
21.	Allgemeine Ressourcennutzung	84	43.	Veröffentlichungen mit weiteren Informatio- nen	220
22.	Benutzerspezifische vorausgesetzte Ressour- cennutzung	84			

Zu diesem Handbuch

Dieses Dokument enthält Hintergrundinformationen zu verschiedenen Konfigurationstasks von IBM® Rational Developer for System z selbst sowie zu anderen z/OS-Komponenten und -Produkten (wie WLM und CICS).

Im weiteren Verlauf dieses Handbuchs werden die folgenden Namen verwendet:

- *IBM Rational Developer for System z* wird als *Developer for System z* bezeichnet.
- *IBM Rational Developer for System z Integrated Debugger* wird als "Integrated Debugger" bezeichnet.
- *Common Access Repository Manager* wird mit CARMA abgekürzt.
- *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.

Dieses Dokument ist Teil einer Reihe von Dokumenten, in denen die Hostkonfiguration von Developer for System z beschrieben wird. Jedes dieser Dokumente hat eine spezielle Zielgruppe. Sie müssen nicht alle Dokumente lesen, um die Konfiguration von Developer for System z abzuschließen.

- Im Handbuch *Rational Developer for System z Hostkonfiguration* (IBM Form SC12-4062) werden alle Planungstasks, Konfigurationstasks und Optionen (einschließlich der optionalen) ausführlich beschrieben und alternative Szenarien bereitgestellt.
- Das Handbuch *Rational Developer for System z Hostkonfigurationsreferenz* (IBM Form SC12-4489) beschreibt die Developer for System z-Gestaltung und liefert außerdem Hintergrundinformationen zu verschiedenen Konfigurationstasks von Developer for System z, z/OS-Komponenten sowie anderen Produkten (wie WLM und CICS) im Zusammenhang mit Developer for System z.
- Im Handbuch *Rational Developer for System z Leitfaden für den Schnelleinstieg in die Hostkonfiguration* (IBM Form GI11-3191) wird eine Minimalkonfiguration von Developer for System z beschrieben.
- Im Handbuch *Rational Developer for System z Host Configuration Utility* (IBM Form SC14-7282) wird das Dienstprogramm für die Hostkonfiguration (Host Configuration Utility) beschrieben, eine ISPF-Anzeigeanwendung, die Sie durch grundlegende und allgemeine optionale Anpassungsschritte für Developer for System z führt.

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

Zielgruppe

Dieses Handbuch wendet sich an Systemprogrammierer, die IBM Rational Developer for System z Version 9.0.1 konfigurieren und optimieren.

Während die eigentlichen Konfigurationsschritte in einer anderen Veröffentlichung beschrieben werden, werden in dieser Veröffentlichung verschiedene zugehörige Themen (wie Optimierung, Sicherheitskonfiguration usw.) ausführlich aufgelistet.

Voraussetzung für die Verwendung dieses Handbuchs ist, dass Sie mit z/OS UNIX System Services und MVS-Hostsystemen vertraut sind.

Zusammenfassung der Änderungen

In diesem Abschnitt werden die Änderungen in der Veröffentlichung *IBM Rational Developer for System z Version 9.0 Hostkonfigurationsreferenz* (IBM Form SC12-4489-05) zusammengefasst (Aktualisierung vom Dezember 2013).

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

| Neue Informationen:

- | • Hinzugefügte Informationen zu den Protokolldateinamen mit Zeitmarke. Lesen Sie hierzu den Abschnitt „Protokolldateien“ auf Seite 176.
- | • Hinzugefügte Informationen zu den neuen prüfbaren Ereignissen. Siehe Prüfdaten.

| Dieses Dokument enthält Informationen, die bisher im Handbuch "IBM Rational Developer for System z Version 9.0 Hostkonfigurationsreferenz" (IBM Form SC12-4489-04) enthalten waren.

| Neue Informationen:

- | • Aktualisierte Nutzung des TCP/IP-Ports. Lesen Sie hierzu den Abschnitt „TCP/IP-Ports“ auf Seite 61.
- | • Hinzugefügtes Beispiel für die automatische Synchronisierung von zwei RSE-Dämonen. Lesen Sie hierzu den Abschnitt „Automatisierte Synchronisierung“ auf Seite 171.
- | • Hinzugefügte Informationen zu neuen Protokolldateien. Lesen Sie hierzu den Abschnitt „Protokolldateien“ auf Seite 176.

| Dieses Dokument enthält Informationen, die bisher im Handbuch *IBM Rational Developer for System z Version 8.5.1 Hostkonfigurationsreferenz* (IBM Form SC12-4489-03) enthalten waren.

| Neue Informationen:

- | • Hinzugefügte Informationen über SAF-Profile zum Ändern von Clientfunktionen. Lesen Sie hierzu den Abschnitt „Clientfunktionen ändern“ auf Seite 36.
- | • Aktualisierte Zahlen für Ressourcennutzung. Siehe Kapitel 5, „Optimierungsaspekte“, auf Seite 83.
- | • Aktualisierter Standardwert für maximale Anzahl von Benutzern pro Thread-Pool. Lesen Sie hierzu den Abschnitt Kapitel 5, „Optimierungsaspekte“, auf Seite 83.

| Dieses Dokument enthält Informationen, die bisher im Handbuch *IBM Rational Developer for System z Version 8.5 Hostkonfigurationsreferenz* (IBM Form SC12-4489-02) enthalten waren.

| Neue Informationen:

- | • Aktualisierte JES Job Monitor-Sicherheitsinformationen. Lesen Sie hierzu den Abschnitt Kapitel 2, „Sicherheitsaspekte“, auf Seite 19.
- | • Hinzugefügte Informationen über Benutzerexits. Lesen Sie hierzu den Abschnitt Kapitel 9, „Hinweise zu Benutzerexits“, auf Seite 159.

Dieses Dokument enthält Informationen, die bisher im Handbuch *IBM Rational Developer for System z Version 8.0.3 Hostkonfigurationsreferenz* (IBM Form SC12-4489-01) enthalten waren.

Neue Informationen:

- Aktualisierte z/OS UNIX-Verzeichnisstruktur. Lesen Sie hierzu den Abschnitt „z/OS UNIX-Verzeichnisstruktur“ auf Seite 15.
- Hinzugefügte Informationen über hostbasierte Clientsteuerung. Lesen Sie hierzu den Abschnitt Kapitel 7, „Push-to-Client-Aspekte“, auf Seite 127.
- Hinzugefügte sicherheitsrelevante Push-to-Client-Informationen. Lesen Sie hierzu den Abschnitt „Push-to-Client-Entwicklergruppen“ auf Seite 38.
- Verwendung von verwalteten ACEEs dokumentieren. Lesen Sie hierzu den Abschnitt „Verwaltetes ACEE“ auf Seite 41.
- Hinzugefügte Informationen über die automatisierte Prüfprotokollverarbeitung. Lesen Sie hierzu den Abschnitt „Prüfprozesse“ auf Seite 25.
- Aktualisierte Informationen über sicherheits- und prüfungsbezogene Anweisungen in Konfigurationsdateien. Lesen Sie hierzu den Abschnitt „Konfigurationsdateien für Developer for System z“ auf Seite 41.
- Hinzugefügte zusätzliche TCP/IP-Informationen. Lesen Sie hierzu den Abschnitt Kapitel 3, „Hinweise zu TCP/IP“, auf Seite 61.
- Aktualisierte Informationen zur Zertifikatsberechtigung für die SSL-Kommunikation. Lesen Sie hierzu den Abschnitt Kapitel 13, „SSL- und X.509-Authentifizierung konfigurieren“, auf Seite 195.
- Aktualisierte Ressourcennutzung. Lesen Sie hierzu den Abschnitt „Ressourcennutzung“ auf Seite 83.

Dieses Dokument enthält Informationen, die bisher im Handbuch *IBM Rational Developer for System z Version 8.0.1 Hostkonfigurationsreferenz* (IBM Form SC12-4489-00) enthalten waren.

Neue Informationen:

- Abschnitt "CARMA" in "Wissenswertes zu Developer for System z". Siehe „CARMA“ auf Seite 11.
- Allgemeine Referenzinformationen zu TCP/IP. Lesen Sie hierzu den Abschnitt Kapitel 3, „Hinweise zu TCP/IP“, auf Seite 61.
- Problemlösung für Fehler "B37" - Abbruch aufgrund fehlenden Speicherplatzes. Lesen Sie hierzu den Abschnitt „Fehlerrückmeldung B37 - Abbruch aufgrund fehlenden Speicherplatzes“ auf Seite 193.

Entfernte Informationen:

- Die Informationen, die bisher im Handbuch *IBM Rational Developer for System z Version 7.6.1 Hostkonfiguration* (IBM Form SC12-4062-04) enthalten waren, wurden jetzt in zwei Dokumente aufgeteilt: *IBM Rational Developer for System z Hostkonfiguration* (IBM Form SC12-4062) und *IBM Rational Developer for System z Hostkonfigurationsreferenz* (IBM Form SC12-4489).
- Informationen zur Konfiguration von APPC wurden in das White Paper *Using APPC to provide TSO command services* (IBM Form SC14-7291) verschoben.
- INETD konfigurieren

Beschreibung der Dokumentinhalte

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

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.

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.

Hinweise zu TCP/IP

Developer for System z verwendet TCP/IP, um Benutzern einer Workstation den Zugriff auf Mainframe-Computer bereitzustellen, wenn diese selbst kein Mainframe-Computer ist. TCP/IP wird außerdem für die Datenübertragung zwischen verschiedenen Komponenten und anderen Produkten verwendet.

Hinweise zu WLM

Im Gegensatz zu herkömmlichen z/OS-Anwendungen ist Developer for System z keine einzelne Anwendung, die von Workload Manager (WLM) auf einfache Weise erkannt wird. Developer for System z 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 17 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.

Push-to-Client-Aspekte

Push-to-Client bzw. die hostbasierte Clientsteuerung unterstützt die zentrale Verwaltung der folgenden Komponenten:

- Clientkonfigurationsdateien
- Clientproduktversion
- Projektdefinitionen

CICSTS-Aspekte

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

Hinweise zu Benutzerexits

In diesem Kapitel finden Sie Informationen dazu, wie Sie Exitroutinen schreiben können, die Developer for System z funktional erweitern.

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.

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

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.

Developer for System z Hostkonfigurationsreferenz

Kapitel 1. 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“ auf Seite 4
- „RSE als Java-Anwendung“ auf Seite 5
- „Taskeigner“ auf Seite 7
- „Verbindungsflow“ auf Seite 8
- „Integrated Debugger“ auf Seite 10
- „CARMA“ auf Seite 11
- „Dateisperreneigner“ auf Seite 13
- „z/OS UNIX-Verzeichnisstruktur“ auf Seite 15

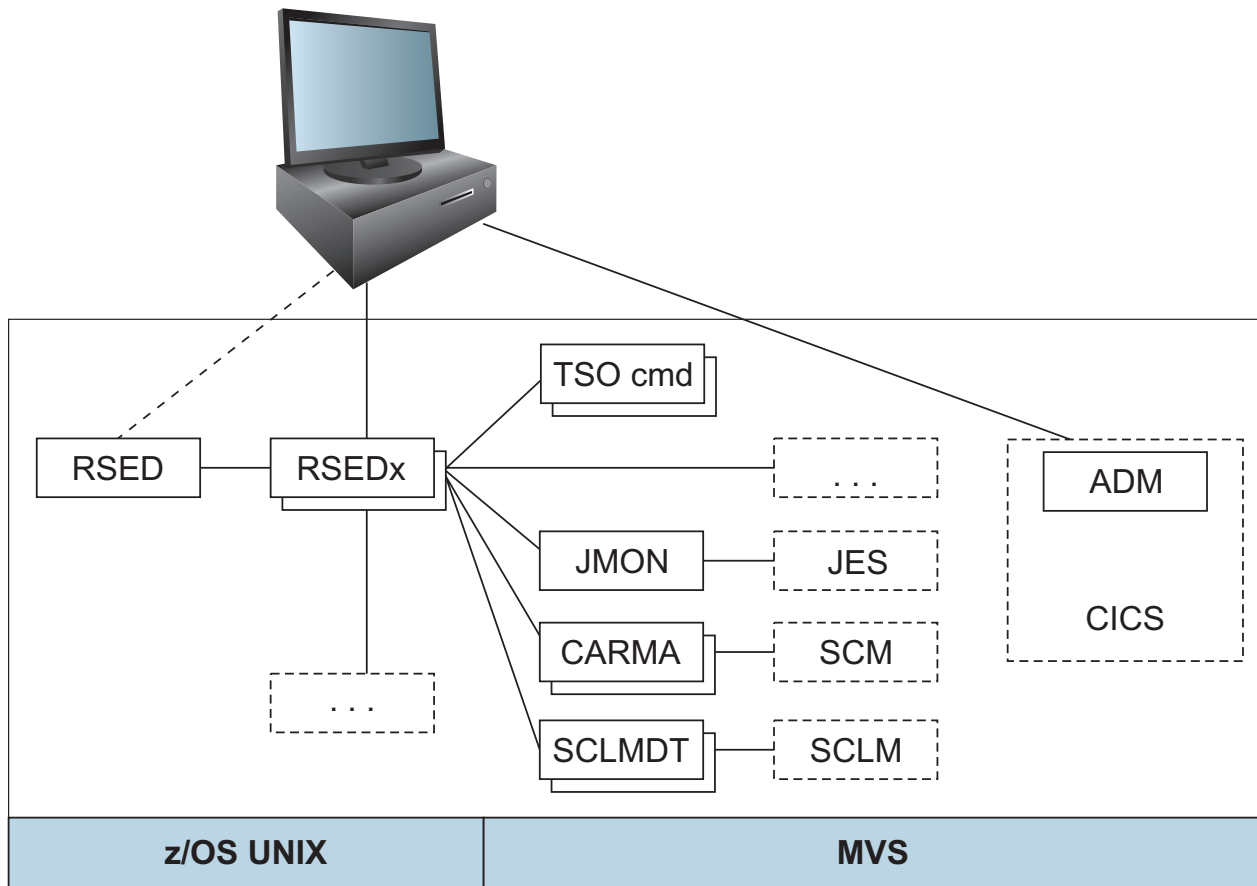


Abbildung 1. Komponentenübersicht

Abb. 1 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.
- 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 Edevor.
- SCLM Developer Toolkit (SCLMDT) stellt eine Schnittstelle zur Verfügung, um SCLM zu erweitern und mit SCLM zu interagieren.

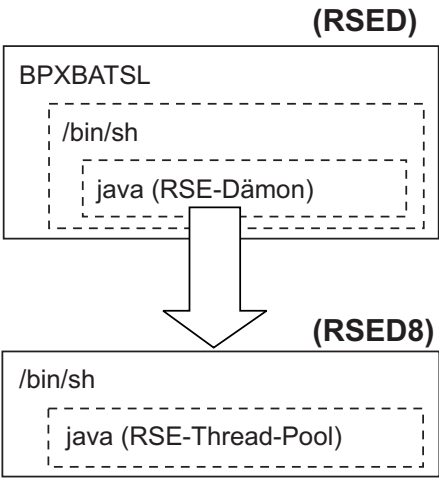
- Application Deployment Manager (ADM) stellt verschiedene CICS-bezogene Services bereit.
- Es sind weitere Services verfügbar. Diese können von Developer for System z selbst oder von zusätzlich erforderlicher Software bereitgestellt werden.

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

System - gemeinsam genutzt
System - privat
Code (z/OS UNIX, Java, RSE)
Java-Heapspeicher
Nicht in Verwendung

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 KERNEL WAIT	308	307	java...

Abbildung 2. RSE als Java-Anwendung

Abb. 2 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. 2 auf Seite 5 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-Skript `rsed.sh` in einem separaten Prozess (`/bin/sh`) ausgeführt – PID 67109114.
3. Das Shell-Skript 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.

RSE kann im 31-Bit- oder 64-Bit-Adressierungsmodus ausgeführt werden, was zu unterschiedlichen Speichergrenzen führt. Im 31-Bit-Modus ist der verfügbare Speicher auf 2 GB begrenzt, während es im 64-Bit-Modus keine Einschränkung gibt, sofern es nicht anders in `SYS1.PARMLIB` angegeben ist.

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. Bei der Ausführung im 64-Bit-Modus versucht Java, den Heapspeicher über der 2-GB-Grenze zuzuordnen, wodurch Speicher unterhalb der Grenze freigegeben wird.

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

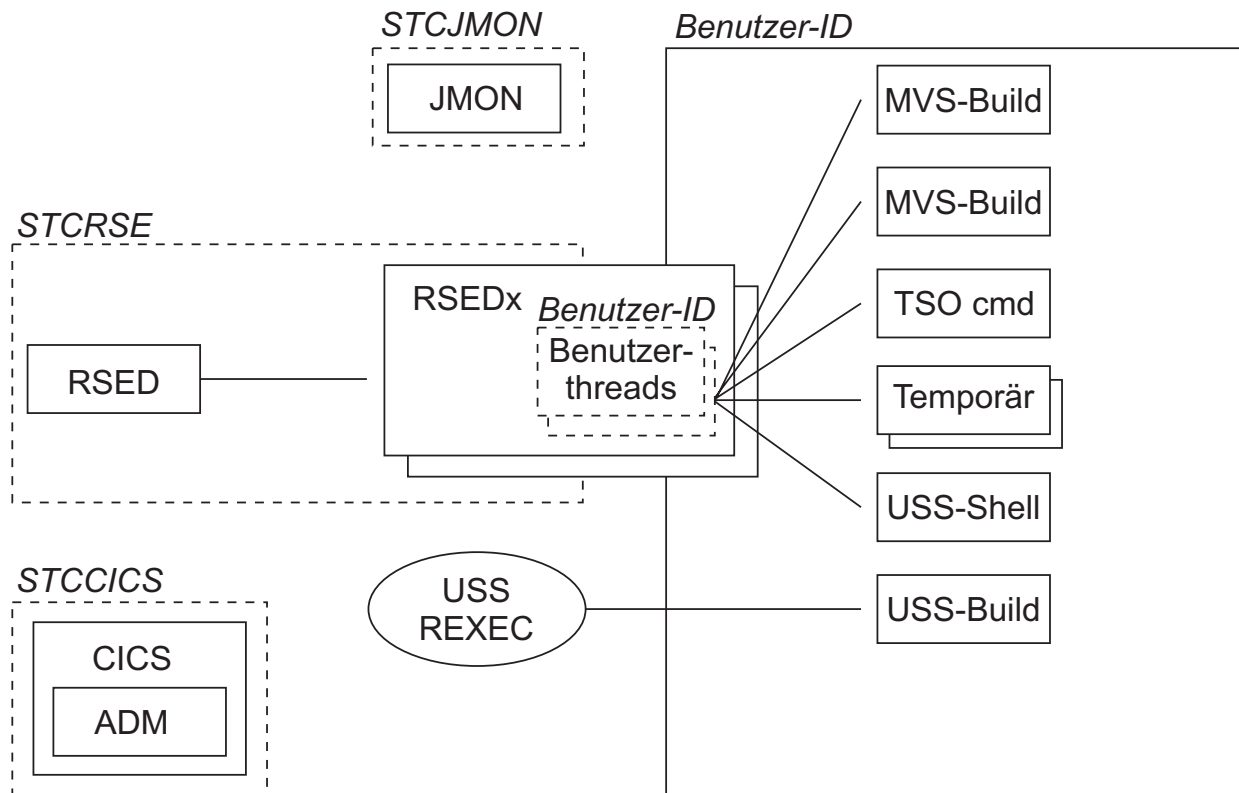


Abbildung 3. Taskeigner

Abb. 3 zeigt eine Basisübersicht über die Eigner der Sicherheitsberechtigungsnaehweise, 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. 3 zeigt die gestarteten Tasks in Developer for System z (JMON und RSED) sowie gestartete Beispieltasks und Beispielsystems-services, mit denen Developer for System z kommuniziert. Application Deployment Manager (ADM) ist innerhalb einer CICS-Region aktiv. Der 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. 3 auf Seite 7 aufgeführten Adressräume bleiben für einen längeren Zeitraum im System sichtbar. Sie sollten allerdings beachten, dass z/OS UNIX so entwickelt wurde, dass es auch einige kurz andauernde, temporäre Adressräume gibt.

Verbindungsflow

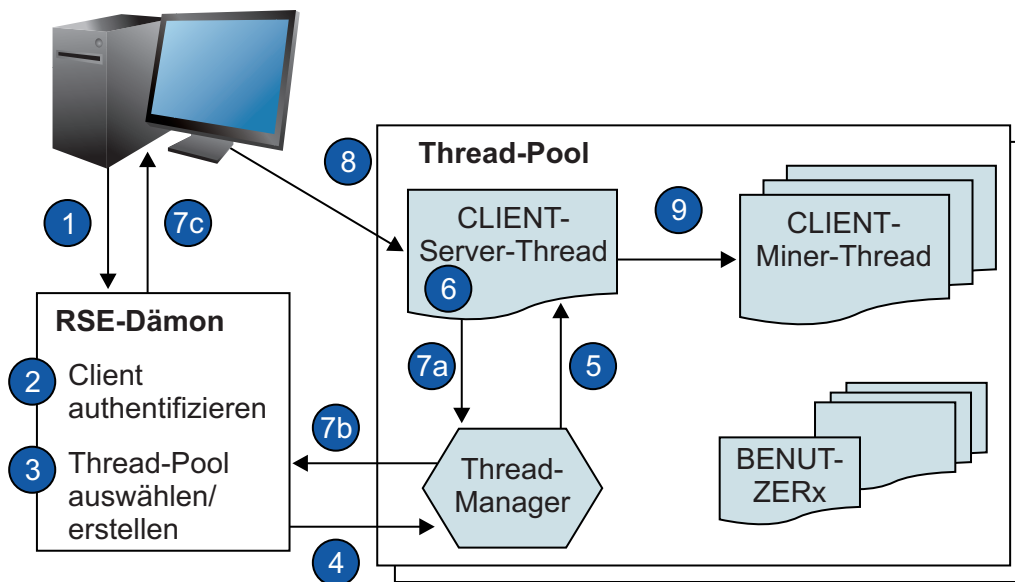


Abbildung 4. Verbindungsflow

Abb. 4 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 vorangegangene 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 17 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 `rsd.envvars` angegeben ist, bindet der RS-Server an einen Port aus diesem Bereich. Steht kein Port zur Verfügung, schlägt die Verbindung fehl. Der RSE-Server muss den Port nicht exklusiv für die Dauer der Clientverbindung benötigen. Es kann sich nur während der Serververbindung an den Port und des Verbindungsaufbaus des Clients kein anderer RSE-Server an den Port binden. Das bedeutet, dass für die meisten Verbindungen der erste Port des Portbereichs verwendet wird, die restlichen Ports des Bereichs also nur als Puffer für den Fall dienen, dass mehrere Anmeldungen gleichzeitig erfolgen.
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.

Integrated Debugger

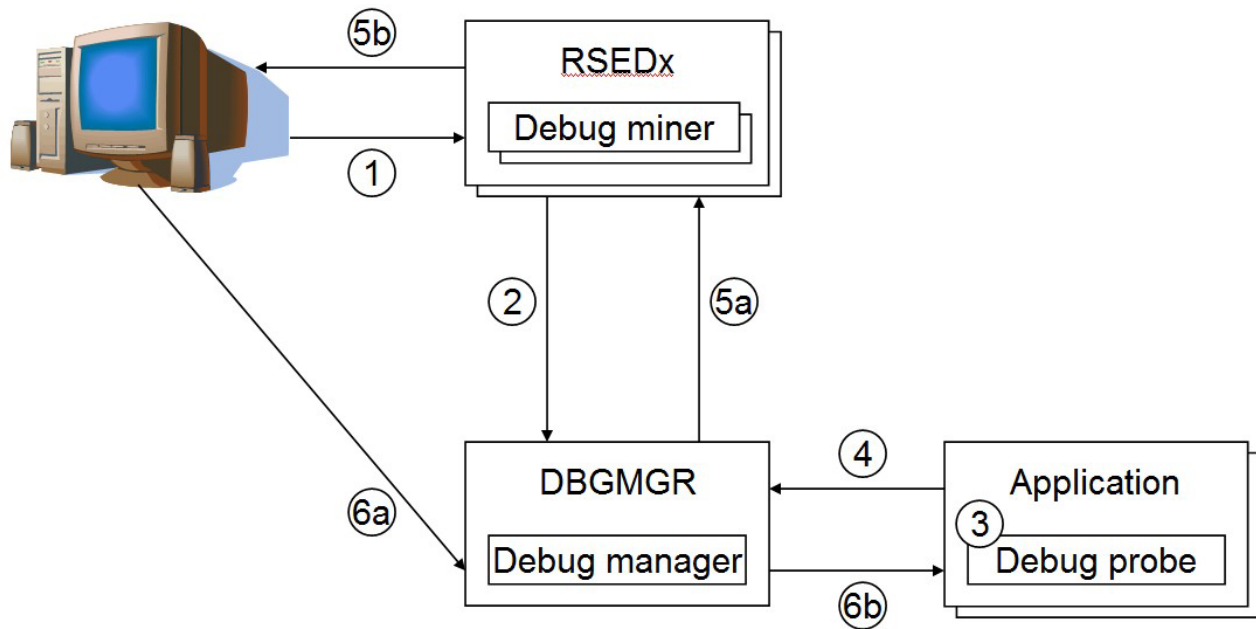


Abbildung 5. Integrated Debugger

Integrated Debugger wird zum Debuggen verschiedener Anwendungen verwendet. In Abbildung 5 wird eine schematische Übersicht gezeigt, wie ein Developer for System z-Client ein Debugging für eine Anwendung durchführen kann.

1. Der Client wird mit der normalen Developer for System z-Host-Anmeldung mit dem Host verbunden.
2. Als Teil der Anmeldung registriert Debug Miner den Benutzer bei Debug Manager, der in der gestarteten DBGMGR-Task aktiv ist.
3. Wenn eine Anwendung mit einem Anzeiger gestartet wird, ruft Language Environment (LE) den Debug-Testmonitor auf.
4. Der Debug-Testmonitor wird bei Debug Manager registriert.
5. Mithilfe von Debug Miner benachrichtigt Debug Manager den Developer for System z-Client des Benutzers, der diese Debugsitzung empfängt. Wenn der Benutzer zu diesem Zeitpunkt nicht registriert wird, ruht die Debugsitzung und wartet darauf, dass der Benutzer bei Debug Manager registriert wird.
6. Die Debug-Engine im Client kontaktiert Debug Manager, der wiederum die Daten zwischen der Debug-Engine und dem Debug-Testmonitor übergibt.

CARMA

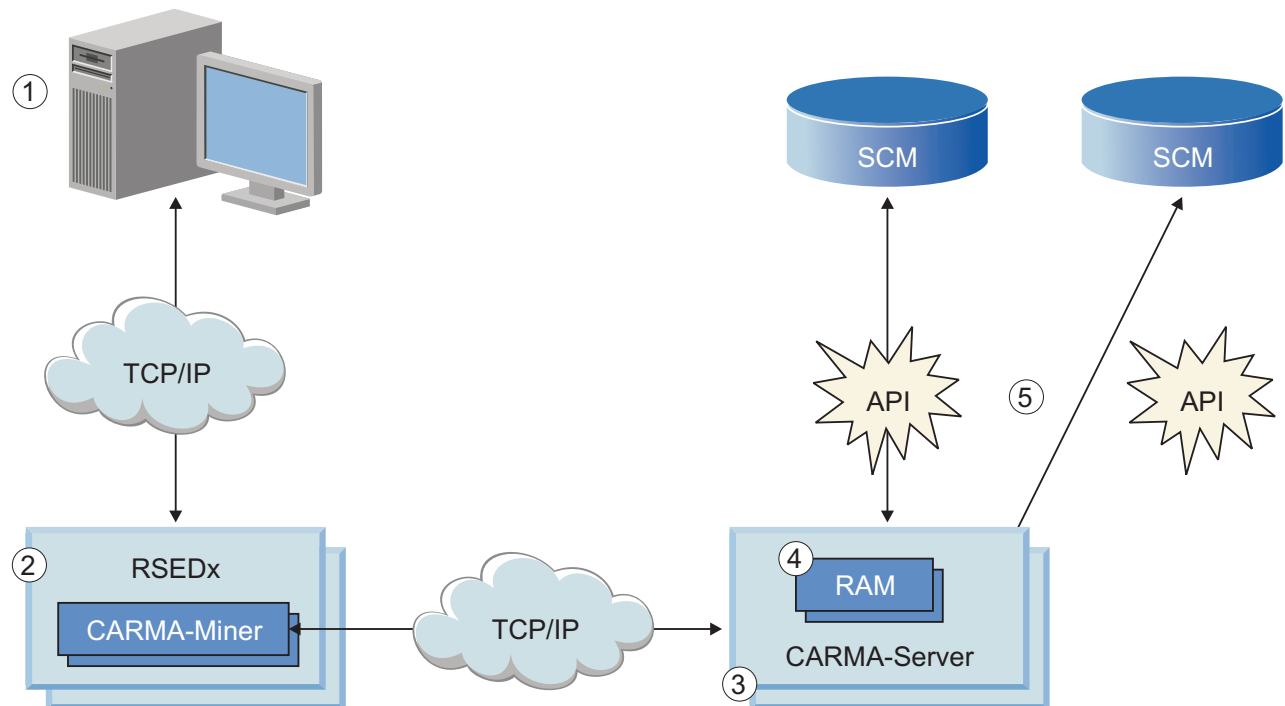


Abbildung 6. CARMA-Flow

CARMA (Common Access Repository Manager) wird für den Zugriff auf einen hostbasierten Software Configuration Manager (SCM) verwendet, beispielsweise CA Endeavor® SCM. Abb. 6 zeigt in einer schematischen Übersicht, wie ein Client von Developer for System z auf jeden beliebigen unterstützten, hostbasierten Software Configuration Manager (SCM) zugreifen kann.

1. Der Client verwendet ein CARMA-Plug-in (Common Access Repository Manager).
2. Das CARMA-Plug-in kommuniziert mit dem CARMA-Miner, der als benutzer-spezifischer Thread im RSE-Thread-Pool (RSEDx) aktiv ist. Diese Kommunikation erfolgt über eine vorhandene RSE-Verbindung.
3. Wenn der Client Zugriff auf einen SCM anfordert, bindet der CARMA-Miner an einen TCP/IP-Port und startet einen benutzerspezifischen CARMA-Server mit der Portnummer als Startargument. Der CARMA-Server stellt dann eine Verbindung zu diesem Port her und verwendet den Pfad für die Kommunikation mit dem Client. Beachten Sie, dass hostbasierte SCMs Einzelbenutzeradress-räume erwarten, um auf ihre Services zuzugreifen; hierfür muss CARMA pro Benutzer einen CARMA-Server starten. Es ist nicht möglich, einen einzigen Ser-ver zu erstellen, der mehrere Benutzer unterstützt.
4. Der CARMA-Server lädt den Repository Access Manager (RAM), der den angeforderten SCM unterstützt.
5. Der RAM bearbeitet die technischen Details der Interaktion mit dem spezi-fischen SCM und stellt eine gemeinsame Schnittstelle für den Client dar.

CARMA-Konfigurationsdateien

Developer for System z unterstützt mehrere Methoden für den Start eines CARMA-Servers. Alle Methoden haben Vor- und Nachteile. Außerdem stellt Developer for System z mehrere Repository Access Manager (RAM) bereit, die in zwei Gruppen eingeteilt werden können: Produktions-RAM und Muster-RAM. Es sind verschiedene Kombinationen von RAM und Serverstartmethoden als vorkonfigurierte Installation verfügbar.

Alle Serverstartmethoden nutzen eine gemeinsame Konfigurationsdatei (CRASRV.properties), die unter anderem angibt, welche Startmethode verwendet wird.

CRASTART

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.

RSE ruft das Lademodul CRASTART auf, das ausgehend von den Definitionen in crastart*.conf eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen erstellt. Developer for System z kann in dieser Umgebung den CARMA-Server CRASERV ausführen. Developer for System z stellt mehrere Dateien crastart*.conf bereit, die jeweils für einen bestimmten RAM vorkonfiguriert sind.

Batchübergabe

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.

RSE ruft die CLIST CRASUB* auf, die wiederum eine eingebettete JCL übergibt, um eine gültige Umgebung für die Ausführung von TSO- und ISPF-Batchbefehlen zu erstellen. Developer for System z führt in dieser Umgebung den CARMA-Server (CRASERV) aus. Developer for System z stellt mehrere Member "CRASUB*" bereit, die jeweils für einen bestimmten RAM vorkonfiguriert sind.

Dateisperreneigner

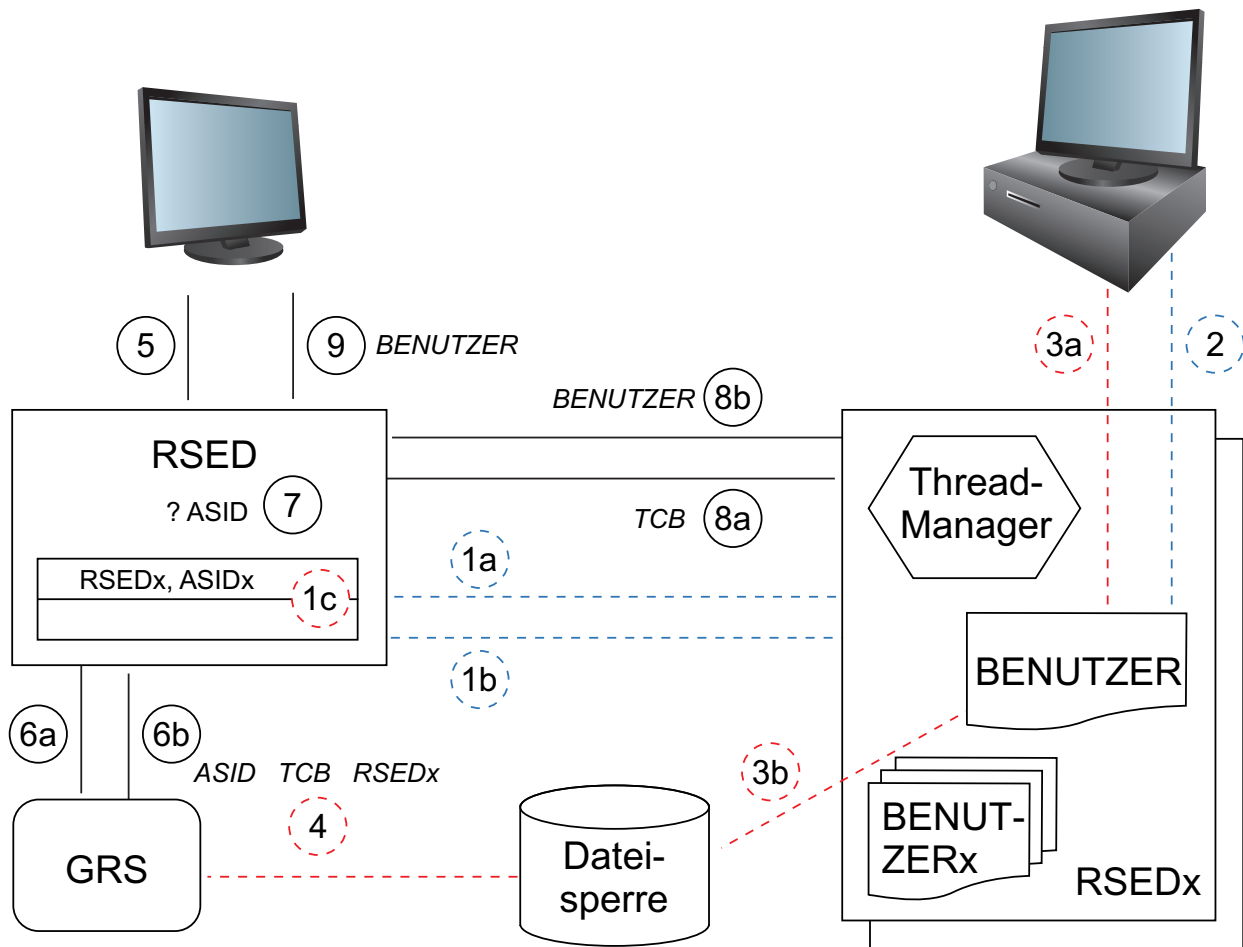


Abbildung 7. Ablauf bei der Enqueue-Bestimmung für Dateien

Abb. 7 veranschaulicht anhand einer schematischen Übersicht, wie der RSE-Dämon ermittelt, welcher Developer for System z-Client Eigner einer Dateisperre ist.

1. Der RSE-Dämon (RSED) erstellt einen Thread-Pool (RSEDx). Der Thread-Pool bestätigt, dass der Startvorgang abgeschlossen wurde, indem er seine Adressraumkennung (ASID des Thread-Pools) zurück an den RSE-Dämon überträgt. Dieser speichert die ASID in dem Steuerblock, der eigens für die Verfolgung (Überwachung) dieses Thread-Pools erstellt wurde.
2. Der Client meldet sich an. Dadurch wird ein benutzerspezifischer RSE-Server-Thread (USER) innerhalb eines Thread-Pools (RSEDx) erstellt. Jeder Thread besitzt eine eindeutige TCB-Kennung (TCB: Task Control Block).
3. Der Client öffnet eine Datei zur Bearbeitung. Dies weist den RSE-Server an, eine exklusive Sperre für die Datei abzurufen (Enqueue bzw. Einreihung).
4. Das System registriert die ASID, den TCB und den Tasknamen (RSEDx) des anfordernden Benutzers als Teil des Einreihungsvorgangs. Diese Informationen werden in den GRS-Warteschlangen (Global Resource Serialization) gespeichert.
5. Ein Operator fragt den Sperrstatus der Datei beim RSE-Dämon ab.

6. Der RSE-Dämon durchsucht die GRS-Warteschlangen nach der Information, ob die Datei gesperrt ist, und ruft die ASID, den TCB und den Tasknamen des Sperreneigners ab.
7. Die abgerufene ASID wird mit den ASIDs der verschiedenen Thread-Pools verglichen.
8. Der RSE-Dämon weist den Thread-Pool, der Eigner der ASID ist, dazu an, zu ermitteln, welcher Benutzer Eigner des TCBs ist.
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, hat z/OS die Fähigkeit verloren, mit dem Bedienerbefehl **DISPLAY GRS,RES=(*,dataset*)** zu verfolgen, wer der Eigner einer Sperre für eine Datei oder ein Member ist. Systembefehle stoppen auf der Adressraumbene, die dem RSE-Thread-Pool entspricht.

Developer for System z spricht dieses Problem durch Bereitstellung des Bedienerbefehls **MODIFY rsed APPL=DISPLAY OWNER,DATASET=dataset** an. Eine entsprechende Beschreibung hierzu enthält das Kapitel "Operatorbefehle" im Handbuch *Hostkonfiguration* (IBM Form SC12-4062). Dieser Bedienerbefehl ist in der Lage, sämtliche von RSE-Benutzern verhängten Sperren für Dateien und Member aufzuheben sowie die von anderen Produkten (wie etwa ISPF) gesetzten Sperren zu lösen.

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 Editierung beendet.

Bestimmte Fehlerbedingungen können den ordnungsgemäßen Ablauf dieses Mechanismus beeinträchtigen. In diesem Fall kann der Benutzer, der Eigentümer der Sperre ist, mithilfe des RSE-Bedienerbefehls **modify cancel** abgebrochen werden, wie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062) beschrieben. Während dieses Prozesses werden alle aktiven Sperren von Dateien aufgehoben, die mit diesem Benutzer verknüpft sind.

z/OS UNIX-Verzeichnisstruktur

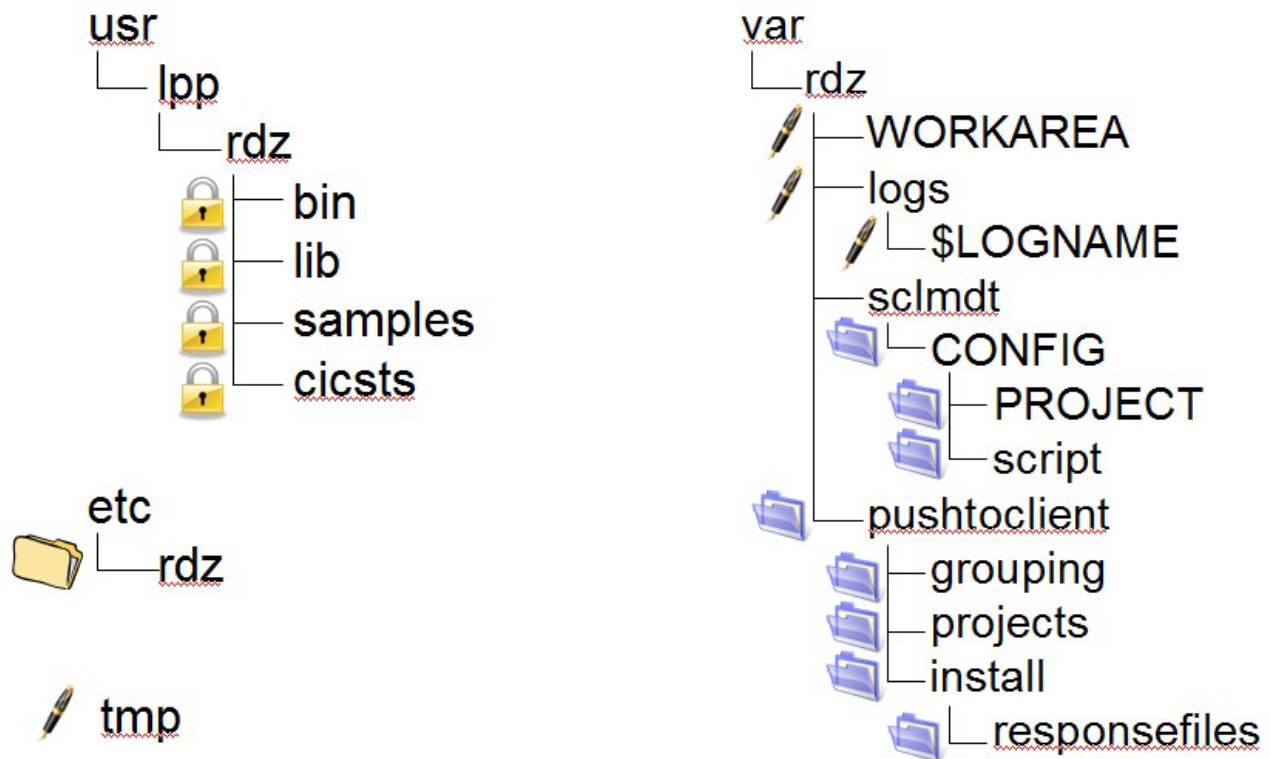


Abbildung 8. z/OS UNIX-Verzeichnisstruktur

Abb. 8 liefert einen Überblick über die von Developer for System z verwendeten z/OS UNIX-Verzeichnisse. Die folgende Liste enthält nicht nur Informationen zu jedem von Developer for System z verwendeten Verzeichnis, sondern gibt auch an, 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 der gestarteten Task RSED angegeben (Variable HOME). Die enthaltenen Dateien werden von SMP/E verwaltet.
- **/etc/rdz/** enthält die RSE- und Miner-bezogenen Konfigurationsdateien. Die eigentliche Position ist in der gestarteten Task RSED angegeben (Variable CNFG). Die enthaltenen Dateien werden vom Systemprogrammierer verwaltet.
- **/tmp/** wird vom TSO/ISPF-Client-Gateway von ISPF und verschiedenen Miners verwendet, um temporäre Daten zu speichern. Einige IVPs speichern ihre Ausgabe hier. Die enthaltenen Dateien werden von ISPF, den Miners und den IVPs verwaltet. Die eigentliche Position kann in `rsed.envvars` mit der Variablen `TMPODIR` angepasst werden. 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.

- **/var/rdz/WORKAREA/** wird vom TSO/ISPF-Client-Gateway von ISPF und SCLMDT verwendet, um Daten zwischen z/OS UNIX und MVS-basierten Adressräu-

men zu übertragen. Die eigentliche Position ist in `rsed.envvars` angegeben (Variable `CGI_ISPWORK`). 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.

- `/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/sclmdt/CONFIG/` sperrt 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/sclmdt/CONFIG/PROJECT/` sperrt SCLMDT-Projektkonfigurationsdateien. Die eigentliche Position ist in `rsed.envvars` angegeben (Variable `SCLMDT_CONF_HOME`). Die enthaltenen Dateien werden vom SCLM-Administrator verwaltet.
- `/var/rdz/sclmdt/CONFIG/script/` sperrt 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/pushtoclient/` enthält Clientkonfigurationsdateien, Update-Informationen zum Clientprodukt und hostbasierte Projektinformationen, die bei der Verbindung mit dem Host mittels Push auf den Client übertragen werden. Die eigentliche Position ist in `pushtoclient.properties` angegeben (Variable `pushtoclient.folder`). Die darin enthaltenen Dateien werden von einem Clientadministrator von Developer for System z verwaltet.
- `/var/rdz/pushtoclient/grouping/` enthält gruppenspezifische Clientkonfigurationsdateien, aktualisierte Informationen zum Clientprodukt und hostbasierte Projektinformationen, die bei der Verbindung mit dem Host mit Push auf den Client übertragen werden. Die eigentliche Position ist in `pushtoclient.properties` angegeben (Variable `pushtoclient.folder` plus Suffix `/grouping`). Die darin enthaltenen Dateien werden von einem Clientadministrator von Developer for System z verwaltet.
- `/var/rdz/pushtoclient/projects/` enthält die hostbasierten Projektdefinitionsdateien. Die tatsächliche Position wird in der Datei `/var/rdz/pushtoclient/keymapping.xml` angegeben, die von einem Administrator der Developer for System z-Clients erstellt und verwaltet wird. Die enthaltenen Dateien werden von einem Projektleiter oder einem leitenden Entwickler verwaltet.
- `/var/rdz/pushtoclient/install/` enthält Konfigurationsdateien, durch die die Produktversion des Clients bei der Verbindung mit dem Host aktualisiert wird. Die tatsächliche Position wird in der Datei `/var/rdz/pushtoclient/keymapping.xml` angegeben, die von einem Administrator der Developer for System z-Clients erstellt und verwaltet wird. Die darin enthaltenen Dateien werden von einem Clientadministrator von verwaltet.
- `/var/rdz/pushtoclient/install/responsefiles/` enthält Konfigurationsdateien, durch die die Produktversion des Clients bei der Verbindung mit dem Host ak-

tualisiert wird. Die tatsächliche Position wird in der Datei `/var/rdz/pushtoclient/keymapping.xml` angegeben, die von einem Administrator der Developer for System z-Clients erstellt und verwaltet wird. Die darin enthaltenen Dateien werden von einem Clientadministrator von verwaltet.

Aktualisierungsberechtigungen für Benutzer ohne Systemadministratorrechte

Die im Verzeichnis `/var/rdz/pushtoclient/` enthaltenen Daten werden von Benutzern ohne Administratorrechte, beispielsweise von Projektmanagern, verwaltet. Diese Benutzer haben unter z/OS UNIX möglicherweise kaum Aktualisierungsberechtigungen. Daher ist es wichtig, zu verstehen, wie z/OS UNIX Zugriffsberechtigungen während der Dateierstellung festlegt, um sicherzustellen, dass Sie über eine betriebsfähige und gleichzeitig sichere Installation verfügen.

UNIX-Standards erfordern, dass Berechtigungen für drei Benutzertypen festgelegt werden können: Eigentümer, Gruppe und Sonstige. Lese-, Schreib- und Ausführungsberechtigungen können für jeden Typ individuell festgelegt werden.

z/OS UNIX legt die UID (Benutzer-ID) und die GID (Gruppen-ID) bei der Erstellung einer Datei auf die folgenden Werte fest:

- Als Benutzer-ID wird die wirksame Benutzer-ID des erstellenden Threads festgelegt.
- Als Gruppen-ID wird die Gruppen-ID des übergeordneten Verzeichnisses festgelegt. Wenn das Sicherheitsprofil `FILE.GROUPOWNER.SETGID` in der Klasse `UNIXPRIV` definiert ist, wird stattdessen standardmäßig die wirksame Gruppen-ID des erstellenden Threads verwendet. Weitere Informationen finden Sie in der Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

Jeder Standort kann seine eigene standardmäßige Zugriffsberechtigungsmaske festlegen, eine allgemeine Maske gewährt dem Eigentümer jedoch Lese- und Schreibberechtigung und der Gruppe und Sonstigen Leseberechtigung.

Daten in `/var/rdz/pushtoclient/` werden mithilfe der Zugriffsberechtigungsmaske erstellt, die in der Anweisung `file.permission` von `pushtoclient.properties` definiert ist. Der Standardwert gewährt dem Eigentümer und der Gruppe Lese- und Schreibberechtigung und Sonstigen Leseberechtigung. Alle verfügen über Ausführungsberechtigung. Die abschließenden Zugriffsberechtigungen sollten allen Benutzern Lese- und Ausführungsberechtigung und den Clientadministratoren von Developer for System z, die die Daten verwalten, Schreibberechtigung gewähren.

Die Daten im Verzeichnis `/var/rdz/pushtoclient/projects/` werden ohne bestimmte Zugriffsberechtigungsmaske erstellt. Die abschließenden Zugriffsberechtigungen sollten allen Benutzern Leseberechtigung und den Projektmanagern, die die Daten verwalten, Schreibberechtigung gewähren.

Nützliche Befehle für Sicherheitsfunktion

Um sicherzustellen, dass eine Gruppe von Projektmanagern oder Clientadministratoren von Developer for System z die Daten in diesen Verzeichnissen auch tatsächlich verwalten können, muss Ihr Sicherheitsadministrator möglicherweise eine Gruppe erstellen, die ein gültiges OMVS-Segment für sie aufweist. Diese Gruppe ist vorzugsweise die Standardgruppe für die einbezogenen Benutzer-IDs. Weitere Informationen zu den folgenden RACF-Musterbefehlen finden Sie in der Veröffentlichung *Security Server RACF Command Language Reference* (IBM Form SA22-7687):

```
ADDGROUP RDZPROJ OMVS(GID(1200))
CONNECT IBMUSER GROUP(RDZPROJ)
ALTUSER IBMUSER DFLTGRP(RDZPROJ)
```

Nützliche z/OS UNIX-Befehle

Informationen zu den folgenden z/OS UNIX-Beispielbefehlen finden Sie in der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802):

- Verwenden Sie den folgenden z/OS UNIX-Befehl **ls**, um alle Dateien innerhalb eines Verzeichnisses anzuzeigen.

```
ls -lR /var/rdz/pushtoclient/
```
- Verwenden Sie den folgenden z/OS UNIX-Befehl **chown**, um den Eigentümer eines Verzeichnisses sowie aller darin enthaltenen Dateien zu ändern.

```
chown -R IBMUSER /var/rdz/pushtoclient/
```
- Verwenden Sie den folgenden z/OS UNIX-Befehl **chgrp**, um die Gruppe dem Verzeichnis und allen darin enthaltenen Dateien zuzuordnen.

```
chgrp -R RDZPROJ /var/rdz/pushtoclient/
```
- Verwenden Sie den folgenden z/OS UNIX-Befehl **chmod**, um dem Eigentümer und der Gruppe Schreibberechtigung für das Verzeichnis und alle darin enthaltenen Dateien zu gewähren. Sonstige verfügen über Leseberechtigung. Alle verfügen über Ausführungsberechtigung.

```
chmod -R 775 /var/rdz/pushtoclient/
```

Beispielkonfiguration

Im folgenden Szenario erhalten alle Entwicklungsprojektmanager - ein aus drei Personen bestehendes Team - die Task, als Clientadministrator von Developer for System z zu fungieren.

Der Sicherheitsadministrator hat dem Team bereits eine Standardgruppe (RDZPROJ) mit eindeutiger Gruppen-ID (1200) zugeordnet. Ihre Benutzer-IDs verfügen in z/OS UNIX über keine besonderen Berechtigungen (wie UID 0). Der Sicherheitsadministrator hat das Profil FILE.GROUPOWNER.SETGID nicht definiert, sodass z/OS UNIX beim Erstellen neuer Dateien die Gruppen-ID des Verzeichnisses verwendet. Vom Systemprogrammierer wurde die Benutzer-ID IBMUSER (mit der UID 0 und der Standardgruppe SYS1) verwendet, um das Verzeichnis /var/rdz/pushtoclient zu erstellen.

1. Der Systemprogrammierer schränkt die Leseberechtigung für /var/rdz/pushtoclient auf den Eigentümer und die Gruppe ein:

```
# chmod 775 /var/rdz/pushtoclient
# ls -ld /var/rdz/pushtoclient
drwxrwxr-x  2 IBMUSER SYS1
/var/rdz/pushtoclient
```

Anmerkung: Der Job FEKSETUP, der während der Anpassungskonfiguration verwendet wurde, führt bereits diesen Schritt aus.

2. Der Systemprogrammierer macht RDZPROJ zur übergeordneten Gruppe:

```
# chgrp RDZPROJ /var/rdz/pushtoclient
# ls -ld /var/rdz/pushtoclient
drwxrwxr-x  2 IBMUSER RDZPROJ
/var/rdz/pushtoclient
```

Damit ist die Konfiguration abgeschlossen, die erforderlich ist, um Schreibberechtigungen für /var/rdz/pushtoclient auf den Systemprogrammierer (IBMUSER) und die Projektmanager (RDZPROJ) zu beschränken.

Kapitel 2. 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 die zugrunde liegenden Dateisysteme geschützt sind. Dies impliziert, dass die Programmbibliotheken und Konfigurationsdateien nur von vertrauenswürdigen Systemadministratoren aktualisiert werden können.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Authentifizierungsmethoden“ auf Seite 20
- „Verbindungssicherheit“ auf Seite 21
- „PassTickets verwenden“ auf Seite 23
- „Prüfprotokollierung“ auf Seite 24
- „JES-Sicherheit“ auf Seite 26
- „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30
- „Clientauthentifizierung unter Verwendung von X.509-Zertifikaten“ auf Seite 32
- „Eingangsport (POE) überprüfen“ auf Seite 36
- „Sonstige Informationen“ auf Seite 40
- „Clientfunktionen ändern“ auf Seite 36
- „Push-to-Client-Entwicklergruppen“ auf Seite 38
- „Debug-Sicherheit“ auf Seite 39
- „CICSTS-Sicherheit“ auf Seite 40
- „SCLM-Sicherheit“ auf Seite 41
- „Konfigurationsdateien für Developer for System z“ auf Seite 41
- „Sicherheitsdefinitionen“ auf Seite 44

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 Verbindungskonfiguration 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 1, „Wissenswertes zu Developer for System z“, auf Seite 3, 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

Anmerkung: Die vom Client bereitgestellten Authentifizierungsdaten werden nur einmalig, und zwar während der einleitenden Verbindungskonfiguration, verwendet. 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.

Anmerkung: Diese Authentifizierungsmethode wird nur von der Verbindungsmethode des RSE-Dämons unterstützt, wobei SSL (Secure Socket Layer) 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 authenti-

fiziert 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.

Debug Manager-Authentifizierung

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 Debug Manager.

Debug Manager muss für die Überprüfung von PassTickets berechtigt sein, damit eine Überprüfung durch Debug Manager für die vom RSE übermittelten Benutzer-IDs und PassTickets möglich ist. Das Lademodul AQEZPCM, das sich standardmäßig in der Ladebibliothek FEK.SFEKAUTH befindet, muss deshalb für APF-autorisiert sein.

Verbindungssicherheit

Verschiedene Ebenen der Kommunikationssicherheit werden vom RSE unterstützt. Dieser steuert die Kommunikation zwischen dem Client und den meisten 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 oder TLS 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.

Einige optionale Developer for System z-Services verwenden einen separaten, externen Kommunikationspfad (Client-Host):

- Die Integrated Debugger-Kommunikation kann mit TLS verschlüsselt werden.
- Die Application Deployment Manager-Kommunikation kann mit SSL verschlüsselt werden, wenn die Web-Service-Schnittstelle verwendet wird.

Developer for System z basiert auf Produkten eines Drittherstellers (z. B. TN3270-Server), um einige Services bereitzustellen. Informationen zu den Verbindungssicherheitsoptionen finden Sie in der entsprechenden Produktdokumentation.

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)" in *Hostkonfiguration* (SC12-4062).

Kommunikation mit SSL oder TLS verschlüsseln

Alle externen Datenströme von Developer for System z, die den RSE-Server passieren, können mit SSL (Secure Sockets Layer) oder TLS (Transport Layer Security) verschlüsselt werden. Die Verwendung der verschlüsselten Kommunikation wird von den Einstellungen in der Konfigurationsdatei `ssl.properties` gesteuert. Lesen Sie hierzu die Beschreibung im Abschnitt „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30. Die Variable `DSTORE_SSL_ALGORITHM` in der Anweisung `_RSE_JAVA_OPTS` von `rsed.envvars` ermöglicht Ihnen, zwischen SSL und dem Nachfolgeprotokoll TLS als Verschlüsselungsmethode zu wählen. Dies wird im Abschnitt "Zusätzliche Java-Startparameter mit `_RSE_JAVA_OPTS` definieren" im Handbuch *Hostkonfiguration* (IBM Form SC12-4062) dokumentiert.

Die Integrated Debugger-Engine des Clients wird mit dem Debug Manager des Hosts verbunden. Die Verwendung von SSL oder TLS wird von einer AT-TLS-Richtlinie (Application Transparent Transport Layer Security) gesteuert.

Der Host-Connect-Emulator auf dem Client stellt eine Verbindung zu einem TN3270-Server auf dem Host her. Die Verwendung von SSL oder TLS wird von TN3270 gesteuert. Diese Art der Verwendung ist im Handbuch *Communications Server IP Configuration Guide* (IBM Form SC31-8775) dokumentiert.

Ferne (hostbasierte) Aktionen in z/OS UNIX-Unterprojekten verwenden einen REXEC- oder SSH-Server auf dem Host. Die SSH-Kommunikation wird immer mit SSL verschlüsselt.

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 Handbuch *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 36.

Die Aktivierung des Eingangsports wirkt sich auch auf andere TCP/IP-Anwendungen aus, die die Überprüfung des Eingangsports unterstützen, wie z. B. auf IN-ETD.

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. Um die Sicherheit zu erhöhen, verarbeitet die Sicherheitssoftware von z/OS PassTickets standardmäßig als einmal verwendbare Kennwörter.

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.

Anmerkung: Ein ähnlicher Mechanismus wird verwendet, um sichere Verbindungen zu Debug Manager einzurichten.

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.

Die kleinste Einheit einer PassTicket-Zeitmarke ist 1 Sekunde. Das bedeutet, dass alle PassTickets, die innerhalb einer Sekunde von der gleichen Anwendung für dieselbe Benutzer-ID generiert werden, identisch sind. Dies sowie die Behandlung von PassTickets durch die Sicherheitssoftware von z/OS als nur einmal verwendbare Kennwörter stellt für Developer for System z ein Problem während der Anmeldung dar, da innerhalb einer Sekunde mehrere PassTickets erforderlich sind. Daher muss für Developer for System z in den Definitionen von PassTickets ein Flag (Markierung) gesetzt werden, das die generierten PassTickets als wiederverwendbar deklariert.

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" in *Hostkonfiguration* (SC12-4062).

- 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 "rsed.envvars (RSE-Konfigurationsdatei)" in *Hostkonfiguration* (SC12-4062) dokumentiert.

Mit dem Bedienerbefehl **modify switch** kann manuell zu einer neuen Prüfprotokolldatei gewechselt werden (siehe Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062)).

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 "Konsolennachrichten" im Handbuch *Hostkonfiguration* (SC12-4062).

Prüfprozesse

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.

Mit den `audit.action*`-Direktiven in `rsed.envvars` können Sie einen Benutzerexit (z/OS UNIX-Shell-Script, z/OS UNIX-REXX oder z/OS UNIX-Programm) angeben, der beim Schließen eines Prüfprotokolls von RSE aufgerufen wird. Dieser Benutzerexit kann dann den Inhalt des Prüfprotokolls verarbeiten.

Prüfprotokolldateien haben die Berechtigungsbitmaske 640 (-rw-r-----), sofern dies nicht in `rsed.envvars` durch die Direktive `audit.log.mode` geändert wurde. Dies bedeutet, dass der Eigner (z/OS UNIX-Benutzer-ID des RSE-Dämons) Lese- und Schreibzugriff auf die Dateien und die Standardgruppe des Eigners Lesezugriff hat. Alle anderen Zugriffsversuche werden zurückgewiesen, sofern sie nicht von einem Superuser (UID 0) oder einer Person mit entsprechenden Berechtigungen für das Profil `SUPERUSER.FILESYS` der Sicherheitsklasse `UNIXPRIV` unternommen werden.

Prüfdaten

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)
- Dateizugriff (READ, WRITE, CREATE, DELETE, RENAME)
- Ausführung von TSO-Befehlen und z/OS UNIX-Befehlen

Jede protokollierte Aktion wird (mit einer Datums-/Zeitmarke) im CSV-Format (Comma Separated Value) gespeichert, das von Automatisierungs- oder Datenanalysetools gelesen werden kann. Beispiel:

```
yyyy/mm/dd hh:mm:ss.sss,userid,action,dataset_name[,returncode] [,additional_information]]
```

Statistikdaten zu Dateien und Mitgliedern werden auch protokolliert, wenn die Datei geöffnet wird. Sie werden an die Zeile angehängt, die den Abschluss der Aktion READ dokumentiert; die Felder sind mit %n begrenzt. Beispiel:

```
yyyy/mm/dd hh:mm:ss.sss,userid,action,dataset_name,returncode,create%modify%n...
```

Die folgenden Attribute werden protokolliert, in der aufgeführten Reihenfolge:

- Erstellungsdatum und -uhrzeit (mm/dd/yyyy hh:mm)
- Datum und Uhrzeit der letzten Änderung (mm/dd/yyyy hh:mm:ss)
- Datum und Uhrzeit des letzten Zugriffs (mm/dd/yyyy hh:mm:ss)

- Satzformat (RECFM)
- SCLM-Überarbeitungsindikator (N = Überarbeitungsnummer ist festgelegt, D = Überarbeitungsnummer ist nicht festgelegt)
- SCLM-Überarbeitungsnummer
- Ungültige Hexadezimalzeichen eingeschlossen (Y = Ja, N = Nein)

Anmerkung: Für ungültige Hexadezimalzeichen sind Developer for System z Zuordnungsservices erforderlich, da mit ihnen aufgrund von Codepageabweichungen keine Übertragungen vom Client und zurück möglich sind.

- Länge eines logischen Satzes (LRECL)
- Dateigröße
- Für zukünftige Verwendung reserviert
- Für zukünftige Verwendung reserviert
- Benutzer-ID
- Eigner für diese Datei oder dieses Member sperren (einreihen)
- Host-Code-Punkte für CR (Wagenrücklauf), LF (Zeilenvorschub) und NL (neue Zeile) und ihre Ersetzungszeichen (nur verfügbar, wenn ein Client der Version 8.0.3 oder höher verwendet wird)

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 1. 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

Tabelle 1. JES Job Monitor, Konsolbefehle (Forts.)

Aktion	JES2	JES3
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 1 auf Seite 26 aufgelistet sind, sind standardmäßig auf Jobs beschränkt, deren Eigner der Benutzer ist. Dies kann mit der Anweisung `LIMIT_COMMANDS` geändert werden (siehe Abschnitt "FEJJCNFG (JES Job Monitor-Konfigurationsdatei)" im Handbuch *Hostkonfiguration* (SC12-4062)).

Tabelle 2. 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 3. Erweiterte JESSPOOL-Profil

Befehl	JESSPOOL-Profil	Erforderlicher Zugriff
Hold	nodeid.userid.jobname.jobid	ALTER
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 "Matrixtabelle LIMIT_COMMANDS mit Befehlsberechtigungen" in "FEJJC�FG (JES Job Monitor-Konfigurationsdatei)" in *Hostkonfiguration* (SC12-4062)). 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 Beschränkungen in der nachfolgenden Liste 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 "FEJJC�FG (JES Job Monitor-Konfigurationsdatei)" in *Hostkonfiguration* (SC12-4062) dokumentiert.

Mit JES Job Monitor können Sie definieren, wieviel Berechtigung der EMCS-Konsole gemäß der Richtlinie LIMIT_CONSOLE gewährt wird, wie in "FEJJC�FG, JES Job Monitor-Konfigurationsdatei" im Handbuch *Hostkonfiguration* (SC12-4062) dokumentiert.

Tabelle 4. LIMIT_CONSOLE (Berechtigungsmatrix für Konsole)

LIMIT_CONSOLE	Aktives Profil in der Klasse OPERCMDS	Kein aktives Profil in der Klasse OPERCMDS
LIMITED (Standardwert)	Zulässig, wenn durch Sicherheitsprofil zugelassen	Nicht zulässig
NOLIMIT	Zulässig, wenn durch Sicherheitsprofil zugelassen	Zulässig

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. JES lässt den Befehl fehlschlagen, wenn LIMIT_CONSOLE=LIMITED in FEJJC�FG definiert 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. Dies kann mit der Anweisung `LIMIT_VIEW` geändert werden (siehe Abschnitt "FEJJCNFG (JES Job Monitor-Konfigurationsdatei)" im Handbuch *Hostkonfiguration* (SC12-4062)).

Tabelle 5. 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 (FEJJCNFG) 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/TLS verschlüsselte Kommunikation

Die externe Kommunikation (Client-Host) mit RSE kann mit SSL (Secure Socket Layer) oder TLS (Transport Layer Security) verschlüsselt werden. Dieses Feature ist standardmäßig inaktiviert und wird von den Einstellungen in `ssl.properties` gesteuert. Weitere Informationen hierzu finden Sie im Abschnitt "ssl.properties, RSE-SSL-Verschlüsselung (optional)" im Handbuch *Hostkonfiguration* (SC12-4062).

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 6. Mechanismen für den SSL-Zertifikatsspeicher

Zertifikatsspeicher	Erstellt und verwaltet von	RSE-Dämon	RSE-Server
Schlüsseldatei	SAF-konformes Sicherheitsprodukt	unterstützt	unterstützt
Schlüsseldatenbank	z/OS UNIX gskkyman	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. Weitere Informationen zu ICSF und zur Steuerung der Verwendung von Verschlüsselungsschlüsseln und Verschlüsselungsservices finden Sie im Handbuch *Cryptographic Services ICSF Administrator's Guide* (IBM Form SA22-7521).

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 folgenden 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

Die Variable DSTORE_SSL_ALGORITHM in der Anweisung _RSE_JAVA_OPTS von rsed.envvars ermöglicht Ihnen, zwischen SSL Nachfolgeprotokoll TLS als Verschlüsselungsmethode zu wählen. Dies wird im Abschnitt "Zusätzliche Java-Startparameter mit _RSE_JAVA_OPTS definieren" im Handbuch *Hostkonfiguration* (IBM Form SC12-4062) dokumentiert.

Weitere Details zur Aktivierung von SSL für Developer for System z finden Sie in Kapitel 13, „SSL- und X.509-Authentifizierung konfigurieren“, auf Seite 195.

Mit Integrated Debugger verschlüsselte Kommunikation

Die externe Kommunikation (Client-Host) mit dem optionalen Debug Manager kann mit SSL oder TLS verschlüsselt werden. Um die Verschlüsselung auf diese Weise durchzuführen, erstellen Sie eine AT-TLS-Richtlinie für den Port, der von Debug Manager für die externe Kommunikation verwendet wird (Standardport 5335). Sie finden eine Beispielrichtlinie unter Abb. 9 auf Seite 32. Weitere Informationen zur Einrichtung von AT-TLS (Application Transparent TLS) finden Sie im Handbuch *Communications Server IP Configuration Guide* (IBM Form SC31-8775).

```

| TLSRule                                RDz_Debug_Manager
| {
|   LocalPortRange                        5335
|   Direction                            Inbound
|   TLSGroupActionRef                    grp_Production
|   TLSEnvironmentActionRef              RDz_Debug_Manager
| }
| TLSEnvironmentAction                  RDz_Debug_Manager
| {
|   HandshakeRole Server
|   TLSKeyRingParms
|   {
|     Keyring dbgmgr.racf                # Keyring must be owned by the Debug Manager
|   }
| }
| TLSGroupAction                        grp_Production
| {
|   TTLEnabled                           On
|   Trace                                2
| }

```

Abbildung 9. AT-TLS-Richtlinie für Debug Manager

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, auf dem der RSE-Dämon aktiv ist, verwendbar 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 HostIdMappings-Erweiterung im Zertifikat basiert. Weitere Informationen hierzu enthält „Authentifizierung durch Ihre Sicherheitssoftware“ auf Seite 34.

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.

- `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 Domänennamen, 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 dazu auf Ihr Sicherheitsprodukt zurückzugreifen. Für die Authentifizierung 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 34 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 33.

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)
- SETROPTS 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_JAVAOPTS` definieren" in *Hostkonfiguration* (SC12-4062)).
- 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.POE 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).

Clientfunktionen ändern

Developer for System z-Clients der Version 8.5.1 und höher können die Zugriffsberechtigung auf SAF-Sicherheitsprofile überprüfen und auf der Basis des Ergebnisses die zugehörige Funktion für den Benutzer aktivieren oder inaktivieren.

Developer for System z überprüft Zugriffszulassungen auf die Profile, die in Tabelle 7 auf Seite 37 aufgeführt sind, um festzustellen, welche Optionen für den Benutzer aktiviert oder inaktiviert werden sollen.

Tabelle 7. SAF-Informationen zur Änderung von Clientfunktionen

FACILITY-Profil	Feste Länge	Erforderlicher Zugriff	Ergebnis
FEK.USR.OFF.REMOTECOPY.MVS.sysname	27	READ	Client inaktiviert Kopierfunktionen und zugehörige Funktionen für MVS-Dateien

Anmerkung: Developer for System z geht davon aus, dass ein Benutzer über keine Zugriffsberechtigung verfügt, wenn die Sicherheitssoftware nicht feststellen kann, ob der Benutzer über die Berechtigung für den Zugriff auf ein Profil verfügt. Dies ist zum Beispiel der Fall, wenn das Profil gar nicht definiert ist.

Der Wert von sysname ist der Systemname des Zielsystems.

In der Spalte "Feste Länge" ist die Länge des festen Teils des zugehörigen Sicherheitsprofils angegeben.

Developer for System z erwartet standardmäßig, dass FEK.*-Profile der Sicherheitsklasse FACILITY angehören. Für Profile der Klasse FACILITY gilt eine Beschränkung auf 39 Zeichen. Falls die Länge des festen Profiltails (FEK.USR.<key>) und die Länge des standortspezifischen Profiltails (sysname) in der Summe diesen Wert überschreiten, können Sie die Profile in einer anderen Klasse speichern und Developer for System z dazu anweisen, diese Klasse zu verwenden. Dazu müssen Sie in rsed.envvars das Kommentarzeichen vor _RSE_FEK_SAF_CLASS entfernen und den Namen der gewünschten Klasse angeben.

Die folgenden Beispielsicherheitsdefinitionen ermöglichen die Aktion REMOTECOPY.MVS für alle Benutzer von CDFMVS08, mit Ausnahme derer in der Gruppe RESTRICT:

```
RDEFINE FACILITY (FEK.USR.OFF.REMOTECOPY.MVS.CDFMVS08) -
  UACC(NONE) DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CLIENT CONTROL')
PERMIT FEK.USR.OFF.REMOTECOPY.MVS.CDFMVS08 CLASS(FACILITY) -
  ID(RESTRICT) ACCESS(READ) SETROPTS RACLIST(FACILITY) REFRESH
```

OFF.REMOTECOPY.MVS

Wenn Benutzer über Lesezugriff auf das Profil FEK.USR.OFF.REMOTECOPY.MVS.sysname verfügen, inaktivieren ihre Developer for System z-Clients der Version 8.5.1 und höher Aktionen vom Typ 'Ziehen', 'Kopieren', 'Speichern unter' und 'Offline arbeiten' für MVS-Dateien. Daraus folgt, dass die Benutzer zwar auf die Dateien auf diesem System zugreifen können, sie aber keine lokale Kopie einer Datei auf ihren Workstations erstellen können. Dadurch wird Weitergabe von vertraulichen Informationen verhindert, falls die lokale Workstation verlorengeht oder gestohlen wird.

Push-to-Client-Entwicklergruppen

Developer for System z-Clients ab Version 8.0.1 können beim Verbindungsaufbau Konfigurationsdateien und Produktaktualisierungsdaten im Pull-Verfahren vom Host abrufen. Dadurch wird sichergestellt, dass alle Clients über die gleichen Einstellungen verfügen und auf dem neuesten Stand sind.

Ab Version 8.0.3 kann der Clientadministrator mehrere Clientkonfigurationssätze und Clientaktualisierungsszenarien für die Anforderungen verschiedener Entwicklergruppen erstellen. Dadurch erhalten Benutzer eine angepasste Konfiguration, die auf Kriterien wie LDAP-Gruppenzugehörigkeit oder Zulassung zu einem Sicherheitsprofil basiert.

Wenn Definitionen Ihrer Sicherheitsdatenbank als Auswahlmechanismus verwendet werden (der SAF-Wert wird für Direktiven in `pushtoclient.properties` angegeben), überprüft Developer for System z die Zugriffserlaubnis auf die in Tabelle 8 aufgelisteten Profile, um festzustellen, zu welchen Entwicklergruppen ein Benutzer gehört und ob der Benutzer Aktualisierungen zurückweisen darf.

Tabelle 8. Push-to-Client-relevante SAF-Informationen

FACILITY-Profil	Feste Länge	Erforderlicher Zugriff	Ergebnis
FEK.PTC.CONFIG.ENABLED. sysname.devgroup	23	READ	Der Client akzeptiert Konfigurationsaktualisierungen für die angegebene Gruppe.
FEK.PTC.PRODUCT. ENABLED.sysname.devgroup	24	READ	Der Client akzeptiert Produktaktualisierungen für die angegebene Gruppe.
FEK.PTC.REJECT.CONFIG. UPDATES.sysname	30	READ	Der Benutzer kann Konfigurationsaktualisierungen zurückweisen.
FEK.PTC.REJECT.PRODUCT. UPDATES.sysname	31	READ	Der Benutzer kann Produktaktualisierungen zurückweisen.

Anmerkung: Developer for System z geht davon aus, dass ein Benutzer über keine Zugriffsberechtigung verfügt, wenn die Sicherheitssoftware nicht feststellen kann, ob der Benutzer über die Berechtigung für den Zugriff auf ein Profil verfügt. Dies ist zum Beispiel der Fall, wenn das Profil gar nicht definiert ist.

Der Wert von `devgroup` ist der Gruppenname, der einer Entwicklergruppe zugewiesen ist. Dieser Gruppenname ist auf Developer for System z-Clients sichtbar.

Der Wert von `sysname` ist der Systemname des Zielsystems.

In der Spalte 'Feste Länge' ist die Länge des festen Teils des zugehörigen Sicherheitsprofils angegeben.

Developer for System z erwartet standardmäßig, dass FEK.*-Profile der Sicherheitsklasse FACILITY angehören. Für Profile der Klasse FACILITY gilt eine Beschränkung auf 39 Zeichen. Falls die Länge des festen Profiteils (FEK.PTC.<key>) und die Länge des standortspezifischen Profiteils (sysname oder sysname.devgroup) in der Summe diesen Wert überschreiten, können Sie die Profile in einer anderen Klasse speichern und Developer for System z dazu anweisen, diese Klasse zu verwenden. Dazu müssen Sie in rsed.envvars das Kommentarzeichen vor _RSE_FEK_SAF_CLASS entfernen und den Namen der gewünschten Klasse angeben.

Nur Clientadministratoren, die in der Zugriffsliste der FEK.PTC.*.ENABLED.*-Profile aufgeführt sind, können die zugehörigen Push-to-Client-Metadaten definieren und verwalten. Dies bedeutet, dass in den Profilen zumindest der Clientadministrator in der Zugriffsliste definiert sein muss, damit Push-to-Client mit Gruppenunterstützung implementiert werden kann.

Weitere Informationen zur Aktivierung der Unterstützung für mehrere Gruppen finden Sie im Abschnitt '(Optional) pushtoclient.properties, Hostbasierte Clientsteuerung' im Handbuch *Hostkonfiguration* (IBM Form SC12-4062). Weitere Informationen zu den Konzepten und der Implementierung von Push-to-Client finden Sie im Abschnitt Kapitel 7, „Push-to-Client-Aspekte“, auf Seite 127.

Debug-Sicherheit

Mit dem optionalen Integrated Debugger kann ein Debugging von CICS-Transaktionen, die in den Nur-Lese-Speicher geladen wurden, durchgeführt werden. Bei der Verwendung für das Debugging einer Transaktion im Fehlerstatus (nicht autorisiert), überprüft Integrated Debugger, ob der Benutzer der Debugsitzung dies durchführen darf.

Developer for System z überprüft den Zugriff auf die Profile, die in Tabelle 9 aufgeführt sind, um festzustellen, welche Debugberechtigungen erteilt wurden.

Tabelle 9. SAF-Informationen für Debugfunktionen

FACILITY-Profil	Erforderlicher Zugriff	Ergebnis
AQE.AUTHDEBUG.WRITEBUFFER	UPDATE	Benutzer kann ein Debugging für schreibgeschützte CICS-Transaktionen durchführen

Anmerkung: Developer for System z geht davon aus, dass ein Benutzer über keine Zugriffsberechtigung verfügt, wenn die Sicherheitssoftware nicht feststellen kann, ob der Benutzer über die Berechtigung für den Zugriff auf ein Profil verfügt. Dies ist zum Beispiel der Fall, wenn das Profil gar nicht definiert ist.

Die folgenden Beispielsicherheitsdefinitionen ermöglichen die Aktion AUTHDEBUG.WRITEBUFFER für alle Benutzer in der Gruppe RDZDEBUG:

```
RDEFINE FACILITY (AQE.AUTHDEBUG.WRITEBUFFER) -
  UACC(NONE) DATA('RATIONAL DEVELOPER FOR SYSTEM Z – DEBUG CICS READ-ONLY')
PERMIT AQE.AUTHDEBUG.WRITEBUFFER CLASS(FACILITY) -
  ID(RDZDEBUG) ACCESS(UPDATE)
SETROPTS RACLIST(FACILITY) REFRESH
```

CICSTS-Sicherheit

Mit dem optionalen Integrated Debugger kann ein Debugging von CICS-Transaktionen durchgeführt werden. Ausführliche Informationen hierzu enthält der Abschnitt „CICS-Transaktionsdebugging“ auf Seite 157.

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 8, „CICSTS-Aspekte“, auf Seite 145.

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.

Sonstige Informationen

GATE-Überlastung (Thrashing)

Wenn ein Adressraum RACF erstmals anweist, auf eine Ressourcenklasse zuzugreifen, die sich nicht in RACLIST befindet, also nicht im Speicher gespeichert ist, wie es beispielsweise bei der Klasse DATASET der Fall ist, so ruft RACF alle zugehörigen generischen Profile des Benutzers ab und speichert diese im Adressraum in einer als GATE (Generic Anchor Table Entry) bezeichneten Liste. Bis z/OS 1.12 pflegt RACF vier generische Anker für jeden Adressraum und vier für jeden MVS-TCB (Task Control Block), der über ein ACEE (Accessor Environment Element) verfügt. Sind alle vier belegt, ersetzt RACF denjenigen, der die längste Zeit nicht referenziert wurde, sobald ein neuer eintrifft.

Wenn Ihre Benutzer häufig auf mehr als vier übergeordnete Qualifikationsmerkmale für Daten zugreifen, tritt in den RSE-Thread-Pools (die mehrere Benutzer unter Verwendung von Threads mit benutzerspezifischen ACEEs bedienen) unter Umständen eine GATE-Überlastung (Thrashing) auf, denn RACF muss neue Einträge anhand der verfügbaren Ankerslots rollieren.

Mit z/OS 1.12 hat RACF die Option **GENERICANCHOR** des Befehls **SET** eingeführt, die Ihnen ermöglicht, die Tabelle zu vergrößern. Diese Vergrößerung kann systemweit definiert oder für jeden Jobnamen festgelegt werden.

Verwaltetes ACEE

Developer for System z verwendet z/OS UNIX-Kernelservices wie `pthread_security_np()` und `__passwd()`, die ihrerseits auf den Sicherheitsservice `InitACEE` zurückgreifen. Dadurch ergeben sich verwaltete ACEE-Sicherheitssteuerungsblöcke. Ein verwaltetes ACEE (Accessor Environment Element) wird von Ihrem Sicherheitsprodukt zwischengespeichert. Solange das Zeitlimit des Cache nicht abläuft, werden bestimmte Änderungen (beispielsweise Kennwortänderungen außerhalb von Developer for System z) von Ihrem Sicherheitsprodukt ignoriert. (Das Zeitlimit kann einige Minuten betragen.)

Nach Sicherheitsänderungen muss der verwaltete ACEE-Cache daher aktualisiert werden, damit sichergestellt wird, dass Developer for System z die neuen Daten verwendet.

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 für Developer for System z

Es gibt mehrere Konfigurationsdateien für Developer for System z, deren Anweisungen Auswirkungen auf die Sicherheits- und Prüfkongfiguration 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 26.

- `LIMIT_CONSOLE={LIMITED | NOLIMIT}`

Definieren der Berechtigungsstufe der zum Ausführen von Aktionen verwendeten EMCS-Konsole. Weitere Informationen enthält der Abschnitt „Aktionen für Beschränkungen der Jobziele“ auf Seite 26.

- `LIMIT_VIEW={USERID | NOLIMIT}`

Definieren, welche Spooldateien durchsucht werden können. Weitere Informationen enthält der Abschnitt „Zugriff auf Spooldateien“ auf Seite 29.

- `LOOPBACK_ONLY={ON | OFF}`
Definieren, ob der Zugriff auf JES Job Monitor auch außerhalb dieses z/OS-Systems möglich ist. Weitere Informationen enthält der Abschnitt *FEJJCENFG (Konfigurationsdatei für JES Job Monitor)* im Kapitel *Basisanpassung* des Handbuchs *Hostkonfiguration* (IBM Form SC12-4062).
- `APPLID={FEKAPPL | *}`
Die Anwendungs-ID, die zum Erstellen/Überprüfen von PassTicket verwendet wird. Weitere Informationen enthält der Abschnitt „PassTickets verwenden“ auf Seite 23.

Anmerkung: Details zu diesen und anderen FEJJCENFG-Anweisungen finden Sie im Abschnitt "FEJJCENFG (JES Job Monitor-Konfigurationsdatei)" in *Hostkonfiguration* (SC12-4062).

RSE - rsed.envvars

- `_RSE_FEK_SAF_CLASS={FACILITY | *}`
Sicherheitsklassen mit FEK.**-Profilen. Weitere Informationen finden Sie im Abschnitt „Push-to-Client-Entwicklergruppen“ auf Seite 38 und im Abschnitt „Clientfunktionen ändern“ auf Seite 36.
- `(_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" in *Hostkonfiguration* (SC12-4062).
- `(_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" in *Hostkonfiguration* (SC12-4062).
- `(_RSE_JAVAOPTS) -DAPPLID={FEKAPPL | *}`
Die Anwendungs-ID, die zum Erstellen/Überprüfen von PassTicket verwendet wird. Weitere Informationen enthält der Abschnitt „PassTickets verwenden“ auf Seite 23.
- `(_RSE_JAVAOPTS) -Denable.port.of.entry={true | false}`
Aktivieren der Überprüfung des Eingangsports. Weitere Informationen enthält der Abschnitt „Eingangsport (POE) überprüfen“ auf Seite 36.
- `(_RSE_JAVAOPTS) -DDSTORE_SSL_ALGORITHM={TLSv1.2 | SSL}`
Wählen Sie SSL oder TLS als Verfahren für die Verschlüsselung der Kommunikation aus. Weitere Informationen enthält der Abschnitt „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30.
- `(_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 32.
- `GSK_CRL_SECURITY_LEVEL={LOW | MEDIUM | HIGH}`
`GSK_LDAP_SERVER=*`
`GSK_LDAP_PORT={389 | *}`
`GSK_LDAP_USER=*`
`GSK_LDAP_PASSWORD=*`
Zusätzliche Sicherheitsprüfungen für die Authentifizierung von X.509. Weitere Informationen enthält der Abschnitt „Zertifikatswiderrufsliste (CRL) abfragen (optional)“ auf Seite 33.
- `(_RSE_JAVAOPTS) -Ddaemon.log={/var/rdz/logs | *}`

Position der Prüfprotokolldateien. Weitere Informationen enthält der Abschnitt „Prüfprotokollierung“ auf Seite 24.

- (`_RSE_JVAOPTS`) `-Daudit.log.mode={RW.R. | * }`

Maske für Dateizugriffsberechtigungen der Prüfprotokolldateien. Weitere Informationen enthält der Abschnitt „Prüfprotokollierung“ auf Seite 24.

- (`_RSE_JVAOPTS`) `-Daudit.action=<Shell-Script>`

(`_RSE_JVAOPTS`) `-Daudit.action.id=<Benutzer-ID>`

z/OS UNIX-basierte Benutzerexit, der Prüfprotokolle verarbeitet. Weitere Informationen enthält der Abschnitt „Prüfprotokollierung“ auf Seite 24.

Anmerkung: Details zu diesen und anderen `rsed.envvars`-Anweisungen finden Sie im Abschnitt "rsed.envvars (RSE-Konfigurationsdatei)" in *Hostkonfiguration* (SC12-4062).

RSE - ssl.properties

- `daemon_keydb_file={SAF-Schlüsseldateiname | gskkyman-Schlüsseldatenbankname}`

Position des RSE-Dämonzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30.

- `daemon_key_label=Zertifikatsbezeichnung`

Name des RSE-Dämonzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30.

- `server_keystore_file={SAF-Schlüsseldateiname | Java-Schlüsseldateiname}`

Position des RSE-Serverzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30.

- `server_keystore_label=Zertifikatsbezeichnung`

Name des RSE-Serverzertifikats. Weitere Informationen enthält der Abschnitt „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30.

- `server_keystore_type={JKS | JCRACFKS | JCECCARACFKS}`

Verwendeter Keystoretyp (Java-Keystore oder SAF-Schlüsseldatei). Weitere Informationen enthält der Abschnitt „Mit SSL/TLS verschlüsselte Kommunikation“ auf Seite 30.

Anmerkung: Details zu diesen und anderen `ssl.properties`-Anweisungen finden Sie im Abschnitt "ssl.properties, RSE-SSL-Verschlüsselung (optional)" in *Hostkonfiguration* (SC12-4062).

RSE - pushtoclient.properties

- `config.enabled={true | false | SAF | LDAP}`

`reject.config.updates={true | false | SAF | LDAP}`

Hostbasierte Steuerung der Clientkonfigurationsdateien für Developer for System z. Weitere Informationen enthält der Abschnitt Kapitel 7, „Push-to-Client-Aspekte“, auf Seite 127.

- `product.enabled={true | false | SAF | LDAP}`

`reject.product.updates={true | false | SAF | LDAP}`

Hostbasierte Steuerung der Clientproduktaktualisierungen für Developer for System z. Weitere Informationen enthält der Abschnitt Kapitel 7, „Push-to-Client-Aspekte“, auf Seite 127.

Anmerkung: Einzelheiten zu diesen und weiteren `pushtoclient.properties`-Anweisungen finden Sie im Abschnitt "(Optional) `pushtoclient.properties`, Hostbasierte Clientsteuerung" im Handbuch *Hostkonfiguration* (IBM Form SC23-7658).

Sicherheitsdefinitionen

Passen Sie das Beispielmember 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 Informationen finden Sie in 'Anpassungskonfiguration' im Handbuch *IBM Rational Developer for System z-Hostkonfiguration*.

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

Anmerkung:

- Rufen Sie für Sites, die CA ACF2™ for z/OS verwenden, die Seite Ihres Produkts auf der CA-Unterstützungssite (<https://support.ca.com>) auf und überprüfen Sie den Inhalt des zugehörigen Dokuments (TEC492389) 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.
- Rufen Sie für Sites, die CA Top Secret® for z/OS verwenden, die Seite Ihres Produkts auf der CA-Unterstützungssite (<https://support.ca.com>) auf und überprüfen Sie den Inhalt des zugehörigen Dokuments (TEC492091) 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 werden die erforderlichen Schritte, die optionale Konfiguration und mögliche Alternativen beschrieben.

Anforderungen und Prüfliste

Der Sicherheitsadministrator muss die in Tabelle 10 aufgelisteten Werte kennen, um die Sicherheitskonfiguration abschließen zu können. Diese Werte wurden in früheren Schritten der Installation und Anpassung von Developer for System z definiert.

Tabelle 10. Variablen für die Sicherheitskonfiguration

Beschreibung	<ul style="list-style-type: none">• Standardwert• Entsprechende Quelle	Wert
Übergeordnetes Qualifikationsmerkmal für Developer for System z-Produkt	<ul style="list-style-type: none">• FEK• SMP/E-Installation	

Tabelle 10. Variablen für die Sicherheitskonfiguration (Forts.)

Beschreibung	<ul style="list-style-type: none"> • Standardwert • Entsprechende Quelle 	Wert
Übergeordnetes Qualifikationsmerkmal für Developer for System z-Anpassung	<ul style="list-style-type: none"> • FEK.#CUST • FEK.SFEKSAMP(FEKSETUP), wie in 'Anpassungskonfiguration' im Handbuch <i>IBM Rational Developer for System z-Hostkonfiguration</i> beschrieben. 	
Name der gestarteten Task von Integrated Debugger	<ul style="list-style-type: none"> • DBGMGR • FEK.#CUST.PROCLIB(DBGMGR), wie in 'PROCLIB-Änderungen' im Handbuch <i>IBM Rational Developer for System z-Hostkonfiguration</i> beschrieben. 	
Name der gestarteten Task von JES Job Monitor	<ul style="list-style-type: none"> • JMON • FEK.#CUST.PROCLIB(JMON), wie in 'PROCLIB-Änderungen' im Handbuch <i>IBM Rational Developer for System z-Hostkonfiguration</i> beschrieben. 	
Name der gestarteten Task des RSE-Dämons	<ul style="list-style-type: none"> • RSED • FEK.#CUST.PROCLIB(RSED), wie in 'PROCLIB-Änderungen' im Handbuch <i>IBM Rational Developer for System z-Hostkonfiguration</i> beschrieben. 	
Anwendungs-ID	<ul style="list-style-type: none"> • FEKAPPL • /etc/rdz/rsed.envvars, wie in "Zusätzliche Java-Startparameter mit _RSE_JVAOPTS definieren" im Handbuch <i>IBM Rational Developer for System z-Hostkonfiguration</i> beschrieben. 	

Die folgende Liste liefert einen Überblick über die Aktionen, die zur vollständigen Durchführung der Basissicherheitskonfiguration von Developer for System z erforderlich sind. Wie in den nachfolgenden Abschnitten dokumentiert ist, können je nach der erforderlichen Sicherheitsebene verschiedene Methoden angewendet werden, um diese Anforderungen zu erfüllen. Informationen zur Sicherheitskonfiguration optionaler Developer for System z-Services, enthaltenen die vorherigen Abschnitte.

- „Sicherheitseinstellungen und -klassen aktivieren“ auf Seite 46

- „OMVS-Segment für Benutzer von Developer for System z definieren“ auf Seite 47
- „Gestartete Tasks für Developer for System z definieren“ auf Seite 47
- „RSE als sicheren z/OS UNIX-Server definieren“ auf Seite 49
- „Programmgesteuerte MVS-Bibliotheken für RSE definieren“ auf Seite 49
- „PassTicket-Unterstützung für RSE definieren“ auf Seite 50
- „Anwendungsschutz für RSE definieren“ auf Seite 51
- „JES-Befehlssicherheit definieren“ auf Seite 52
- „Dateiprofile definieren“ auf Seite 54
- „Programmgesteuerte z/OS UNIX-Dateien für RSE definieren“ auf Seite 58
- „Sicherheitseinstellungen prüfen“ auf Seite 59

Sicherheitseinstellungen und -klassen aktivieren

Developer for System z verwendet eine Vielzahl von Sicherheitsmechanismen, um für den Client eine geschützte und kontrollierte Hostsystemumgebung bereitzustellen. Dazu 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 gesicherten Anmeldung unter Verwendung von PassTickets für RSE
 - SETROPTS GENERIC(PTKTDATA)
 - SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
- Aktivieren der Programmsteuerung, um sicherzustellen, dass RSE nur gesicherten Code laden kann
 - RDEFINE PROGRAM ** ADDMEM('SYS1.COMDLIB'//NOPADCHK) UACC(READ)
 - SETROPTS WHEN(PROGRAM)

Anmerkung: Wenn die Klasse PROGRAM bereits ein Profil * enthält, sollten Sie das Profil ** nicht erstellen. Es verkompliziert den von der Sicherheitssoftware verwendeten Suchpfad und macht ihn teilweise unkenntlich. Führen Sie

in einem solchen Fall die vorhandenen Definitionen aus dem Profil * mit den neuen Definitionen des Profils ** zusammen. Verwenden Sie das Profil **, wie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683) dokumentiert.

Achtung: Wenn "WHEN PROGRAM" aktiv ist, müssen einige Produkte (beispielsweise FTP) programmgesteuert sein. Testen Sie eine solche Programmsteuerung, bevor Sie sie auf einem Produktionssystem aktivieren.

- (Optional) Unterstützung für X.509-HostIdMappings und erweiterten Eingangsport (POE) aktivieren
 - SETROPTS GENERIC(SERVAUTH)
 - SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)

OMVS-Segment für Benutzer von Developer for System z definieren

Für jeden Benutzer von Developer for System z muss ein RACF-OMVS-Segment oder 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 eines jeden Benutzers ist ebenfalls ein OMVS-Segment mit einer Gruppen-ID erforderlich.

Wenn Integrated Debugger (optional) verwendet wird, benötigen die Benutzer-ID, unter der die Anwendung, für die ein Debugging ausgeführt ist, aktiv ist, sowie die Standardgruppe auch ein gültiges RACF OMVS-Segment oder ähnliches.

Ersetzen Sie in den folgenden RACF-Beispielbefehlen die Platzhalter #userid, #user-identifizier, #group-name und #group-identifizier durch tatsächliche Werte:

- ```
ALTUSER #userid
OMVS(UID(#user-identifizier) HOME(/u/#userid) PROGRAM(/bin/sh) NOASSIZEMAX)
```
- ```
ALTGROUP #group-name OMVS(GID(#group-identifizier))
```

Gestartete Tasks für Developer for System z definieren

Die folgenden RACF-Beispielbefehle erstellen die gestarteten Tasks DBGMR, JMON und RSED mit der ihnen jeweils zugeordneten geschützten Benutzer-ID (STCDBGMR, STCJMON beziehungsweise STCRSE) und der Gruppe STCGROUP. Ersetzen Sie die Platzhalter #group-id und #user-id-* durch gültige OMVS-IDs.

- ```
ADDGROUP STCGROUP OMVS(GID(#group-id))
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
```
- ```
ADDUSER STCDBGMR DFLTGRP(STCGROUP) NOPASSWORD
NAME('RDZ - DEBUG MANAGER')
OMVS(UID(#user-id-debug) HOME(/tmp) PROGRAM(/bin/sh) )
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
ADDUSER STCJMON DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - JES JOBMONITOR')
OMVS(UID(#user-id-jmon) HOME(/tmp) PROGRAM(/bin/sh))
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
-

```

ADDUSER STCRSE DFLTGRP(STCGROUP) NOPASSWORD NAME('RDZ - RSE DAEMON')
OMVS(UID(#user-id-rse) HOME(/tmp) PROGRAM(/bin/sh) ASSIZEMAX(2147483647)
)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
• RDEFINE STARTED DBGMR.* DATA('RDZ - DEBUG MANAGER')
 STDATA(USER(STCDBM) GROUP(STCGROUP) TRUSTED(NO))
•
RDEFINE STARTED JMON.* DATA('RDZ - JES JOBMONITOR')
 STDATA(USER(STCJMON) GROUP(STCGROUP) TRUSTED(NO))
•
RDEFINE STARTED RSED.* DATA('RDZ - RSE DAEMON')
 STDATA(USER(STCRSE) GROUP(STCGROUP) TRUSTED(NO))
•
SETROPTS RACLIST(STARTED) REFRESH

```

#### Anmerkung:

- Stellen Sie sicher, dass die Benutzer-IDs der gestarteten Tasks durch Angabe des Schlüsselworts NOPASSWORD geschützt sind.
- Stellen Sie sicher, dass der RSE-Server eine eindeutige OMVS-Benutzer-ID besitzt, denn dieser Benutzer-ID werden Zugriffsrechte für z/OS UNIX gewährt.
- Der RSE-Dämon benötigt für den ordnungsgemäßen Betrieb einen großen Adressraum (2 GB). Legen Sie diesen Wert in der Variable ASSIZEMAX des OMVS-Segments für die Benutzer-ID STCRSE fest. Durch Festlegung dieses Werts wird sichergestellt, dass der RSE-Dämon unabhängig von Änderungen an MAXASSIZE in SYS1.PARMLIB(BPXPRMxx) die erforderliche Regionsgröße erhält.
- Für den ordnungsgemäßen Betrieb von RSE ist außerdem eine große Anzahl von Threads erforderlich. Sie können einen Grenzwert in der Variable THREADSMAX des OMVS-Segments für die Benutzer-ID STCRSE festlegen. Durch Festlegung dieses Grenzwerts wird sichergestellt, dass RSE unabhängig von Änderungen an MAXTHREADS oder MAXTHREADTASKS in SYS1.PARMLIB(BPXPRMxx) den erforderlichen Grenzwert für Threads erhält. Lesen Sie den Abschnitt "Optimierungsaspekte" im Handbuch *Hostkonfigurationsreferenz* (SC12-4489), um den korrekten Grenzwert für Threads zu ermitteln.
- Für die Benutzer-ID STCJMON ist es ebenfalls sinnvoll, THREADSMAX im OMVS-Segment festzulegen, da JES Job Monitor für jede Clientverbindung einen Thread verwendet.
- Die gestartete Task von Integrated Debugger (DBGMR) wird ausschließlich vom optionalen Integrated Debugger-Feature verwendet.

Überlegen Sie, ob für die Benutzer-ID STCRSE Einschränkungen definiert werden sollten. 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, ist es erforderlich, das Profil RESTRICTED.FILESYS.ACCESS in der Klasse UNIXPRIV mit UACC(NONE) zu definieren. Weitere Informationen zur Einschränkung von Benutzer-IDs finden Sie im *Security Server RACF Security Administrator's Guide* (SA22-7683).

**Achtung:** Wenn Sie Zugriffseinschränkungen 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. Die Ressourcen schließen auch solche Ressourcen ein, bei denen die Dokumentation von Developer for System z UACC verwendet, wie etwa das Profil **\*\*** in der Klasse **PROGRAM**, oder bei denen allgemeine z/OS UNIX-Konventionen gelten, wie zum Beispiel, dass jeder die Lese- und Ausführungsberechtigung für Java-Bibliotheken besitzt. Testen Sie den Zugriff vor der eigentlichen Aktivierung auf einem Produktionssystem.

## RSE als sicheren z/OS UNIX-Server definieren

RSE benötigt die Zugriffsberechtigung **UPDATE** für das Profil **BPX.SERVER**, um die Sicherheitsumgebung für den Client-Thread erstellen oder löschen zu können. Wenn dieses Profil nicht definiert ist, muss für RSE **UID(0)** verwendet werden. Dieser Schritt ist erforderlich, damit Clients die Verbindung herstellen können.

Integrated Debugger benötigt die Zugriffsberechtigung **UPDATE** für das Profil **BPX.SERVER**, um die Sicherheitsumgebung für den Debug-Thread erstellen oder löschen zu können. Wenn dieses Profil nicht definiert ist, muss für die Benutzer-ID der gestarteten **STCDBM**-Task **UID(0)** verwendet werden. Diese Genehmigung ist nur erforderlich, wenn das optionale Integrated Debugger-Feature verwendet wird.

- **RDEFINE FACILITY BPX.SERVER UACC(NONE)**
- **PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCRSE)**
- **PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCDBM)**
- **SETROPTS RACLIST(FACILITY) REFRESH**

**Achtung:** Mit dem Definieren des Profils **BPX.SERVER** wechselt z/OS UNIX vollständig von der Sicherheit auf UNIX-Ebene zur Sicherheit auf z/OS UNIX-Ebene, die bedeutend sicherer ist. Möglicherweise hat dieser Wechsel Auswirkungen auf andere z/OS UNIX-Anwendungen und -Operationen. Testen Sie die Sicherheit vor der eigentlichen Aktivierung auf einem Produktionssystem. Weitere Informationen zu den verschiedenen Sicherheitsebenen enthält die Veröffentlichung *UNIX System Services Planning* (IBM Form GA22-7800).

## Programmgesteuerte MVS-Bibliotheken für RSE definieren

Server mit der Berechtigung für **BPX.SERVER** müssen in einer sauberen, programm-gesteuerten Umgebung ausgeführt werden. Diese Anforderung impliziert, dass alle von RSE aufgerufenen Programme ebenfalls programmgesteuert sein müssen. Die Programmsteuerung von MVS-Ladebibliotheken wird von Ihrer Sicherheitssoftware verwaltet. Dieser Schritt ist erforderlich, damit Clients die Verbindung herstellen können.

RSE verwendet die Systembibliothek (**SYS1.LINKLIB**), die Bibliothek der Laufzeit von Language Environment (**CEE.SCEERUN\***) und die Ladebibliothek des TSO/ISPF-Client-Gateways von ISPF (**ISP.SISPLOAD**).

- **RALTER PROGRAM \*\* UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)**
- **RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)**
- **RALTER PROGRAM \*\* UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)**
- **RALTER PROGRAM \*\* UACC(READ) ADDMEM('ISP.SISPLOAD'//NOPADCHK)**
- **SETROPTS WHEN(PROGRAM) REFRESH**

**Anmerkung:** Wenn die Klasse PROGRAM bereits ein Profil \* enthält, sollten Sie das Profil \*\* nicht verwenden. Das Profil verkompliziert den von der Sicherheitssoftware verwendeten Suchpfad und macht ihn teilweise unkenntlich. Führen Sie in einem solchen Fall die vorhandenen Definitionen aus dem Profil \* mit den neuen Definitionen des Profils \*\* zusammen. Verwenden Sie das Profil \*\*, wie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683) dokumentiert.

Zur Unterstützung optionaler Services müssen die folgenden zusätzlich vorausgesetzten Bibliotheken programmgesteuert sein. Diese Liste enthält keine Dateien, die für ein Produkt spezifisch sind, mit dem Developer for System z interagiert, beispielsweise IBM File Manager.

- Alternative REXX-Laufzeitbibliothek für SCLM Developer Toolkit
  - REXX.\*.SEAGALT
- Systemladebibliothek für SSL-Verschlüsselung
  - SYS1.SIEALNKE
- Developer for System z-Bibliothek, für Integrated Debugger
  - FEK.SFEKAUTH

**Anmerkung:** Bibliotheken, die in den Link-Pack-Bereich (LPA) gestellt werden müssen, erfordern Programmsteuerberechtigungen, wenn für den Zugriff auf diese Bibliotheken LINKLIST oder STEPLIB verwendet wird. In dieser Veröffentlichung ist die Verwendung der folgenden LPA-Bibliotheken dokumentiert:

- ISPF für das TSO/ISPF-Client-Gateway
  - ISP.SISPLPA
- REXX-Laufzeitbibliothek für SCLM Developer Toolkit
  - REXX.\*.SEAGLPA
- Developer for System z für CARMA
  - FEK.SFEKLPA

## PassTicket-Unterstützung für RSE definieren

Das Kennwort des Clients (oder andere Identifikationsmittel, wie ein X.509-Zertifikat) wird nur verwendet, um die Identität bei der Verbindungsherstellung zu überprüfen. Danach wird die Threadsicherheit mit PassTickets verwaltet. Dieser Schritt ist erforderlich, damit Clients die Verbindung herstellen können.

PassTickets sind vom System generierte Kennwörter mit einer Lebensdauer von ca. 10 Minuten. Die generierten PassTickets basieren auf einem geheimen Schlüssel. Bei diesem Schlüssel handelt es sich um eine 64-Bit-Zahl (16 Hexadezimalzeichen). Ersetzen Sie in den folgenden RACF-Beispielbefehlen den Platzhalter key16 durch eine vom Benutzer angegebene 16-stellige Hexadezimalzeichenfolge mit den Zeichen 0-9 und A-F.

- ```
RDEFINE PTKTDATA FEKAPPL UACC(NONE) SSIGNON(KEYMASKED(key16))
APPLDATA('NO REPLAY PROTECTION – DO NOT CHANGE')
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
RDEFINE PTKTDATA IRRPTAUTH.FEKAPPL.* UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
PERMIT IRRPTAUTH.FEKAPPL.* CLASS(PTKTDATA) ACCESS(UPDATE) ID(STCRSE)
```
-

SETROPTS RACLIST(PTKTDATA) REFRESH

RSE unterstützt die Verwendung von anderen Anwendungs-IDs als FEKAPPL. Entfernen Sie in `rsed.envvars` die Kommentarzeichen vor der Option 'APPLID=FEKAPPL' und passen Sie die Option an, um sie zu aktivieren. Lesen Sie hierzu die Informationen im Abschnitt 'Zusätzliche Java-Startparameter mit `_RSE_JAVAOPTS` definieren' im Handbuch *IBM Rational Developer for System z Hostkonfiguration*. 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. Ebenso wenig sollten Sie 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. Entfernen Sie diese Berechtigung für die PTKTDATA-Profile.
- Damit JES Job Monitor die vom RSE angegebenen PassTickets überprüfen kann, müssen JES Job Monitor und RSE dieselbe Anwendungs-ID besitzen. Für JES Job Monitor erfolgt die Festlegung der Anwendungs-ID in der Konfigurationsdatei FEJJCNG mit der Anweisung APPLID.
- 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 hierzu das Schlüsselwort KEYENCRYPTED anstelle von KEYMASKED. Weitere Informationen finden Sie im *Security Server RACF Security Administrator's Guide* (IBM Form SA22-7683).

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

Anwendungsschutz für RSE definieren

Während der Clientanmeldung prüft der RSE-Dämon, ob ein Benutzer die Anwendung verwenden darf.

- ```
RDEFINE APPL FEKAPPL UACC(READ) DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
SETROPTS RACLIST(APPL) REFRESH
```

Anmerkung:

- RSE unterstützt die Verwendung von anderen Anwendungs-IDs als FEKAPPL. Ausführlichere Informationen dazu enthält der Abschnitt „PassTicket-Unterstüt-

zung für RSE definieren“ auf Seite 50. Die Klassendefinition APPL muss mit der eigentlichen, von RSE verwendeten Anwendungs-ID übereinstimmen.

- Die Clientverbindungsanforderung ist erfolgreich, wenn die Anwendungs-ID nicht in der Klasse APPL definiert ist.
- Die Clientverbindungsanforderung schlägt nur dann fehl, wenn die Anwendungs-ID definiert ist und der Benutzer nicht über Lesezugriff (READ) auf das Profil verfügt.

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 'FEJJCNFG, JES Job Monitor-Konfigurationsdatei' im Handbuch *IBM Rational Developer for System z Hostkonfiguration* dokumentiert.

Die folgenden RACF-Beispielbefehle gewähren Benutzern von Developer for System z bedingten Zugriff auf eine eingeschränkte Gruppe von JES-Befehlen, nämlich 'Hold', 'Release', 'Cancel' und 'Purge'. Benutzer haben nur dann eine Ausführungsberechtigung, wenn sie die Befehle über JES Job Monitor absetzen. Ersetzen Sie den Platzhalter `#console` durch den aktuellen Konsolennamen.

- ```
RDEFINE OPERCMDS MVS.MCSOPER.#console UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
RDEFINE OPERCMDS JES%.** UACC(NONE)
```
- ```
PERMIT JES%.** CLASS(OPERCMDS) ACCESS(UPDATE) WHEN(CONSOLE(JMON)) ID(*)
```
- ```
SETROPTS RACLIST(OPERCMDS) REFRESH
```

Anmerkung:

- Die Verwendung der Konsole ist zulässig, wenn kein Profil `MVS.MCSOPER.#console` definiert ist.
- 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 die Sicherheit vor der eigentlichen Aktivierung auf einem Produktionssystem.

In Tabelle 11 auf Seite 53 und Tabelle 12 auf Seite 53 sehen Sie die Bedienerbefehle, die für JES2 und JES3 abgesetzt werden, sowie die eigenständigen Sicherheitsprofile zu deren Schutz.

Tabelle 11. 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 12. 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 solche 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 enthält der Abschnitt "Aktionen für Beschränkungen der Jobziele" in *Hostkonfigurationsreferenz* (SC12-4489).
- Benutzer können jede Spooldatei anzeigen, sofern in der Konfigurationsdatei von JES Job Monitor nicht LIMIT_VIEW=USERID definiert ist. Weitere Informationen enthält der Abschnitt "Zugriff auf Spooldateien" in der *Hostkonfigurationsreferenz* (SC12-4489).
- Selbst Benutzer, die eigentlich 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, wie zum Beispiel 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, zum Beispiel JES Job Monitor im Gegensatz zu 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, die JES-Befehle über andere Mechanismen zu verwenden.

Dateiprofile definieren

Für die meisten Dateien von Developer for System z ist das Zugriffsrecht READ für Benutzer und ALTER für Systemprogrammierer ausreichend. Ersetzen Sie den Platzhalter #sysprog durch gültige Benutzer-IDs oder RACF-Gruppennamen. Erkundigen Sie sich außerdem bei dem Systemprogrammierer, der das Produkt installiert und konfiguriert hat, nach den korrekten Dateinamen. Das während der Installation verwendete übergeordnete Qualifikationsmerkmal (High Level Qualifier, HLQ) ist FEK. Das standardmäßige übergeordnete Qualifikationsmerkmal für Dateien, die während des Anpassungsprozesses erstellt werden, ist FEK.#CUST.

- ```
ADDGROUP (FEK) OWNER(IBMUSER) SUPGROUP(SYS1)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
```
- ```
ADDSD 'FEK.*.**' UACC(READ)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```
- ```
PERMIT 'FEK.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
```
- ```
SETROPTS GENERIC(DATASET) REFRESH
```

Anmerkung:

- Schützen Sie FEK.SFEKAUTH vor Aktualisierungen, denn diese Datei ist APF-autorisiert. Dasselbe gilt für FEK.SFEKLOAD und FEK.SFEKLPA, hier jedoch, weil diese Dateien programmgesteuert sind.
- Bei den Beispielbefehlen in dieser Veröffentlichung und im Job FEKRACF wird vorausgesetzt, dass EGN (Enhanced Generic Naming) aktiv ist. Wenn EGN aktiv ist, kann das Qualifikationsmerkmal ** verwendet werden, um eine beliebige Anzahl von Qualifikationsmerkmalen in der Klasse DATASET darzustellen. 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-Server) 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 ausnahmslos 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.SFEKMOD' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSD 'FEK.SFEKPROC' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSD 'FEK.#CUST.PARMLIB' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

```

```

-
ADDSD 'FEK.#CUST.CNTL' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSD 'FEK.#CUST.SQL' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
-
ADDSD 'FEK.#CUST.LSTRANS.*.**' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
-
ADDSD 'FEK.#CUST.CRA*.**' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
-
ADDSD 'FEK.#CUST.ADNREP*.**' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
-
ADDSD 'FEK.#CUST.ADNMAN*.**' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
• Berechtigung für den Systemprogrammierer zur Verwaltung aller Bibliotheken
-
PERMIT 'FEK.*.** CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.SFEKAUTH CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.SFEKLOAD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.SFEKLMOD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.SFEKPROC CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.PARMLIB CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.CNTL CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.SQL CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.LSTRANS.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.CRA*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.ADNREP*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEK.#CUST.ADNMAN*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
• Berechtigung für Clients zum Zugriff auf die Ladebibliotheken und Exec-Biblio-
theiken
-
PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(*)
-
PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(*)
-

```

```
PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(READ) ID(*)
```

```
—
```

```
PERMIT 'FEK.#CUST.CNTL' CLASS(DATASET) ACCESS(READ) ID(*)
```

```
— PERMIT 'FEK.#CUST.SQL' CLASS(DATASET) ACCESS(READ) ID(*)
```

Anmerkung: Für FEK.SFEKLPA sind keine Berechtigungen erforderlich, da der im LPA befindliche Code für alle zugänglich ist.

- Berechtigung für Integrated Debugger zum Zugriff auf die Ladebibliothek.

```
— PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCDBM)
```

- 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 CICS TS-Server, IMS-Regionen und MVS-Batch-Jobs zum Zugriff auf die Ladebibliothek für IRZ-Nachrichten (optional)

```
—
```

```
PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
```

```
PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#ims)
```

```
PERMIT 'FEK.SFEKLMOD' CLASS(DATASET) ACCESS(READ) ID(#batch)
```

- Sicherheitsprofile aktivieren

```
—
```

```
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.SISPLDAD
- ISP.SISPLPA
- SYS1.LINKLIB
- SYS1.SIEALNKE
- SYS1.SIEAMIGE
- REXX.V1R4M0.SEAGLPA

Anmerkung: Wenn Sie die Alternativbibliothek für das REXX-Produktpaket verwenden, lautet der Standardname der REXX-Laufzeitbibliothek REXX.*.SEAGALT und nicht REXX.*.SEAGLPA, wie im Beispiel oben verwendet.

Programmgesteuerte z/OS UNIX-Dateien für RSE definieren

Server mit der Berechtigung für BPX.SERVER müssen in einer sauberen, programm-gesteuerten Umgebung ausgeführt werden. Diese Anforderung impliziert, dass alle von RSE aufgerufenen Programme ebenfalls programmgesteuert sein müssen. Die Programmsteuerung 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 im programmgesteuerten Modus installiert.
- Ab z/OS 1.10 ist /usr/lib/libIRRRacf*.so Teil der System Authorization Facility (SAF), die zusammen mit dem Basisprodukt z/OS bereitgestellt wird, und ist daher auch für Kunden verfügbar, die RACF nicht 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 festgelegt.
- Verwenden Sie zum Anzeigen des aktuellen Status des Programmsteuerungsbits den z/OS UNIX-Befehl **ls -Eog**. Die Datei ist programmgesteuert, wenn der Buchstabe **p** in der zweiten Zeichenfolge angezeigt wird).

```
$ ls -Eog /usr/lib/libIRRRacf*.so
-rwxr-xr-x  aps-  2      69632 Oct  5  2007 /usr/lib/libIRRRacf.so
-rwxr-xr-x  aps-  2      69632 Oct  5  2007 /usr/lib/libIRRRacf64.so
```

Sicherheitseinstellungen prüfen

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

- Sicherheitseinstellungen und -klassen
 - SETROPTS LIST
- OMVS-Segment für Benutzer
 - LISTUSER #userid NORACF OMVS
 - LISTGRP #group-name NORACF OMVS
- Gestartete Tasks
 - LISTGRP STCGROUP OMVS
 - LISTUSER STCDBM OMVS
 - LISTUSER STCJMON OMVS
 - LISTUSER STCRSE OMVS
 - RLIST STARTED DBGMR.* ALL STDATA
 - RLIST STARTED JMON.* ALL STDATA
 - RLIST STARTED RSED.* ALL STDATA
- RSE als sicherer z/OS UNIX-Server
 - RLIST FACILITY BPX.SERVER ALL
- Programmgesteuerte MVS-Bibliotheken für RSE
 - RLIST PROGRAM ** ALL
- PassTicket-Unterstützung für RSE
 - RLIST PTKTDATA FEKAPPL ALL SSIGNON
 - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.* ALL
- Anwendungsschutz für RSE
 - RLIST APPL FEKAPPL ALL
- JES-Befehlssicherheit
 - RLIST CONSOLE JMON ALL
 - RLIST OPERCMDS MVS.MCSOPER.JMON ALL
 - RLIST OPERCMDS JES%.** ALL
- Dateiprofile
 - LISTGRP FEK
 - LISTDSO PREFIX(FEK) ALL
- Programmgesteuerte z/OS UNIX-Dateien für RSE
 - ls -E /usr/lib/libIRRRacf*.so

Optional kann es Profile geben, die das Verhalten von Developer for System z für einen bestimmten Benutzer steuern. Diese Profile stimmen mit dem Filter FEK.** überein und befinden sich standardmäßig in der Klasse FACILITY (siehe Steueranweisung _RSE_FEK_SAF_CLASS in rsed.envvars). Mithilfe des Befehls **SEARCH** können Sie die Profilnamen auflisten. Mit dem Befehl **RLIST** können Sie die Details für ein Profil anzeigen.

- SEARCH CLASS(FACILITY) FILTER(FEK.**)
- RLIST FACILITY #profile-name ALL

Kapitel 3. Hinweise zu TCP/IP

Developer for System z verwendet TCP/IP, um Benutzern einer Workstation den Zugriff auf Mainframe-Computer bereitzustellen, wenn diese selbst kein Mainframe-Computer ist. TCP/IP wird außerdem für die Datenübertragung zwischen verschiedenen Komponenten und anderen Produkten verwendet.

Beachten Sie, dass die meisten Funktionen von Developer for System z auf z/OS UNIX basieren und TCP/IP daher die z/OS UNIX-Suchreihenfolge verwendet, um nach den entsprechenden Konfigurationsdateien zu suchen. Weitere Informationen finden Sie im Abschnitt Kapitel 14, „TCP/IP konfigurieren“, auf Seite 209.

Dieses Kapitel enthält die folgenden Abschnitte:

- „TCP/IP-Ports“
- „TCP/IP-Standardverhalten überschreiben“ auf Seite 64
- „Mehrfachstack (CINET)“ auf Seite 64
- „Verteilte dynamische VIPA“ auf Seite 66

TCP/IP-Ports

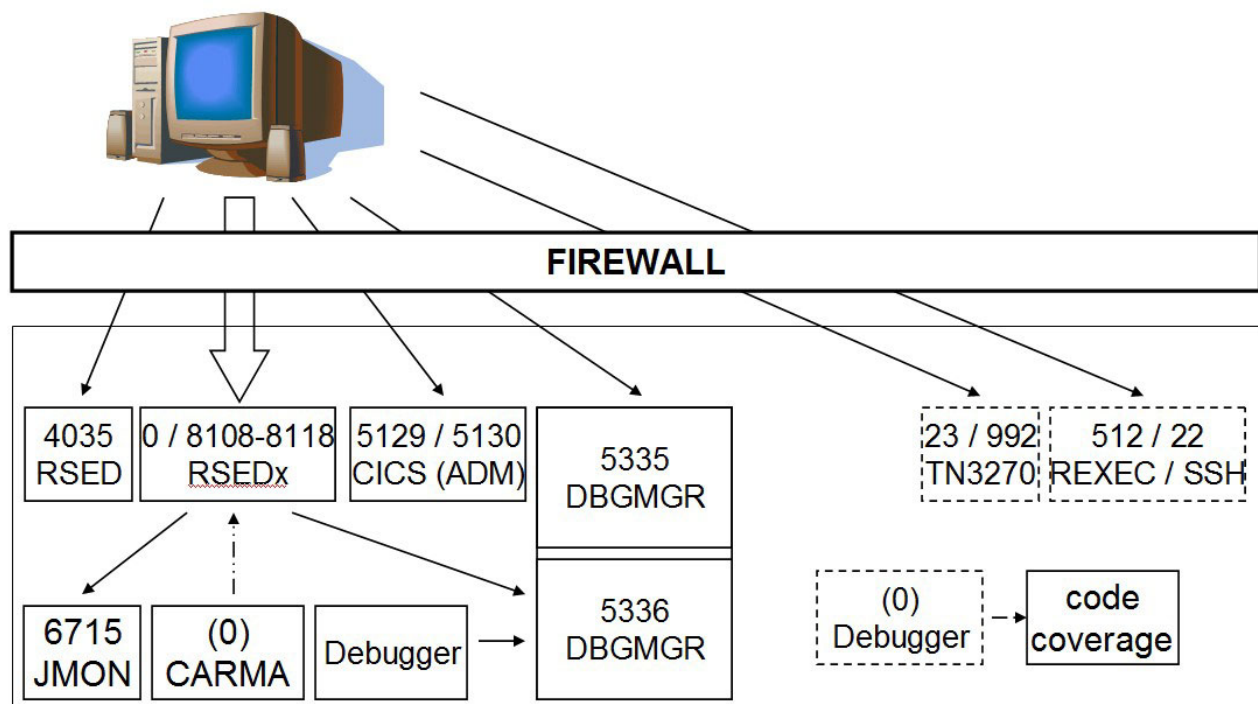


Abbildung 10. TCP/IP-Ports

Abb. 10 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). Der Port kann in der Konfigurationsdatei `rsed.envvars` festgelegt werden. Die Kommunikation über diesen Port kann mit SSL oder TLS 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. Der Standardportbereich für `_RSE_PORTRANGE` ist 8108-8118 (11 Ports). Die Kommunikation über diesen Port kann mit SSL oder TLS verschlüsselt werden.
- (Optional) Debug Manager für Integrated Debugger-Services, Standardport 5335. Der Port kann in der gestarteten `DBGMR-Task-JCL` festgelegt werden. Die Kommunikation über diesen Port kann mit SSL oder TLS 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 oder TLS verschlüsselt werden (Standardport 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. Der Port kann in der `CICS-CSD` festgelegt werden.
 - `Web-Service-Schnittstelle`, Standardport 5129. Der Port kann in der `CICS-CSD` festgelegt werden. Die Kommunikation über diesen Port kann mit SSL verschlüsselt werden.

Anmerkung: In der Regel gibt der Client an, welche TCP/IP-Adresse für die Verbindung zum Host verwendet werden soll. Um jedoch sicherzustellen, dass Debug-sitzungen mit dem korrekten Host kommunizieren, wird mit Debug Manager der Client festgelegt, dessen TCP/IP-Adresse verwendet werden muss.

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 `FEJJCNFG` festgelegt werden und wird in der Konfigurationsdatei `rsed.envvars` wiederholt.

- (Optional) Die CARMA-Kommunikation verwendet standardmäßig einen ephemeren Port. In der Konfigurationsdatei CRASRV.properties kann jedoch ein Portbereich festgelegt werden.
- (Optional) Debug Manager für Debug-bezogene Services, Standardport 5336. Der Port kann in der gestarteten DBGGMGR-Task-JCL festgelegt werden.
- Bei der hostbasierten Codeabdeckung, einem Batch-Job, wird ein ephemerer Port so zugeordnet, dass das Integrated Debugger mit ihm kommuniziert und die für den Codeabdeckungsbericht benötigten Daten liefert.

TCP/IP-Portreservierung

Wenn Sie in PROFILE.TCPIP die Anweisung PORT oder PORTRANGE verwenden, um die von Developer for System z verwendeten Ports zu reservieren, erfolgen zahlreiche Bindungen durch Threads, die im RSE-Thread-Pool aktiv sind. Der Jobname des RSE-Thread-Pools lautet RSEDx, wobei RSED der Name der gestarteten RSE-Task und x eine zufällige Ziffer ist. Daher sind in der Definition Platzhalter erforderlich.

```
PORT      4035      TCP RSED  ; Developer for System z - RSE daemon
PORT      6715      TCP JMON  ; Developer for System z - JES job monitor
PORT      5335      TCP DBGGMGR ; Developer for System z - Integrated
Debugger
PORT      5336      TCP DBGGMGR ; Developer for System z - Integrated
Debugger
PORTRange 8108 11   TCP RSED*  ; Developer for System z - _RSE_PORTRANGE
;PORTRange 5227 100 TCP RSED*  ; Developer for System z - CARMA
```

CARMA und TCP/IP-Ports

CARMA (Common Access Repository Manager) wird für den Zugriff auf einen hostbasierten Software Configuration Manager (SCM) verwendet, beispielsweise CA Endevor® SCM. 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, fordert der CARMA-Miner, der als Benutzerthread in einem RSE-Thread-Pool aktiv ist, einen ephemeren Port an oder sucht in dem Bereich, der in der Konfigurationsdatei CRASRV.properties angegeben ist, nach einem freien Port 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. Wenn der Server mit dem Port verbunden ist, kann der Client Anforderungen an den Server senden und Ergebnisse empfangen.

Aus Sicht des TCP/IP ist also 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.

Wenn Sie die Anweisung PORT oder PORTRANGE in PROFILE.TCPIP verwenden, um den Portbereich zu reservieren, der von CARMA verwendet wird, beachten Sie, dass der CARMA-Miner in einem RSE-Thread-Pool aktiv ist. Der Jobname des RSE-Thread-Pools lautet RSEDx, wobei RSED der Name der gestarteten RSE-Task und x eine zufällige Ziffer ist. Daher sind in der Definition Platzhalter erforderlich.

```
PORTRange 5227 100 RSED*      ; Developer for System z - CARMA
```

LDAP-Aspekte

Der RSE-Server kann so konfiguriert werden, dass er einen oder mehrere LDAP-Server nach verschiedenen Developer for System z-Services abfragt:

- LDAP-Gruppen können nach Unterstützung für Push-to-Client für mehrere Entwicklergruppen abgefragt werden.
- Eine oder mehrere Zertifikatswiderrufsliste können nach der X.509-Authentifizierung abgefragt werden.

TCP/IP-Sicherheitsvorkehrungen wie Firewalls können das Zustandekommen der Kommunikation zwischen dem (hostbasierten) RSE-Server und dem LDAP-Server verhindern. Beachten Sie daher folgende Informationen, um sicherzustellen, dass der LDAP-Server erreicht werden kann:

- Die TCP/IP-Adressen oder DNS-Namen des LDAP-Servers sind in `rsed.envvars` in den entsprechenden `*_LDAP_SERVER`-Variablen aufgelistet.
- Die Portnummern des LDAP-Servers sind in `rsed.envvars` in den entsprechenden `*_LDAP_PORT`-Variablen aufgelistet.
- LDAP verwendet das TCP-Protokoll.
- Der LDAP-Server wird vom hostbasierten RSE-Server kontaktiert.
- Der RSE-Server ist in einem RSEDx-Adressraum aktiv. Hierbei steht RSED für den Namen der gestarteten RSE-Task und x für eine zufällige Ziffer (z. B. RSED8).

TCP/IP-Standardverhalten überschreiben

Verzögertes ACK

Durch verzögertes ACK wird die Empfangsbestätigung (ACK) eines TCP-Pakets um bis zu 200 ms verzögert. Diese Verzögerung erhöht die Chance, dass die ACK zusammen mit der Antwort zum empfangenen Paket gesendet werden kann. Beziweckt wird damit eine Reduzierung des Netzverkehrs. Wenn der Absender allerdings vor dem Versenden eines neuen Pakets auf die ACK wartet (z. B. aufgrund der Implementierung des Nagle-Algorithmus) und auf das gerade gesendete Paket keine Antwort eingeht (z. B. weil es Teil einer Dateiübertragung ist), wird die Kommunikation unnötig verzögert.

Die Verzögerung der ACK kann in Developer for System z inaktiviert werden. Auf dem Host erfolgt dies, wie im Handbuch *Hostkonfiguration* (IBM Form SC23-7658) beschrieben, in `rsed.envvars` mit der Direktive `DSTORE_TCP_NO_DELAY`.

Mehrfachstack (CINET)

Mit z/OS Communication Server können auf einem einzelnen System mehrere TCP/IP-Stacks aktiv sein. Dies wird CINET-Konfiguration genannt.

Wenn Developer for System z im Standardstack nicht aktiv ist, schlagen bestimmte Developer for System z-Funktionen möglicherweise fehl. Die Verwendung von Stackaffinität ist eine sichere Möglichkeit, dieses Problem zu lösen. Stackaffinität weist Developer for System z an, statt jeden verfügbaren TCP/IP-Stack (Standardeinstellung für die gestarteten Tasks) nur einen bestimmten TCP/IP-Stack zu verwenden.

Stackaffinität wird für die gestarteten RSED-Tasks festgelegt, indem das Kommentarzeichen vor der Anweisung `_BPXX_SETIBMOPT_TRANSPORT` in der Konfigurationsdatei `rsed.envvars` entfernt und die Anweisung angepasst wird. Weitere Informationen zur Anpassung dieser Konfigurationsdatei finden Sie im entsprechenden Abschnitt in 'Kapitel 2. Basisanpassung' im Handbuch *Hostkonfiguration* (IBM Form SC12-4062).

CARMA und Stackaffinität

CARMA (Common Access Repository Manager) wird für den Zugriff auf einen hostbasierten Software Configuration Manager (SCM) verwendet, beispielsweise CA Endevor® SCM. Dazu startet CARMA einen benutzerspezifischen Server, der eine zusätzliche Konfiguration erfordert, um Stackaffinität umzusetzen.

Ähnlich den gestarteten Developer for System z-Tasks wird Stackaffinität für einen CARMA-Server mit der Variable `_BPXK_SETIBMOPT_TRANSPORT` festgelegt, die an LE (Language Environment) weitergegeben werden muss. Dies erfolgt durch Anpassung des Startbefehls in der aktiven Konfigurationsdatei `crastart*.conf` oder `CRA-SUB*`.

Anmerkung:

- Der genaue Name der Konfigurationsdatei, in der der Startbefehl enthalten ist, hängt von verschiedenen, vom Systemprogrammierer, der CARMA konfiguriert hat, getroffenen Auswahlmöglichkeiten ab. Weitere Informationen hierzu finden Sie in 'Kapitel 3. Common Access Repository Manager (optional)' im Handbuch *Hostkonfiguration* (IBM Form SC23-7658).
- `_BPXK_SETIBMOPT_TRANSPORT` gibt den Namen des zu verwendenden TCP/IP-Stacks an und wird in der Anweisung `TCPIPJOBNAME` in der zugehörigen Datei `"TCPIP.DATA"` definiert.
- Die Codierung einer `SYSTCPD DD`-Anweisung legt nicht die erforderliche Stackaffinität fest.
- Standardmäßig verwendet CARMA nicht die herkömmlichen TCP/IP-Stacks. CARMA verwendet vielmehr die Loopback-Adresse für die Kommunikation zwischen dem CARMA-Miner und dem CARMA-Server. Hierdurch wird nicht nur die Sicherheit gesteigert (nur lokale Prozesse haben Zugriff auf die Loopback-Adresse), sondern wahrscheinlich auch vermieden, dass Stackaffinität zur CARMA-Kommunikation hinzugefügt werden muss.

crastart*.conf

Ersetzen Sie den folgenden Abschnitt:

```
... PARM(&CRAPRM1. &CRAPRM2.)
```

durch Folgendes (dabei stellt TCPIP den gewünschten TCP/IP-Stack dar):

```
... PARM(ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIP") / &CRAPRM1. &CRAPRM2.)
```

Anmerkung: CRASTART unterstützt keine Zeilenfortsetzung. Die zulässige Zeilenlänge ist jedoch nicht begrenzt.

CRASUB*

Ersetzen Sie den folgenden Abschnitt:

```
... PARM(&PORT &TIMEOUT)
```

durch Folgendes (dabei stellt TCPIP den gewünschten TCP/IP-Stack dar):

```
... PARM(ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIP") / &PORT &TIMEOUT)
```

Anmerkung: Eine Jobübergabe begrenzt die Zeilenlänge auf 80 Zeichen. Sie können eine längere Zeile bei einem Leerzeichen () umbrechen und ein Pluszeichen (+) am Ende der ersten Zeile verwenden, um zwei Zeilen zu verknüpfen.

Verteilte dynamische VIPA

Mit der verteilten DVIPA (Dynamic Virtual IP Addressing, dynamische virtuelle IP-Adressierung) können Sie gleichzeitig zwei verschiedene Konfigurationen von Developer for System z auf unterschiedlichen Systemen in Ihrem Sysplex ausführen und die Clientverbindungen (mit optionaler Unterstützung durch WLM) über TCP/IP auf diese Systeme aufteilen.

Es gibt verschiedene Methoden, um eine verteilte DVIPA zu konfigurieren. In Developer for System z gelten allerdings einige Einschränkungen für diese Optionen.

- Der RSE-Dämon ist Eigner des Ports, der für die verteilte DVIPA definiert ist, aber die eigentliche Arbeit wird auf dem RSE-Server ausgeführt, der als Thread in einem anderen Adressraum aktiv ist. Aus diesem Grund können Sie nicht die Verteilungsmethode SERVERWLM für den Lastausgleich zwischen Ihren Systemen verwenden, weil WLM Ratschläge basierend auf den Statistiken des RSE-Dämons und nicht des RSE-Servers erstellt.
- Der Client kennt nur die DVIPA-Adresse, die von Sysplex Distributor für den RSE-Dämon verwendet wird. Sysplex Distributor übergibt die Verbindungsanforderungen an einen der verfügbaren RSE-Dämonen. Dieser startet anschließend einen RSE-Server-Thread, der an einen Port auf diesem System gebunden wird. Wenn der Client eine Verbindung zu diesem Port herstellt, verwendet er wieder die DVIPA-Adresse, nicht die tatsächliche Systemadresse. Sie müssen also sicherstellen, dass Sysplex Distributor die neue Verbindung auf das richtige System umleitet.

Aus diesem Grund ist in Developer for System z die Definition von `SYSPLXPORTS` in der Anweisung `VIPADISTRIBUTE` erforderlich, um sicherzustellen, dass die von den RSE-Server-Threads verwendeten Ports im Sysplex eindeutig sind.

Anmerkung:

- Die Syntax von `SYSPLXPORTS` setzt voraus, dass die Struktur `EZBEPOR` in Ihrer Coupling-Facility definiert ist.
- Die Syntax von `SYSPLXPORTS` setzt voraus, dass TCP/IP einen ephemeren Port für die sekundäre Verbindung auswählt. Aus diesem Grund können Sie in den TCP/IP-Profilen mit den Anweisungen `PORT` und `PORTRANGE` keine Ports für diese Verbindungen reservieren. Es ist auch nicht möglich, `_RSE_PORTRANGE` in `rsed.envvars` zu verwenden, um die Ports einzuschränken, die von Developer for System z verwendet werden. Developer for System z stellt eine Ausweichlösung für diese Einschränkung bereit, da diese die Konfiguration der Firewall erschwert.

Beim Verwenden der verteilten DVIPA gelten auch einige Einschränkungen innerhalb von Developer for System z:

- Sie sollten die Anweisung `deny.nonzero.port` in der Datei `rsed.envvars` aktivieren, um sicherzustellen, dass der Client von Developer for System z die Auswahl des richtigen TCP/IP-Ports nicht beeinflusst.
- Alle teilnehmenden Server von Developer for System z müssen eine identische Konfiguration verwenden. `/usr/lpp/rdz` und `/etc/rdz` sollten auf allen teilnehmenden Systemen gemeinsam genutzt werden. Sie sollten außerdem die Verzeichnisse `/var/rdz/projects`, `/var/rdz/pushtoclient` und `/var/rdz/scldmt` gemeinsam nutzen, wenn diese verwendet werden. Beachten Sie, dass `/var/rdz/WORKAREA` und `/var/rdz/logs` für jedes System eindeutig sein müssen.

- In Kapitel 11, „Ausführung mehrerer Instanzen“, auf Seite 169 finden Sie Informationen dazu, welche Developer for System z-Komponenten gemeinsam genutzt werden müssen und welche für jedes System eindeutig sein müssen.

JES Job Monitor, CARMA und weitere Server von Developer for System z interagieren nur mit dem lokalen RSE. Daher ist keine DVIPA-Konfiguration erforderlich.

Integrated Debugger interagiert nur mit dem lokalen RSE. Daher ist keine DVIPA-Konfiguration erforderlich. Um sicherzustellen, dass Debugsitzungen mit dem korrekten Host kommunizieren, wird mit Debug Manager der Client festgelegt, dessen TCP/IP-Adresse verwendet werden muss. Daher ist keine DVIPA-Konfiguration erforderlich.

Verteilte DVIPA werden in Ihrem TCP/IP-Profil im Block VIPADynamic durch die Schlüsselwörter VIPADefine und VIPABackup definiert. Mit dem Schlüsselwort VIPADISTribute werden die erforderlichen Sysplex Distributor-Definitionen hinzugefügt. Bei der verteilten DVIPA ist es erforderlich, dass alle teilnehmenden Stacks sysplexfähig sind. Dies wird in Ihrem TCP/IP-Profil im Block IPCONFIG mit den Schlüsselwörtern SYSPLEXRouting und DYNAMICXCF festgelegt. Weitere Details zu diesen Anweisungen finden Sie im Handbuch *Communications Server: IP Configuration Reference* (IBM Form SC31-8776).

Weitere Informationen zum Einrichten der EZBEPORTS-Struktur in Ihrer Coupling-Facility finden Sie in *MVS Setting Up a Sysplex* (IBM Form SA22-7625) und *Communication Server: SNA Network Implementation Guide* (IBM Form SC31-8777).

Portauswahl beschränken

Die Syntax von SYSPLEXPORTS setzt voraus, dass TCP/IP einen ephemeren Port für die sekundäre Verbindung auswählt. Ein ephemerer Port ist ein beliebiger freier Port, der nicht reserviert ist. Die Verwendung eines ephemeren Ports kollidiert mit den bewährten Verfahren für Firewalls zur Einschränkung der Ports, die für die Kommunikation geöffnet werden, da nicht bekannt ist, welcher Port verwendet wird.

Sie können dieses Problem umgehen, indem Sie Developer for System z dazu zwingen, für die sekundäre Verbindung bekannte Ports zu verwenden; hierbei wird ein eindeutiger _RSE_PORTRANGE für jedes System definiert und sichergestellt, dass die verwendeten Portbereiche für die Verwendung von Developer for System z auf allen Systemen reserviert werden. Beachten Sie, dass für diese Umgehung der TCP/IP-APAR PM63379 erforderlich ist.

Um sicherzustellen, dass TCP/IP die sekundäre Verbindung an das richtige System weiterleitet, muss Developer for System z einen eindeutigen Portbereich auf jedem System verwenden. Dies impliziert, dass Sie keinen gemeinsamen, identischen Setup für die Systeme verwenden können, da _RSE_PORTRANGE in rsed.envvars eindeutig sein muss. Informationen zur Einrichtung mehrerer Server mit unterschiedlichen Konfigurationsdateien bei der Verwendung desselben Codes finden Sie im Abschnitt zu „Identische Softwareversionen mit unterschiedlichen Konfigurationsdateien“ auf Seite 170 in Kapitel 11, „Ausführung mehrerer Instanzen“, auf Seite 169. Sie sollten eine Masterkopie von rsed.envvars und ein Script verwenden, um die Datei anzupassen und auf ein systemspezifisches Setup zu kopieren, um sicherzustellen, dass die Datei auf den unterschiedlichen Systemen identisch bleibt.

1. Richten Sie Developer for System z unter SYS1 wie eine Einzelsystemkonfiguration ein; stellen Sie aber sicher, dass sich /usr/lpp/rdz und /etc/rdz auf einem gemeinsam genutzten Dateisystem befinden. Alle MVS-basierten Teile müssen auch mit SYS2 gemeinsam genutzt werden.

2. Verwenden Sie die Datei /etc/rdz/rsed.envvars als Masterkopie und fügen Sie einen Verweis auf /etc/rdz am Ende der Datei hinzu, damit die systemspezifischen Kopien die übrigen Konfigurationsdateien aufnehmen können.

```
$ oedit /etc/rdz/rsed.envvars
-> add the following at the END:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

3. Erstellen Sie das Shell-Skript /etc/rdz/update.sh, mit dem die Masterdatei rsed.envvars kopiert und _RSE_PORTRANGE angepasst wird.

```
$ oedit /etc/rdz/update.sh
$ chmod 755 /etc/rdz/update.sh
```

```
#!/bin/sh
# Licensed materials - Property of IBM
# 5724-T07 Copyright IBM Corp. 2012
# clone rsed.envvars and set PORTRANGE for use with RDz & DDVIPA

file=rsed.envvars          #; echo file $file
sys=${1:-$(sysvar SYSNAME)} #; echo sys $sys
dir=$(dirname $0)          #; echo dir $dir
# if sysname has a special char, precede it with \ (eg. SYS\1)
case "$sys" in
    "SYS1") range=8108-8118;;
    "SYS2") range=8119-8129;;
esac                        #; echo range $range
echo "setting port range $range for $sys using $dir/$file"

if test ! $range ; then
    echo ERROR: no port range defined for $sys ; exit 12 ; fi
if test ! -e $dir/$file ; then
    echo ERROR: file $dir/$file does not exist ; exit 12 ; fi
if test ! -d $dir/$sys ; then
    echo ERROR: directory $dir/$sys does not exist ; exit 12 ; fi

mv $dir/$sys/$file $dir/$sys/prev.$file 2>/dev/null
sed="/_RSE_PORTRANGE/s/.*/_RSE_PORTRANGE=$range/"
sed "$sed" $dir/$file > $dir/$sys/$file

if test ! -s $dir/$sys/$file ; then
    echo ERROR creating $dir/$sys/$file, restoring backup
    mv $dir/$sys/prev.$file $dir/$sys/$file ; exit 8 ; fi
```

Abbildung 11. update.sh - DDVIPA-Setup mit einer Firewall unterstützen

4. Erstellen Sie die Verzeichnisse /etc/rdz/SYS1 und /etc/rdz/SYS2 und führen Sie /etc/rdz/update.sh aus, um die Verzeichnisse zu füllen.

```
$ mkdir /etc/rdz/SYS1 /etc/rdz/SYS2
$ /etc/rdz/update.sh SYS1
setting port range 8108-8118 for SYS1 using
/etc/rdz/rsed.envvars
$ /etc/rdz/update.sh SYS2
setting port range 8119-8129 for SYS2 using
/etc/rdz/rsed.envvars
```


5. Stellen Sie sicher, dass die gestartete Task RSED auf /etc/rdz/&SYSNAME verweist.
// CNFG='/etc/rdz/&SYSNAME.'

Anschließend müssen Sie sicherstellen, dass die definierten Portbereiche für Developer for System z auf allen Systemen im Sysplex reserviert sind, um sicherzustellen, dass die Portnummer im Sysplex eindeutig bleibt. Verwenden Sie die Anweisung PORT oder PORTRANGE in PROFILE.TCPIP, um alle Bereiche auf allen Systemen zu reservieren. Der Jobname des RSE-Thread-Pools lautet RSEDx, wobei RSED der Name der gestarteten RSE-Task und x eine zufällige Ziffer ist. Daher sind in der Definition Platzhalter erforderlich.

```
PORTRange 8108 22 RSED*           ; 8108-8129 - Developer for System z
                                   ; - secondary connection
```

Wie in „Verbindungsflow“ auf Seite 8 beschrieben, kann der Portbereich in _RSE_PORTRANGE schmal sein. Der RSE-Server muss den Port nicht exklusiv für die Dauer der Clientverbindung benötigen. Es kann sich nur während der Serverbindung an den Port und des Verbindungsaufbaus des Clients kein anderer RSE-Server an den Port binden. Das bedeutet, dass für die meisten Verbindungen der erste Port des Portbereichs verwendet wird, die restlichen Ports des Bereichs also nur als Puffer für den Fall dienen, dass mehrere Anmeldungen gleichzeitig erfolgen.

Beispielkonfiguration

Die folgende Musterkonfiguration enthält zwei z/OS-Systeme, SYS1 und SYS2, die Teil eines Sysplex sind. System "SYS1" ist als das System definiert, das gewöhnlich Sysplex Distributor für die verteilte DVIPA von Developer for System z bereitstellt.

Nachdem die verteilte DVIPA definiert wurde, kann Developer for System z auf den Systemen gestartet werden, um den Lastausgleich für Clientverbindungen zwischen den Systemen zu ermöglichen. JES Job Monitor interagiert nur mit dem lokalen RSE. Daher ist keine DVIPA-Konfiguration erforderlich. Clients stellen eine Verbindung mit Port 4035 für die IP-Adresse 10.10.10.1 her.

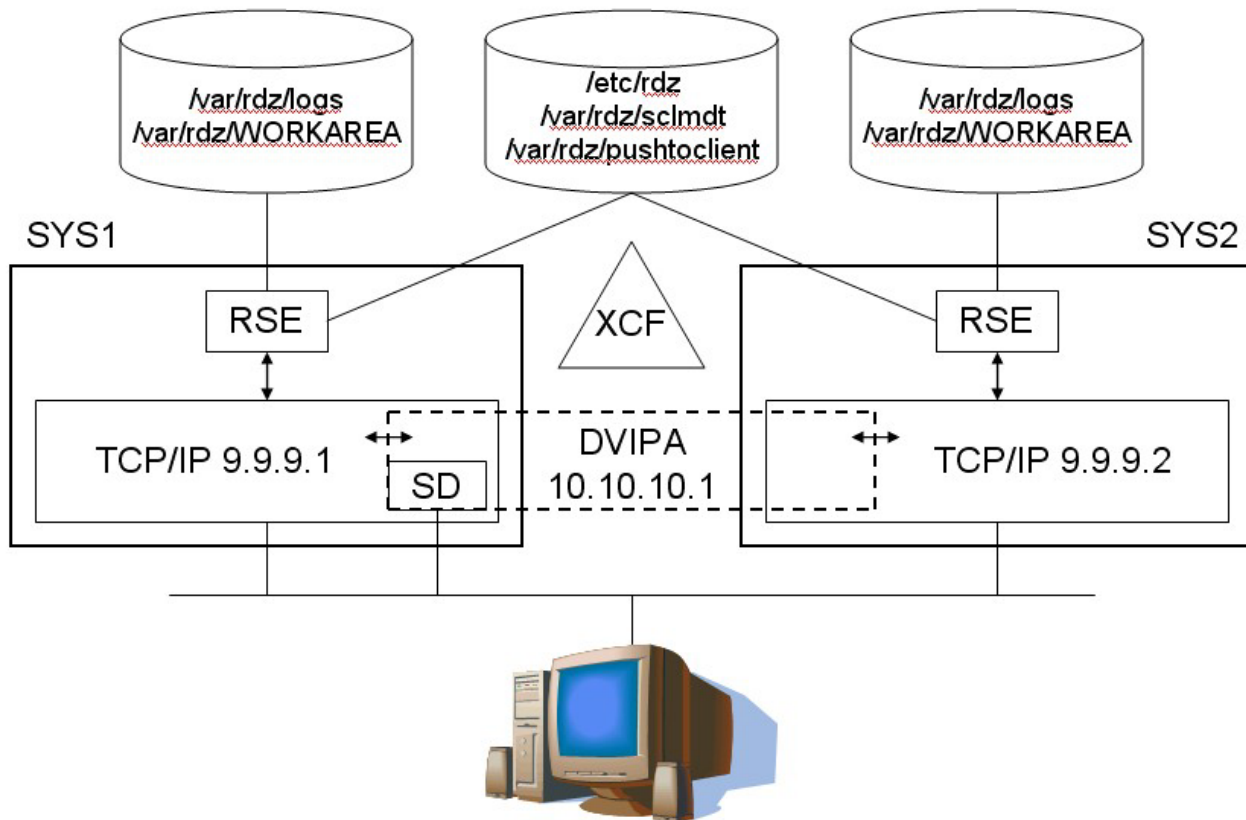


Abbildung 12. Beispiel für eine verteilte dynamische VIPA

System "SYS1" – TCP/IP-Profil

IPCONFIG

```
SYSPLEXRouting
; SYSPLEXROUTING is required as this stack needs sysplex communication
DYNAMICXCF 9.9.9.1 255.255.255.0 1
; DYNAMICXCF defines device/link with home address 9.9.9.1 as needed
IGNORERedirect
```

VIPADYNAMIC

```
VIPADefine 255.255.255.0 10.10.10.1
; VIPADefine defines 10.10.10.1 as main DVIPA on SYS1 for RDz
VIPADISTRIBUTE DEFINE
; VIPADISTRIBUTE makes 10.10.10.1 a distributed DVIPA, must match SYS2
SYSPLEXEXPORTS ; RDz prereq
DISTMETHOD BASEWLM ; BASEWLM or ROUNDROBIN
10.10.10.1 ; DVIPA address used by RDz clients
PORT 4035 ; port used by RDz clients
DESTIP 9.9.9.1 9.9.9.2 ; RDz active on SYS1 and SYS2
ENDVIPADYNAMIC
```

System "SYS2" – TCP/IP-Profil

IPCONFIG

```
SYSPLEXRouting
; SYSPLEXROUTING is required as this stack needs sysplex communication
DYNAMICXCF 9.9.9.2 255.255.255.0 1
; DYNAMICXCF defines device/link with home address 9.9.9.2 as needed
IGNORERedirect
```

VIPADYNAMIC

```
VIPABACKUP 255.255.255.0 10.10.10.1
; VIPABACKUP defines 10.10.10.1 as backup DVIPA on SYS2 for RDz
VIPADISTRIBUTE DEFINE
; VIPADISTRIBUTE makes 10.10.10.1 a distributed DVIPA, must match SYS1
SYSPLEXEXPORTS ; RDz prereq
DISTMETHOD BASEWLM ; BASEWLM or ROUNDROBIN
10.10.10.1 ; DVIPA address used by RDz clients
PORT 4035 ; port used by RDz clients
DESTIP 9.9.9.1 9.9.9.2 ; RDz active on SYS1 and SYS2
ENDVIPADYNAMIC
```

Kapitel 4. 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 1, „Wissenswertes zu Developer for System z“, auf Seite 3 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 75

Klassifikation für Verarbeitungsprozesse

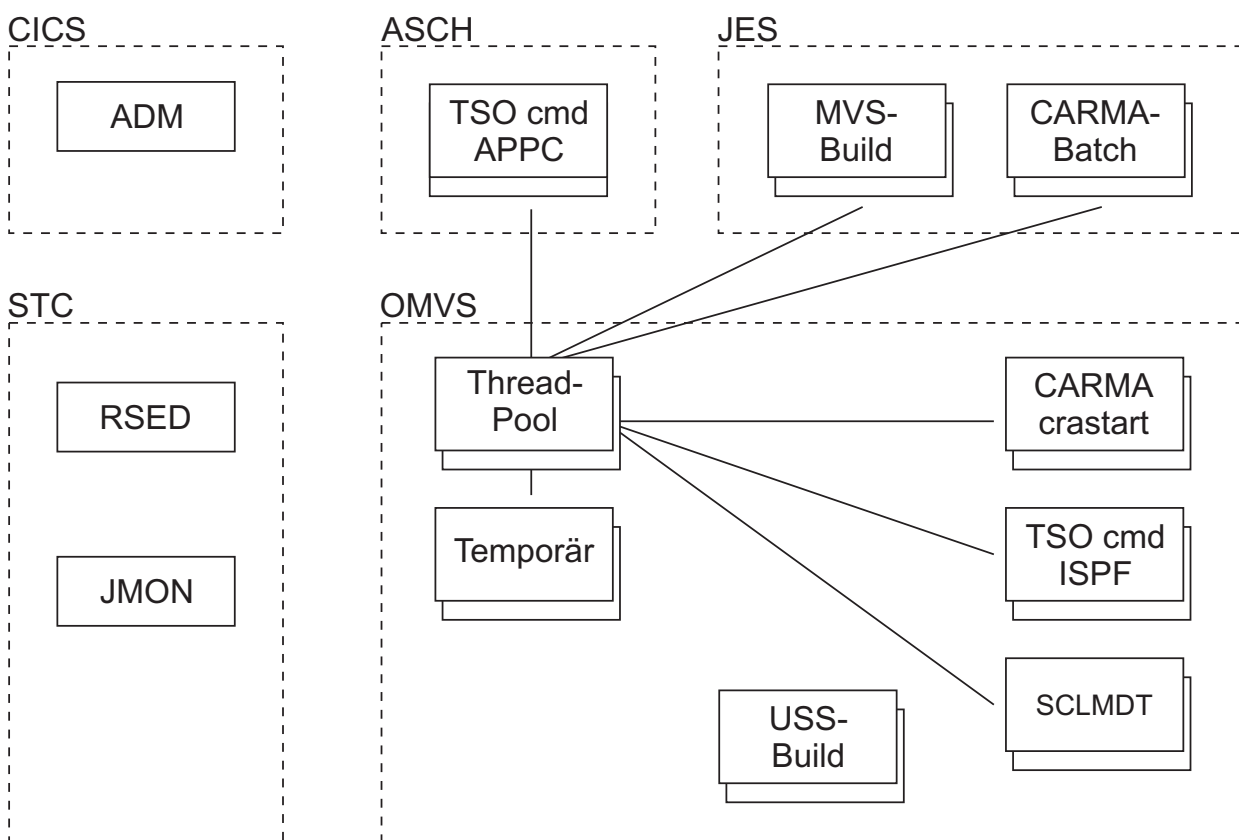


Abbildung 13. WLM-Klassifikation

Abb. 13 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) und JES Job Monitor (JMON) sind gestartete Tasks in Developer for System z (oder lang laufende Batch-Jobs) mit individuellen Adressräumen.

Wie in „RSE als Java-Anwendung“ auf Seite 5 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. 13 auf Seite 73 aufgeführten Adressräume bleiben für einen längeren Zeitraum im System sichtbar. Sie sollten allerdings beachten, dass z/OS UNIX so entwickelt wurde, dass es auch einige 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 13 werden die Subsystemtypen aufgeführt, die Verarbeitungsanforderungen von Developer for System z empfangen können.

Tabelle 13. WLM-Einstiegspunkt-Subsysteme

Subsystemtyp	Beschreibung der Arbeit
ASCH	Die Verarbeitungsanforderungen umfassen alle APPC-Transaktionsprogramme, die von dem von IBM gelieferten APPC/MVS-Transaktionsscheduler (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 13. WLM-Einstiegspunkt-Subsysteme (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.

Tabelle 14 listet zusätzliche Merkmale auf, 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 14. WLM-Qualifikationsmerkmale für Arbeitsvorgänge

		ASCH	CICS	JES	OMVS	STC
AI	Accountinformationen					
LU	LU-Name (*)					
PF	Ausführung (*)					
PRI	Priorität					
SE	Name der Terminierungsumgebung					
SSC	Objektgruppenname des Subsystems					
SI	Subsysteminstanz (*)					
SPM	Subsystemparameter					
PX	Sysplex-Name					
SY	Systemname (*)					
TC	Transaktions-/Jobklasse (*)					
TN	Transaktions-/Jobname (*)					
UI	Benutzer-ID (*)					

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 73 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 15 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 15. WLM-Verarbeitungsprozesse

Beschreibung	Taskname	Verarbeitungsprozess
JES Job Monitor	JMON	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 "SYS-STC" 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 und JES Job Monitor) bedienen Echtzeit-Clientanforderungen.

Tabelle 16. WLM-Verarbeitungsprozesse - STC

Beschreibung	Taskname	Verarbeitungsprozess
JES Job Monitor	JMON	STC
RSE-Dämon	RSED	STC

- JES Job Monitor

JES Job Monitor stellt alle Services mit Bezug zu JES bereit. Diese schließen das Übergeben von Jobs, das Durchsuchen von Spooldateien und das Ausführen von JES-Bedienerbefehlen ein. 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.

- 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 17. 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 Endeavor[®] 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

Bei dieser Workload werden vom Client ausgegebene z/OS UNIX-Shellbefehle verarbeitet. 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 gestartet 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 18. 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 Eingangsraten 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. Sie sollten beachten, dass die APPC-Methode nicht weiter unterstützt wird.

Tabelle 19. 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 20. 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 Regionsziel 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 5. Optimierungsaspekte

Wie in Kapitel 1, „Wissenswertes zu Developer for System z“, auf Seite 3 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, von denen jeder einzelne mindestens 17 Threads, eine bestimmte Speichermenge und möglicherweise einen oder mehr Adressräume verwendet, so müssen sowohl Developer for System z als auch z/OS ordnungsgemäß konfiguriert sein.

Dieses Kapitel enthält die folgenden Abschnitte:

- „Ressourcennutzung“
- „Speicherbelegung“ auf Seite 95
- „Speicherbelegung im z/OS UNIX-Dateisystem“ auf Seite 101
- „Definitionen von wichtigen Ressourcen“ auf Seite 104
- „Definitionen von verschiedenen Ressourcen“ auf Seite 108
- „Überwachung“ auf Seite 109
- „Beispielkonfiguration“ auf Seite 113

Ressourcennutzung

Verwenden Sie die Informationen in diesem Abschnitt, um Schätzwerte für 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 relativ präzisen Schätzwerten 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 enthalten die nachfolgenden Abschnitte:

- „Anzahl der Adressräume“ auf Seite 85
- „Anzahl der Prozesse“ auf Seite 88
- „Anzahl der Threads“ auf Seite 91

Tabelle 21 liefert einen allgemeinen Überblick über die Schlüsselressourcen, die von den gestarteten Developer for System z-Tasks verwendet werden. Diese Ressourcen werden nur einmal angelegt. Sie werden von allen Developer for System z-Clients gemeinsam genutzt.

Tabelle 21. Allgemeine Ressourcennutzung

Gestartete Task	Adressräume	Prozesse	Threads
JMON	1	1	3
RSED	1	3	11
RSEDx	1 + 1 (a)	1 + 2	1 + 10

Anmerkung: (a) Ein APF-autorisierter Adressraum und mindesten 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 85 angegeben.

Tabelle 22 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 zugehörige Funktion aufruft.

Tabelle 22. Benutzerspezifische vorausgesetzte Ressourcennutzung

Vorausgesetzte Software	Adressräume	Prozesse	Threads
ISPF Client Gateway	1	2	4
APPC-Administrator	1	1	2

Tabelle 23 auf Seite 85 liefert einen allgemeinen Überblick über die 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 22.

Tabelle 23. Benutzerspezifische Ressourcennutzung

Benutzeraktion	Adressräume	Prozesse	Threads		
	Benutzer-ID	Benutzer-ID	Benutzer-ID	RSEDx	JMON
Anmelden	-	-	-	17	1
Zeitgeber für Inaktivitätszeit-limit	-	-	-	1	-
Suchen	-	-	-	1	-
Komprimierung für PDS(E) aufheben	ISPF	ISPF	ISPF	-	-
Datei öffnen	ISPF	ISPF	ISPF	1	-
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	-	-

Anmerkung: ISPF kann durch APPC ersetzt werden, außer bei SCLM Developer Toolkit.

Anzahl der Adressräume

In Tabelle 24 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 24. Anzahl der Adressräume

Anzahl	Beschreibung	Taskname	Gemeinsame Nutzung	Endet nach
1	JES Job Monitor	JMON	Ja	Nie
1	RSE-Dämon	RSED	Ja	Nie
1	RSE-APF berechtigt	RSEDx	Ja	Nie
(a)	RSE-Thread-Pool	RSEDx	Ja	Nie
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

Tabelle 24. Anzahl der Adressräume (Forts.)

Anzahl	Beschreibung	Taskname	Gemeinsame Nutzung	Endet nach
1u	CARMA (crastart)	<Benutzer-ID>x	Nein	7 Minuten oder Abmeldung des Benutzers
4u	CARMA (ispf, nicht weiter unterstützt)	(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

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 30 Benutzer pro Thread-Pool festgelegt.

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

- (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.
- 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. 14, um einen Schätzwert der maximalen Anzahl der Threads zu berechnen, die Developer for System z verwendet.

$$3 + A + N \cdot (x + y + z) + (2 + N \cdot 0.01)$$

Abbildung 14. 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, nicht weiter unterstützt)

- "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.
- "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. 15, um einen Schätzwert der maximalen Anzahl der Adressräume zu berechnen, die ein Developer for System z-Client verwendet (nicht dokumentierte temporäre Adressräume werden nicht berücksichtigt).

$$x + y + z$$

Abbildung 15. 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. 14 auf Seite 86).
- hängt "y" von den ausgewählten Konfigurationsoptionen ab und wird für die Formel dokumentiert, um die maximale Anzahl von Adressräumen zu berechnen (Abb. 14 auf Seite 86).
- "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. 14 auf Seite 86).

Die Definitionen in Tabelle 25 können die eigentliche Anzahl von Adressräumen begrenzen.

Tabelle 25. 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

Tabelle 26 listet die von Developer for System z verwendete Anzahl von Prozessen pro Adressraum auf. Dabei gibt "u" in der Spalte "Adressräume" an, dass der Betrag mit der Anzahl von gleichzeitigen Benutzern dieser Funktion multipliziert werden muss.

Tabelle 26. Anzahl der Prozesse

Prozesse	Adressräume	Beschreibung	Benutzer-ID
1	1	JES Job Monitor	STCJMON
3	1	RSE-Dämon	STCRSE
1	1	Von RSE APF autorisiert	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, nicht weiter unterstützt)	<Benutzer-ID>
1	3u	z/OS UNIX-Build (Shellbefehle)	<Benutzer-ID>
1	1u	z/OS UNIX-Shell	<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 85 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 85 beschrieben.

- 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 26 auf Seite 88 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. 16, um einen Schätzwert der maximalen Anzahl der Prozesse zu berechnen, die Developer for System z verwendet.

$$5 + 2 * A + N * (x + y + z) + (10 + N * 0.05)$$

Abbildung 16. Maximale Anzahl von Prozessen

Dabei

- entspricht "5" 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, nicht weiter unterstützt)

- "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.
- "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. 17, um einen Schätzwert der maximalen Anzahl der Prozesse zu berechnen, die STCRSE verwendet - Benutzer-ID der gestarteten RSED-Task (nicht dokumentierte temporäre Prozesse werden nicht berücksichtigt).

$$4 + 2 * A$$

Abbildung 17. Anzahl von Prozessen für STCRSE

Dabei

- entspricht "4" der Anzahl der Prozesse, die vom RSE-Dämon und den Adressräumen verwendet werden, für die es eine RSE-APF-Berechtigung gibt.
- stellt "A" die Anzahl der Adressräume der RSE-Thread-Pools dar.

Verwenden Sie die Formel in Abb. 18, um einen Schätzwert der maximalen Anzahl von Prozessen zu berechnen, die ein Developer for System z-Client verwendet (nicht dokumentierte temporäre Adressräume werden nicht berücksichtigt).

$$(x + y + z) + 5 * s$$

Abbildung 18. 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. 16 auf Seite 89).
- hängt "y" von den ausgewählten Konfigurationsoptionen ab und wird für die Formel dokumentiert, um die maximale Anzahl von Prozessen zu berechnen (Abb. 16 auf Seite 89).
- "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. 16 auf Seite 89).
- "s" entspricht 1, wenn SCLM Developer Toolkit verwendet wird. Ist dies nicht der Fall, hat "s" den Wert 0.

Die Definitionen in Tabelle 27 können die eigentliche Anzahl von Prozessen begrenzen.

Tabelle 27. 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 geschätzte 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 28 ist die Anzahl der Threads angegeben, die von ausgewählten Developer for System z-Funktionen verwendet werden. 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 28. Anzahl der Threads

Threads			Benutzer-ID	Beschreibung
RSEDx	Aktiv	Booten		
-	3 + 1u	-	STCJMON	JES Job Monitor
-	15	2	STCRSE	RSE-Dämon
-	1	-	STCRSE	RSE-APF berechtigt
10 (a) + 17u	-	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)

Tabelle 28. Anzahl der Threads (Forts.)

Threads			Benutzer-ID	Beschreibung
5u	4u	3u	STCRSE und <Benutzer-ID>	CARMA (ispf, nicht weiter unterstützt)
-	1u (c)	2u	<Benutzer-ID>	z/OS UNIX-Build (Shellbefehle)
6u	1u	-	STCRSE und <Benutzer-ID>	z/OS UNIX-Shell
1 (d)	-	-	STCRSE	Herunterladen
1 (e)	-	-	STCRSE	Suchen
-	(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 85 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 85 beschrieben.
- 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 28 auf Seite 91 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.
- (c) Ein z/OS UNIX-Build ruft verschiedene Builddienstprogramme auf, die möglicherweise Multithreaddienstprogramme sind. In Tabelle 28 auf Seite 91 ist die minimale Anzahl von gleichzeitigen z/OS UNIX-Build-Threads angegeben.
- (d) Für jeden Download von Hostdaten wird ein separater Thread verwendet. Dieser Thread wird beendet, sobald die Daten an den Client übertragen worden sind.
- (e) Bei jeder fernen Suche wird ein separater Thread verwendet. Dieser Thread wird beendet, sobald die Ergebnisse an den Client übertragen worden sind.
- Alle aufgelisteten Threads bleiben solange aktiv, bis der zugehörige Prozess endet (wenn nicht anders angegeben).
- Die normale Anzahl von Threads für von RSE APF autorisierten Code ist 1. Beim Start sind allerdings temporär 13 oder mehr simultane Threads aktiv.

Verwenden Sie die Formel in Abb. 19, um einen Schätzwert der maximalen Anzahl der Threads zu berechnen, die von einem RSE-Thread-Pool verwendet werden. Verwenden Sie die Formel in Abb. 20, um einen Schätzwert der maximalen Anzahl der Threads zu berechnen, die JES Job Monitor verwendet.

$$10 + N \cdot (17 + x + y + z) + (20 + N \cdot 0.1)$$

Abbildung 19. Maximale Anzahl von Threads in einem RSE-Thread-Pool

$$3 + N$$

Abbildung 20. 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 30 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, nicht weiter unterstützt)

- "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. Mehrere gleichzeitige Downloads und Suchvorgänge sind zwei Beispiele, für die Sie möglicherweise diese Puffergröße erhöhen müssen.

Die Definitionen in Tabelle 29 können die eigentliche Anzahl von Threads in einem Prozess begrenzen. Diese Option ist vor allem für die RSE-Thread-Pools wichtig.

Tabelle 29. 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 threadbezogene Steuerblöcke
rsed.envvars	Xmx	Legt die maximale Größe des Java-Heapspeichers fest. Dieser Speicher ist reserviert und deshalb nicht mehr für threadbezogene 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
FEJJCNFG	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.
- Der Operatorbefehl **DISPLAY PROCESS,CPU**, mit dem die aktiven Threads in einem Thread-Pool angezeigt werden, ist auf die Anzeige der ersten 4000 Threads beschränkt.

Temporäre Ressourcennutzung

Die in den vorigen Abschnitten dokumentierte Ressourcennutzung ist permanent für die Lebensdauer von Developer for System z oder semipermanent für bestimmte benutzerspezifische Tasks.

Developer for System z verwendet jedoch vorübergehend zusätzliche Ressourcen für Verwaltungstasks und um folgende Anforderungen zu erfüllen:

- Die Verarbeitung eines Prüflistendateiereignisses (Anweisung audit.action in rsed.envvars) verwendet einen zusätzlichen Thread, einen zusätzlichen Prozess und möglicherweise (falls audit.action.id festgelegt ist) einen zusätzlichen Adressraum.
- Die Verarbeitung eines Anmeldeereignisses (Anweisung logon.action in rsed.envvars) verwendet einen zusätzlichen Thread, einen zusätzlichen Prozess und möglicherweise (falls logon.action.id festgelegt ist) einen zusätzlichen Adressraum.
- Der Bedienerbefehl IVP PASSTICKET verwendet zwei zusätzliche Threads.
- Der Bedienerbefehl IVP DAEMON verwendet einen zusätzlichen Thread, einen zusätzlichen Prozess und einen zusätzlichen Adressraum.
- Der Bedienerbefehl IVP ISPF verwendet einen zusätzlichen Thread, einen zusätzlichen Prozess und einen zusätzlichen Adressraum plus die vom ISPF-Client-Gateway verwendeten Ressourcen.

Speicherbelegung

RSE ist eine Java-Anwendung, was bedeutet, dass bei der Planung der Speicherbelegung für Developer for System z zwei Speicherzuordnungsbegrenzungen berücksichtigt werden müssen: eine für die Größe des Java-Heapspeichers und eine 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. Beachten Sie, dass zur Speicherung von CPU-Zyklen die Garbage-Collection tendenziell wartet, bis der belegte Speicher tatsächlich gebraucht wird; daher bleibt Speicher, der nicht mehr verwendet wird, länger zugeordnet als absolut notwendig (und wird somit ausgelagert).

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 512 MB. Sie müssen einen Wert von 256 MB oder höher angeben. Bei der Ausführung im 64-Bit-Modus versucht Java, den Heapspeicher über der 2-GB-Grenze zuzuordnen, wodurch Speicher unterhalb der Grenze freigegeben wird.

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 „Adressraum, Größe“ auf Seite 191, 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
- `MEMLIMIT` in `SYS1.PARMLIB(SMFPRMxx)` für den 64-Bit-Adressierungsmodus

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. Bei der Ausführung im 64-Bit-Modus versucht Java, den Heapspeicher über der 2-GB-Grenze zuzuordnen, wodurch Speicher unterhalb der Grenze freigegeben wird.

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 Speichermenge (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.
- Bei der Ausführung im 64-Bit-Modus müssen Sie sicherstellen, dass der Speicher über der 2-GB-Grenze tatsächlich für RSE verfügbar ist.
- Weitere Informationen für die Festlegung der Größenbeschränkungen für Adressräume finden Sie in „Adressraum, Größe“ auf Seite 191.
- Die Beispielkonfiguration `rsed.envvars` ist auf 30 Benutzer pro Thread-Pool festgelegt.
 - `maximum.clients=30`
 - `maximum.threads=520` ($10+17*30 = 520$, also 520 für 30 Clients möglich)
- Die Beispielkonfiguration `rsed.envvars` lässt eine Java-Heapspeichergröße von 512 MB zu. Damit können 30 Clients mit einem Durchschnitt von 17 MB Speicher pro Client verwendet werden ($30*17 = 510$).

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.

Als Referenz sehen Sie in Tabelle 30 die Werte, die von aktuellen Developer for System z-Kunden für die zentralen `rsed.envvars`-Einstellungen verwendet werden, die sich auf die Speicherbelegung auswirken.

Tabelle 30. Referenzeinstellungen für die Speicherbelegung

xmx (maximaler Java-Heapspeicher)	maximum.clients	Primärtyp der Entwicklung
512 MB	30	PL/I
512 MB	10	COBOL
384 MB	12	COBOL
800 MB (64-Bit)	20	Keine Angabe

Beispielanalyse der Speicherbelegung

Die Anzeigen in den folgenden Abbildungen zeigen einige Beispielzahlen für die Ressourcennutzung einer Standardkonfiguration von Developer for System z mit diesen Änderungen.

- `single.logon` ist inaktiviert, um RSE daran zu hindern, mindestens zwei Thread-Pool-Adressräume zu erstellen.
- 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
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
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
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
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
RSED	4.61	32.9M	16125
RSED8	4.77	62.3M	27404

Abbildung 21. 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
RSED	4.64	32.9M	16229
RSED8	4.78	62.4M	27413
RSED9	4.60	56.6M	24065

Abbildung 22. Ressourcennutzung mit 5 Anmeldungen (Fortsetzung)

Abb. 21 auf Seite 98 und Abb. 22 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
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
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
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
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 23. Ressourcennutzung beim Bearbeiten eines Members der untergliederten Datei

Abb. 23 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 Mitgliedern 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 12, „Konfigurationsprobleme lösen“, auf Seite 175 dokumentiert werden
- Protokollierung, die im Abschnitt „Prüfprotokollierung“ auf Seite 24 dokumentiert ist
- Push-to-Client-Metadaten, wie in „Push-to-Client-Metadaten“ auf Seite 129 dokumentiert.
- Temporäre Daten

Wie in Kapitel 12, „Konfigurationsprobleme lösen“, auf Seite 175 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 24. Speicherbelegung im z/OS UNIX-Dateisystem

Abb. 24 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 "Konsolennachrichten" im Handbuch *Hostkonfiguration* (SC12-4062).

Die Definitionen in Tabelle 31 steuern, welche Daten in den Protokollverzeichnissen gespeichert werden und wo sich diese Verzeichnisse befinden.

Tabelle 31. Anweisungen für die Protokollausgabe

Position	Anweisung	Funktion
resecomm.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
rsed.envvars	CGI_ISPWORK	Ausgangspfad für Protokolle von ISPF Client Gateway
rsed.envvars	TMPDIR	Verzeichnis für IVP-Protokolle
rsed.envvars	_CEE_DMPTARG	Verzeichnis für Java-Speicherauszüge

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 stets, diese temporären Dateien zu bereinigen. Eine manuelle Bereinigung, wie im Abschnitt "Bereinigung von WORKAREA und /tmp (optional)" in the *Hostkonfiguration* (SC12-4062) dokumentiert, kann jedoch quasi jederzeit durchgeführt werden.

Die Definitionen in Tabelle 32 steuern, wo temporäre Datenverzeichnisse gespeichert werden.

Tabelle 32. Anweisungen für temporäre Ausgabe

Position	Anweisung	Funktion
rsed.envvars	CGI_ISPWORK	Ausgangspfad für temporäre Daten
rsed.envvars	TMPDIR	Verzeichnis für temporäre Daten

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, für die keine der optionalen Komponenten von Developer for System z benötigt werden. Im Abschnitt "rsed.envvars (RSE-Konfigurationsdatei)" im Handbuch *Hostkonfiguration* (SC12-4062) werden alle Variablen beschrieben, die in der Beispieldatei definiert sind. Bei den folgenden Variablen müssen Sie allerdings besonders vorsichtig sein:

_RSE_JAVA_OPTS="\$_RSE_JAVA_OPTS -Xms128m -Xmx512m"

Festlegen der anfänglichen Heapgröße (Xms) und der maximalen Heapgröße (Xmx). Die Standardwerte sind jeweils 128M und 512M. 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.

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

Maximale Anzahl der Clients, die ein Thread-Pool bedienen kann. Die Standardeinstellung ist 30. 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_JAVA_OPTS="\$_RSE_JAVA_OPTS -Dmaximum.threads=520"

Maximale Anzahl von aktiven Threads in einem Thread-Pool, um neue Clients zuzulassen. Die Standardeinstellung ist 520. 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 (17 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.

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

#_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 bekannterweise 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: 'nnnnn' ist ein Dezimalwert zwischen 0 und 100.000.
Standardwert: 200

Anmerkung:

- Jeder Client verwendet mindestens 17 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 17 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.

MAXMAPAREA(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(nnnnnnnn)

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(FEJJC�FG)

Die in FEJJC�FG 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 "FEJJC�FG (JES Job Monitor-Konfigurationsdatei)" im Handbuch *Hostkonfiguration* (SC12-4062) 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 bekannterweise 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 zwischen 0 und 32767. Beachten Sie, dass die Summe der für die Systemparameter MAXUSER, RSVSTR1 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 bekannterweise 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 Transaktionscheduler ASCH und wird in *MVS Initialization and Tuning Reference* (IBM Form SA22-7592) beschrieben. Die folgenden Anweisungen wirken sich bekannterweise 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 Basisin-

formationen 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 finden Sie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062).

```
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        30
  MAXFILEPROC         83        103       64000
  MAXPROCUSER         97         99        200
  MAXTHREADS           9         14       1500
  MAXTHREADTASKS       9         14       1500
```

Die CPU-Option des Änderungsbefehls **DISPLAY PROCESS** zeigt die kumulierte CPU-Auslastung (in Millisekunden) der einzelnen Threads in einem Thread-Pool an:

```
f rsed,appl=d p,cpu
BPXM023I (STCRSE)
ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
USERID  THREAD-ID  TCB@  ACC_TIME TAG
STCRSE  0EDE54000000000  005E6B60  822 1/ThreadPoolProcess
STCRSE  0EDE8700000000001  005E69C8  001
STCRSE  0EDE9800000000002  005E6518  1814
STCRSE  0EDEBA00000000003  005E66B0  2305
STCRSE  0EDECB00000000004  005E62F8  001
STCRSE  0EDED000000000005  005E60D8  001
STCRSE  0EDF8600000000006  005C2BF8  628 6/ThreadPoolMonitor$Memory
UsageMonitor
STCRSE  0EDF9700000000007  005C2D90  003 7/ThreadPoolMonitor
IBMUSER 0EE2C700000000024  005C08B0  050 38/JESMiner
IBMUSER 0EE2B600000000026  005C0690  004 40/FAMiner
IBMUSER 0EE30B00000000027  005C0250  002 41/LuceneMiner
IBMUSER 0EE31C00000000028  005C0030  002 42/CDTParserMiner
IBMUSER 0EE32D00000000029  005BDE00  002 43/MVSLuceneMiner
IBMUSER 0EE33E0000000002A  005BDBE0  002 44/CDTMVSParserMiner
```

- Wenn ein RSE-Thread-Pool-Prozess endet, werden detaillierte Statistiken zur Ressourcennutzung angezeigt (wie nach der Ausgabe des Änderungsbefehls **DISPLAY PROCESS,DETAIL** nur für diesen RSE-Thread-Pool-Prozess). Die obere Grenze zeigt die maximale gleichzeitige Ressourcennutzung während der Dauer des RSE-Thread-Pool-Prozesses an, woran ein Systemoptimierer feststellen kann, ob RSE zu viele oder zu wenig Ressourcen zugewiesen sind.

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 Befeh-

len 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 IPCMSGQNUM = 10000
IPCMSGNIDS = 500 IPCSEMNIDS = 500
IPCSEMNOPS = 25 IPCSEMNSEMS = 1000
IPCshmPAGES = 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 BRM 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 *
IPCMGQBYTES	---	0	262144
IPCMGQMNUM	---	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 IOEFSCM
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

Wenn eine große Anzahl von Clients unterstützt wird, die eine Verbindung mit dem Host herstellen, so muss nicht nur Developer for System z in der Lage sein, die Arbeitslast zu verarbeiten, sondern auch Ihre Netzinfrastruktur. Die Netzverwaltung ist ein umfangreiches und gut dokumentiertes Thema, das nicht in der Dokumentation Developer for System z erörtert 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. Mit dem z/OS UNIX-Befehl **df** können Sie anzeigen, 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.LPP.FEK)   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 20 MB
- Benutzer verwenden eindeutige z/OS UNIX-Benutzer-IDs

Thread-Pool-Anzahl

Developer for System z versucht standardmäßig, 30 Benutzer zu einem einzigen Thread-Pool hinzuzufügen. In den Anforderungen ist allerdings angegeben, dass die Zeitlimitüberschreitung aufgrund von Inaktivität aktiv ist. In Tabelle 28 auf Seite 91 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 30 Benutzer einem einzigen Thread-Pool hinzufügt ($10+30*(17+1)=550$) und `maximum.threads` ist standardmäßig auf 520 gesetzt.

Es wäre möglich, `maximum.threads` zu erhöhen. Weil laut Anforderung allerdings pro Benutzer ein durchschnittlicher Java-Heapspeicher von 20 MB bereitgestellt werden soll, wird `maximum.clients` auf 25 ($10+25*18 = 460$) verringert. Auf diese Weise ($20*25 = 500$) wird die standardmäßige maximale Größe des Java-Heapspeichers (512 MB) beachtet.

Mit 25 Clients pro Thread-Pool und der Anforderung zur Unterstützung von 500 Verbindungen werden somit 20 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)
 $3 + A + N*(x + y + z) + (2 + N*0.01)$
 $3 + 20 + 500*1 + 200*1 + 300*1 + (2 + 500*0.01) = 1030$
- Anzahl der Adressräume (pro Benutzer)
 $x + y + z$
 $1 + 1 + 1 = 3$
- Anzahl der Prozesse (Maximum)
 $5 + 2*A + N*(x + y + z) + (10 + N*0.05)$
 $5 + 2*20 + 500*2 + 200*1 + 300*0 + (10 + 500*0.05) = 1570$
- Anzahl der Prozesse - STCRSE
 $4 + 2*A$
 $4 + 2*20 = 44$
- Anzahl der Prozesse (pro Benutzer)
 $(x + y + z) + 5*s$
 $(2 + 1 + 0) + 5*0 = 3$
- Anzahl der Threads – RSE-Thread-Pool
 $10 + N*(17 + x + y + z) + (20 + N*0.1)$
 $10 + 25*(17 + 1 + 4 + 0) + (20 + 25*0.1) = 583$
- Anzahl der Threads – JES Job Monitor
 $3 + N$
 $3 + 500 = 503$

- Benutzer-IDs

$$500 + 2 = 502$$

Die 2 zusätzlichen Benutzer-IDs werden für STCJMON und STCRSE (die Benutzer-IDs der gestarteten Tasks für Developer for System z) benötigt.

Grenzwerte definieren

Da jetzt die Zahlen für die Ressourcennutzung bekannt sind, können die begrenzenden Anweisungen mit entsprechenden Werten angepasst werden.

- /etc/rdz/rsed.envvars

- Xmx512m

nicht geändert

- Dmaximum.clients=25
- Dmaximum.threads=520

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(FEJCNFG)

- MAX_THREADS=503

- SYS1.PARMLIB(BPXPRMxx)

- MAXPROCSYS(2500)

mindestens 1572, zusätzliche Puffer für Tasks hinzugefügt, die nicht zu Developer

for System z gehören

- MAXPROCUSER(80)

mindestens 44, zusätzliche Puffer hinzugefügt, falls die RSE-Thread-Pools weniger als die geplanten 25 Clients unterstützen

- 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 582)

- 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 582)

- 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 1030, 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 „Grenzwerte definieren“ auf Seite 115 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. 25 auf Seite 117 zeigt die Ressourcennutzung, nachdem sich 499 Benutzer angemeldet haben. (Das Beispiel in der Abbildung zeigt nur die Anmeldung. Es sind keine Benutzeraktionen angegeben.)

```

F RSED,APPL=D P
BPXM023I (STCRSE)
ProcessId(83886168) Memory Usage(17%) Clients(25) Order(1)
ProcessId(91      ) Memory Usage(17%) Clients(25) Order(2)
ProcessId(122     ) Memory Usage(17%) Clients(25) Order(3)
ProcessId(16777348) Memory Usage(17%) Clients(25) Order(4)
ProcessId(16777358) Memory Usage(17%) Clients(25) Order(5)
ProcessId(16777368) Memory Usage(17%) Clients(25) Order(6)
ProcessId(16777378) Memory Usage(17%) Clients(25) Order(7)
ProcessId(16777388) Memory Usage(17%) Clients(25) Order(8)
ProcessId(16777398) Memory Usage(17%) Clients(25) Order(9)
ProcessId(33554622) Memory Usage(17%) Clients(25) Order(10)
ProcessId(16777416) Memory Usage(17%) Clients(25) Order(11)
ProcessId(16777426) Memory Usage(17%) Clients(25) Order(12)
ProcessId(16777436) Memory Usage(9%)  Clients(25) Order(13)
ProcessId(16777446) Memory Usage(17%) Clients(25) Order(14)
ProcessId(16777456) Memory Usage(17%) Clients(25) Order(15)
ProcessId(16777466) Memory Usage(17%) Clients(25) Order(16)
ProcessId(16777476) Memory Usage(17%) Clients(25) Order(17)
ProcessId(16777487) Memory Usage(17%) Clients(25) Order(18)
ProcessId(16777497) Memory Usage(17%) Clients(25) Order(19)
ProcessId(16777507) Memory Usage(16%) Clients(24) Order(20)

```

```

F RSED,APPL=D P,D
BPXM023I (STCRSE)
ProcessId(83886168) ASId(0022) JobName(RSED857 ) Order(1)
PROCESS LIMITS:      CURRENT  HIGHWATER    LIMIT
  JAVA HEAP USAGE(%)    17        17        100
    CLIENTS              25        25         25
  MAXFILEPROC           365       366      64000
  MAXPROCUSER            44        44         80
  MAXTHREADS            310       311      1500
  MAXTHREADTASKS        311       311      1500

```

TASID	Cpu time	Storage	EXCP
-----	-----	-----	-----
JMON	0.00	1780	73
RSED	5.88	95.2M	41958
RSED1	8.26	190.1M	58669
RSED1	8.17	187.0M	58605
RSED2	8.06	185.3M	58653
RSED2	8.19	183.1M	60209
RSED3	8.12	189.1M	58650
RSED3	8.03	186.7M	58590
RSED4	8.15	188.2M	58646
RSED4	5.50	182.5M	58585
RSED5	7.72	184.4M	58631
RSED5	7.82	184.1M	58576
RSED6	7.14	184.1M	58622
RSED6	6.27	186.9M	58583
RSED7	5.17	185.1M	58804
RSED7	6.57	185.2M	58621
RSED7	5.86	182.8M	58565
RSED8	0.36	1560	2459
RSED8	7.94	184.1M	58615
RSED8	7.45	181.8M	58548
RSED9	8.16	190.6M	58802
RSED9	7.62	183.8M	58610
RSED9	7.36	177.7M	57478

Abbildung 25. Ressourcennutzung der Beispielkonfiguration

Kapitel 6. 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 zuvor genannten Gründen sollten Sie diese Anweisung nicht ändern und die resultierenden Dateien stattdessen in die LINKLIST oder den LPA (Link-Pack-Bereich) stellen.

Verbesserung des Zugriffs auf Systembibliotheken

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(LNKLISTxx) 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 vorangehenden 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).

Verbesserung des Durchsatzes von Sicherheitsprüfungen

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), 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).

Die anfängliche und die maximale Größe des Heapspeichers einer z/OS Java Virtual Machine (JVM) können mit den Java-Befehlszeilenoptionen `-Xms` (Anfangsgröße) und `-Xmx` (maximale Größe) gesetzt werden.

In Developer for System z sind Java-Befehlszeilenoptionen in der Steueranweisung `_RSE_JAVA_OPTS` der Datei `rsed.envvars` definiert. Eine diesbezügliche Beschreibung finden Sie im Abschnitt "Zusätzliche Java-Startparameter mit `_RSE_JAVA_OPTS` definieren" in *Hostkonfiguration* (SC12-4062).

```
#_RSE_JAVA_OPTS="$_RSE_JAVA_OPTS -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)" in *Hostkonfiguration* (SC12-4062) 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.

Die Option `-Xquickstart` ist für Anwendungen geeignet, die eine kurze Laufzeit haben, insbesondere für jene, bei denen sich die Ausführungszeit nicht nur auf einige Methoden konzentriert. 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_JAVA_OPTS="$_RSE_JAVA_OPTS -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 An-

zahl 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 BPX-PRMxx-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.

- IPCSEMIDS und IPCSEMSEMS

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 7. Push-to-Client-Aspekte

Push-to-Client bzw. die hostbasierte Clientsteuerung unterstützt die zentrale Verwaltung der folgenden Komponenten:

- Clientkonfigurationsdateien
- Clientproduktversion
- Projektdefinitionen

Dieses Kapitel enthält die folgenden Abschnitte:

- „Einführung“
- „Primäres System“ auf Seite 128
- „Push-to-Client-Metadaten“ auf Seite 129
- „Clientkonfigurationssteuerung“ auf Seite 130
- „Clientversionssteuerung“ auf Seite 131
- „Mehrere Entwicklergruppen“ auf Seite 131
- „LDAP-basierte Gruppenauswahl“ auf Seite 135
- „SAF-basierte Gruppenauswahl“ auf Seite 140
- „Hostbasierte Projekte“ auf Seite 143

Einführung

Developer for System z-Clients ab Version 8.0.1 können beim Verbindungsaufbau Konfigurationsdateien und Produktaktualisierungsdaten im Pull-Verfahren vom Host abrufen. Dadurch wird sichergestellt, dass alle Clients über die gleichen Einstellungen verfügen und auf dem neuesten Stand sind.

Ab Version 8.0.3 kann der Clientadministrator mehrere Clientkonfigurationssätze und Clientaktualisierungsszenarien für die Anforderungen verschiedener Entwicklergruppen erstellen. Dadurch erhalten Benutzer eine angepasste Konfiguration, die auf Kriterien wie LDAP-Gruppenzugehörigkeit oder Zulassung zu einem Sicherheitsprofil basiert.

z/OS-Projekte können einzeln auf dem Client in der Perspektive für z/OS-Projekte erstellt werden. Sie können aber auch zentral auf dem Host erstellt werden, in welchem Fall sie auf Benutzerbasis auf den Client repliziert werden. Solche hostbasierten Projekte sind hinsichtlich Aussehen und Funktionsweise 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.

`pushtoclient.properties` entnimmt der Client, ob diese Funktionen aktiviert sind und wo die zugehörigen Daten gespeichert werden. Weitere Informationen hierzu finden Sie im Abschnitt '(Optional) `pushtoclient.properties`, Hostbasierte Clientsteuerung' im Handbuch *Hostkonfiguration* (IBM Form SC23-7658).

In der Regel werden z/OS-Systeme, die Workstations von Entwicklern und Entwicklungsprojekte von verschiedenen Benutzergruppen verwaltet. Das Design von Push-to-Client folgt diesem Prinzip, indem es jeder Gruppe bestimmte Aufgaben zuteilt:

- Der z/OS-Systemprogrammierer bestimmt die Speicherposition der Push-to-Client-Metadaten, die grundlegenden Sicherheitsaspekte sowie, ob Push-to-Client aktiv ist.
- Der Clientadministrator verwaltet die Push-to-Client-Metadaten. Er verwendet dazu den Developer for System z-Client zur Erstellung einer oder mehrerer Clientkonfigurationen sowie IBM Installation Manager zur Erstellung der Antwortdateien für die Aktualisierung des Developer for System z-Clients.
- Manager für Entwicklungsprojekte definieren ein Projekt und weisen ihm Entwickler zu.

Details zur Durchführung der zugewiesenen Aufgaben durch den Clientadministrator und den Manager für Entwicklungsprojekte finden Sie im Information Center für Developer for System z unter <http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp>.

Wenn die Konfigurations- oder Versionssteuerung Unterstützung für mehrere Entwicklergruppen umfasst, ist ein weiteres Team für die Verwaltung von Push-to-Client erforderlich. Wie sich dieses Team zusammensetzt, richtet sich danach, welche Option zur Identifizierung der Gruppen ausgewählt wurde, denen die Entwickler zugeteilt werden können:

- Ein LDAP-Administrator verwaltet Gruppendefinitionen, durch die die Entwickler keiner, einer oder mehreren FEK.PTC.*-LDAP-Gruppen zugewiesen werden können.
- Ein Sicherheitsadministrator verwaltet Zugriffslisten für FEK.PTC.*-Sicherheitsprofile. Ein Entwickler kann Zugriff auf keines, eines oder mehrere dieser Profile haben.

Primäres System

Push-to-Client sieht vor, dass systemspezifische Daten auf den einzelnen Systemen verbleiben, während allgemeine bzw. globale Daten auf einem zentralen System, dem primären System, verwaltet werden. Dadurch reduziert sich der Verwaltungsaufwand. Das primäre System wird in `pushtoclient.properties` durch die Direktive `primary.system` festgelegt. Die Standardeinstellung ist `false`.

Wichtig ist, dass nur ein System als primäres System definiert ist. Die Administratoren der Developer for System z-Clients können nur dann globale Konfigurationsdaten exportieren, wenn es sich beim Zielsystem um ein primäres System handelt. Sind allerdings mehrere primäre Systeme konfiguriert und deren Konfigurationen nicht synchronisiert, kann es auf den Developer for System z-Clients beim Verbindungsaufbau mit diesen Systemen zu Fehlern kommen.

Mehrere primäre Systeme können nur dann verwendet werden, wenn diese eine gemeinsame Developer for System z-Konfiguration (`/etc/rdz`) und gemeinsame Push-to-Client-Metadaten (`/var/rdz/pushtoclient`) verwenden. Da die Konfiguration in diesem Fall gemeinsam verwendet wird, gelten alle beteiligten Systeme als ein zusammengehöriges primäres System. So lange aber alle Systeme die gleichen Metadaten verwenden, ist diese Duplizierung kein wirkliches Thema.

Push-to-Client-Metadaten

Position von Metadaten

Die Direktive `pushtoclient.folder` in `pushtoclient.properties` legt das Basisverzeichnis fest, in dem die Push-to-Client-Metadaten gespeichert werden. Die Standardeinstellung ist `/var/rdz/pushtoclient`.

Das Basisverzeichnis enthält die Push-to-Client-Stammkonfigurationsdatei `keymapping.xml`. Alle anderen Metadaten werden in den Unterverzeichnissen gespeichert.

Die meisten Unterverzeichnisse werden dynamisch erstellt, wenn der Clientadministrator die Push-to-Client-Arbeitsbereichskonfiguration exportiert. Die Metadaten werden in diesen Unterverzeichnissen nach Inhalt (z. B. Zuordnungen und Benutzervorgaben) unterteilt. Je mehr Developer for System z-Clientkomponenten durch Push-to-Client verwaltet werden, desto mehr Unterverzeichnisse werden dynamisch erstellt. Welche Metadaten in diesen Unterverzeichnissen gespeichert werden, entnehmen Sie den Konfigurationsdateien, die Sie im Exportassistenten des Developer for System z-Clients mit der Befehlsfolge **Datei > Exportieren > Rational Developer for System z > Konfigurationsdateien** öffnen können.

Einige Unterverzeichnisse werden bereits bei der anfänglichen Hostanpassung erstellt. Diese Unterverzeichnisse werden vom Clientadministrator oder vom Manager des Entwicklungsprojekts manuell verwaltet.

- `/var/rdz/pushtoclient/projects/` enthält die hostbasierten Projektdefinitionsdateien. Die tatsächliche Position wird in der Datei `/var/rdz/pushtoclient/keymapping.xml` angegeben, die von einem Administrator der Developer for System z-Clients erstellt und verwaltet wird. Die enthaltenen Dateien werden von einem Projektleiter oder einem leitenden Entwickler verwaltet.
- `/var/rdz/pushtoclient/install/` enthält Konfigurationsdateien, durch die die Produktversion des Clients bei der Verbindung mit dem Host aktualisiert wird. Die tatsächliche Position wird in der Datei `/var/rdz/pushtoclient/keymapping.xml` angegeben, die von einem Administrator der Developer for System z-Clients erstellt und verwaltet wird. Die darin enthaltenen Dateien werden von einem Clientadministrator von verwaltet.
- `/var/rdz/pushtoclient/install/responsefiles/` enthält Konfigurationsdateien, durch die die Produktversion des Clients bei der Verbindung mit dem Host aktualisiert wird. Die tatsächliche Position wird in der Datei `/var/rdz/pushtoclient/keymapping.xml` angegeben, die von einem Administrator der Developer for System z-Clients erstellt und verwaltet wird. Die darin enthaltenen Dateien werden von einem Clientadministrator von verwaltet.

Weitere Informationen zur Erstellung dieser Unterverzeichnisse finden Sie im Abschnitt 'Anpassungskonfiguration' im Kapitel 'Basisanpassung' des Handbuchs *Hostkonfiguration* (IBM Form SC23-7658).

Sicherheit der Metadaten

Standardmäßig (siehe Direktive `file.permission` in `pushtoclient.properties`) erhalten alle Dateien und Verzeichnisse im Basisverzeichnis die Berechtigungsbitmaske 775 (`rw-rw-r-x`), durch die der Eigentümer und die Standardgruppe des Eigentümers Lese- und Schreibzugriff auf die Verzeichnisstruktur und die darin enthaltenen Dateien erhalten. Alle anderen Benutzer und Gruppen haben nur Lesezugriff auf die Verzeichnisstruktur und deren Inhalt.

Vor der Konfiguration der Push-to-Client-Funktion muss für diese Verzeichnisse die UID (Benutzer-ID) und GID (Gruppen-ID) des Eigentümers festgelegt werden.

Die folgenden RACF-Beispielbefehle erstellen eine neue Gruppe (RDZADMIN), weisen ihr eine eindeutige GID zu (2) und legen diese Gruppe als Standardgruppe für die Benutzer-ID RDZADM1 fest, die ebenfalls eine eindeutige UID erhält (6).

```
ADDGROUP RDZADMIN OWNER(IBMUSER) SUPGROUP(SYS1) –  
    DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CLIENT ADMIN')  
ALTGROUP RDZADMIN OMVS(GID(2))  
CONNECT RDZADM1 GROUP(RDZADMIN) AUTH(USE)  
ALTUSER RDZADM1 DFLTGRP(RDZADMIN) OMVS(UID(6))
```

Der folgende z/OS UNIX-Beispielbefehl für **chown** ändert den Eigentümer sowie die Gruppe der Verzeichnisstruktur /var/rdz/pushtoclient und aller darin enthaltenen Dateien in RDZADM1 (Eigentümer) bzw. RDZADMIN (Gruppe). Zur Vermeidung von Berechtigungskonflikten sollte der Befehl nur von einem Superuser (UID 0) ausgeführt werden.

```
chown -R rdzadm1:rdzadmin /var/rdz/pushtoclient
```

Der folgende z/OS UNIX-Beispielbefehl für **chmod** ändert die Berechtigungsbitmaske der Verzeichnisstruktur /var/rdz/pushtoclient und aller darin enthaltenen Dateien in 775. Diesen Befehl sollten Sie ausführen, um sicherzustellen, dass jegliches manuelles Hinzufügen zu diesem Verzeichnis der von Developer for System z verwendeten Logik folgt. Zur Vermeidung von Berechtigungskonflikten sollte der Befehl nur von einem Superuser (UID 0) ausgeführt werden.

```
chmod -R 775 /var/rdz/pushtoclient
```

Weitere Informationen zu RACF-Beispielbefehlen finden Sie in der Veröffentlichung *Security Server RACF Command Language Reference* (IBM Form SA22-7687). Weitere Informationen zu z/OS UNIX-Beispielbefehlen finden Sie in der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802). Weitere Informationen finden Sie im Abschnitt „z/OS UNIX-Verzeichnisstruktur“ auf Seite 15.

Speicherbelegung durch Metadaten

Die Push-to-Client-Metadaten benötigen unter z/OS UNIX nur wenig Plattenspeicher, da der größte Teil der Metadaten als UTF-8-codierte XML-Dateien vorliegt. Zudem kann der Produktcode für die Clientaktualisierungsszenarien überall im Netz gespeichert werden. Eine Speicherung unter z/OS UNIX ist nicht erforderlich, da die zugehörigen Push-to-Client-Metadaten (Antwortdateien) den Client auf die richtige Speicherposition verweisen.

Clientkonfigurationssteuerung

Beim Verbindungsaufbau eines Developer for System z-Clients (ab Version 8.0.1) mit dem Host liest der Client die Definitionen in `pushtoclient.properties`. Wenn die Direktive `config.enabled` aktiviert ist, vergleicht der Client seine aktuelle Konfiguration mit den Definitionen der Push-to-Client-Metadaten. Bei Abweichungen startet der Client einen Assistenten, der die erforderlichen Daten extrahiert und die Konfiguration nach Vorgabe durch Push-to-Client aktiviert.

Die Direktive `reject.config.updates` in `pushtoclient.properties` steuert, ob ein Benutzer die von Push-to-Client bereitgestellten Konfigurationsaktualisierungen zurückweisen darf.

Ein Developer for System z-Client (ab Version 8.0.1) stellt einen Assistenten bereit, mit dem der Clientadministrator die aktuelle Konfiguration exportieren kann, die dann wiederum mittels Push-to-Client von allen Developer for System z-Clients importiert wird. Da diese Funktion auf allen Clients verfügbar ist, sollten Sie sicherstellen, dass nur Clientadministratoren Schreibberechtigung für z/OS UNIX-Verzeichnisse haben, die Push-to-Client-Metadaten enthalten (/var/rdz/pushtoclient).

Zur Aktivierung der Unterstützung für Gruppen ist, wie in Abschnitt „Mehrere Entwicklergruppen“ beschrieben, sowohl auf dem Client als auch auf dem Host Version 8.0.3 oder höher erforderlich.

Clientversionssteuerung

Beim Verbindungsaufbau eines Developer for System z-Clients (ab Version 8.0.1) mit dem Host liest der Client die Definitionen in `pushtoclient.properties`. Wenn die Direktive `product.enabled` aktiviert ist, vergleicht der Client seine aktuelle Produktversion mit den Definitionen der Push-to-Client-Metadaten. Bei Abweichungen startet der Client einen Assistenten, der die erforderlichen Daten extrahiert und die Konfiguration nach Vorgabe durch Push-to-Client aktiviert.

Die Direktive `reject.product.updates` in `pushtoclient.properties` steuert, ob ein Benutzer die von Push-to-Client bereitgestellten Produktaktualisierungen zurückweisen darf.

Zur Aktivierung der Unterstützung für Gruppen ist, wie in Abschnitt „Mehrere Entwicklergruppen“ beschrieben, sowohl auf dem Client als auch auf dem Host Version 8.0.3 oder höher erforderlich.

Mehrere Entwicklergruppen

Ab Version 8.0.3 kann der Clientadministrator mehrere Clientkonfigurationssätze und Clientaktualisierungsszenarien für die Anforderungen verschiedener Entwicklergruppen erstellen. Dadurch erhalten Benutzer eine angepasste Konfiguration, die auf Kriterien wie LDAP-Gruppenzugehörigkeit oder Zulassung zu einem Sicherheitsprofil basiert.

Aktivierung

Unterstützung für mehrere Entwicklergruppen, jede mit eigener Clientkonfiguration und Clientaktualisierungsanforderungen, aktivieren Sie, wie in Tabelle 33 gezeigt, durch die Zuweisung des gewünschten Werts zu den jeweiligen Direktiven (`config.enabled` und `product.enabled`) in der Datei `pushtoclient.properties`.

Tabelle 33. Matrix zur Unterstützung von Push-to-Client-Gruppen für '.enabled'*

Wert von '*.enabled'	Funktion aktiviert?	Mehrere Gruppen unterstützt?
False	Nein	Nein
True	Ja	Nein
LDAP	Ja	Ja, basierend auf Zugehörigkeit der LDAP-Gruppen in FEK.PTC.*.ENABLED.sysname.devgroup
SAF	Ja	Ja, basierend auf Zulassung zu Sicherheitsprofilen in FEK.PTC.*.ENABLED.sysname.devgroup

Wenn die Funktion aktiviert ist (dies schließt den Wert TRUE ein), sind Entwickler immer Mitglied einer Standardgruppe. Zusätzlich kann ein Entwickler Mitglied keiner, einer oder mehrerer anderer Gruppen sein.

Die Zurückweisung von Aktualisierungen kann auch, wie in Tabelle 34 gezeigt, bedingt festgelegt werden.

Tabelle 34. Matrix zur Unterstützung von Push-to-Client-Gruppen für 'reject..updates'*

Wert von 'reject.*.updates'	Funktion aktiviert?
False	Nein
True	Ja
LDAP	Abhängig von LDAP-Gruppenzugehörigkeit in FEK.PTC.REJECT.*.UPDATES.sysname
SAF	Abhängig von Zulassung zu Sicherheitsprofil in FEK.PTC.REJECT.*.UPDATES.sysname

Die Direktiven in pushtoclient.properties sind unabhängig voneinander. Sie können jeder Direktive jeden unterstützten Wert zuweisen. Die Einstellungen müssen nicht gleich sein.

Informationen zur erforderlichen Konfiguration der jeweiligen Funktion finden Sie in den Abschnitten „LDAP-basierte Gruppenauswahl“ auf Seite 135 und „SAF-basierte Gruppenauswahl“ auf Seite 140. Weitere Informationen zur Aktivierung der Unterstützung für mehrere Gruppen finden Sie im Abschnitt '(Optional) pushtoclient.properties, Hostbasierte Clientsteuerung' im Handbuch *Hostkonfiguration* (IBM Form SC23-7658).

Gruppenverkettungen

Wenn eine *.enabled-Funktion in pushtoclient.properties aktiviert ist (dies schließt den Wert TRUE ein), sind Entwickler für die jeweilige Funktion immer Mitglied einer Standardgruppe. Zusätzlich kann ein Entwickler Mitglied keiner, einer oder mehrerer anderer Gruppen sein.

Damit die Anwendung von Änderungen, die in mehreren Gruppen definiert sind, nicht zu komplex wird, grenzt Developer for System z auf Basis einer vom Benutzer getroffenen Auswahl ein, welche Definitionen verwendet werden.

Tabelle 35. Push-to-Client-Gruppenverkettungen

Zusätzliche Gruppen	Verwendete Definitionen
Keine	Standard
Eine	Standard oder (Standard + Gruppe)
Mehrere	Standard oder (Standard + 1 Gruppe)

Developer for System z geht bei der Zusammenstellung und Anwendung des Änderungssets nach folgender Logik vor:

1. Übernahme der in den Standarddefinitionen angegebenen Aktualisierungen (sofern vorhanden).
2. Übernahme der in der ausgewählten Gruppendifinition angegebenen Aktualisierungen (sofern vorhanden), wobei die Standarddefinitionen gegebenenfalls überschrieben werden.

3. Anwenden der Aktualisierungen auf den Client.

Anmerkung: Aktualisierungen können die Aktionen 'Löschen', 'Hinzufügen' und 'Überschreiben' beinhalten.

Arbeitsbereichsbindung

Ein Entwickler kann mehreren Gruppen angehören, sein aktiver Arbeitsbereich nicht. Um Konfigurations- oder Produktaktualisierungen zu erhalten, muss der aktive Arbeitsbereich an eine bestimmte Gruppe gebunden sein (dies kann auch die Standardgruppe sein). Eine einmal erfolgte Bindung kann nicht mehr rückgängig gemacht werden. Falls eine neue Gruppenbindung erforderlich wird, muss ein neuer Arbeitsbereich erstellt werden.

Wenn ein Arbeitsbereich, der an noch keine Gruppe gebunden ist, eine Verbindung zum Host herstellt und `config.enabled` (oder `product.enabled`) angibt, dass Push-to-Client aktiv ist, fragt Developer for System z alle Gruppen ab, um festzustellen, zu welchen Gruppen der Benutzer gehört. Aus diesen Gruppen muss der Benutzer eine Gruppe für die jeweilige Funktion auswählen. Bei nachfolgenden Verbindungen wird nur noch die ausgewählte Gruppe abgefragt, um festzustellen, ob die Gruppenzugehörigkeit noch gültig ist.

Die `reject.*.updates`-Direktiven funktionieren nicht mit mehreren Gruppen. Ihre Konfiguration ist daher einfacher und erfordert keine Arbeitsbereichsbindung. Bei Vorliegen einer Aktualisierung stellt Developer for System z lediglich fest, ob der Benutzer die Aktualisierung zurückweisen darf, und handelt entsprechend.

Position der Gruppen-Metadaten

Wie im Abschnitt „Position von Metadaten“ auf Seite 129 beschrieben, werden alle Push-to-Client-Metadaten bei einer Konfiguration ohne Gruppenunterstützung in einer Verzeichnisstruktur direkt unter `/var/rdz/pushtoclient/` gespeichert. Dieses Datenlayout bleibt auch bei aktivierter Gruppenunterstützung erhalten, allerdings mit einer etwas abweichenden Interpretation des Basisverzeichnisses `/var/rdz/pushtoclient/`:

- Vorhandene Daten in `/var/rdz/pushtoclient/` werden als Daten der Standardgruppe interpretiert. Bei einem Export in die Standardgruppe werden die Metadaten in `/var/rdz/pushtoclient/` erstellt oder aktualisiert. Durch diese Interpretation wird die Kompatibilität mit Clients der Version 8.0.1 und 8.0.2 sichergestellt, die zwar Push-to-Client-fähig sind, aber keine Unterstützung für mehrere Gruppen bieten.
- Bei einem Export in eine Gruppe werden die Metadaten in `/var/rdz/pushtoclient/grouping/<devgroup>/` erstellt oder aktualisiert (genauso, als wäre dies statt `/var/rdz/pushtoclient/` das Basisverzeichnis). Der Wert von `<devgroup>` ist der Gruppenname, der einer Entwicklergruppe zugewiesen ist.

Bei der anfänglichen Produktanpassung wird das Verzeichnis `'grouping/'` unter `'/var/rdz/pushtoclient/'` erstellt. Der Clientadministrator muss die `<devgroup>/-` Verzeichnisse zu `/var/rdz/pushtoclient/grouping/` hinzufügen.

Die Verzeichnisse `projects/`, `install/` und `install/responsefiles/` werden bei der anfänglichen Produktanpassung in `/var/rdz/pushtoclient/` erstellt. Sollten gruppenspezifische Produktaktualisierungsszenarien oder gruppenspezifische host-basierte Projekte erforderlich sein, muss der Clientadministrator diese Verzeichnisse auch in `/var/rdz/pushtoclient/grouping/<devgroup>/` erstellen.

Folgende z/OS UNIX-Befehlsfolge zeigt, wie die Unterverzeichnisse mit der richtigen Berechtigungsbitmaske erstellt werden. Zur Vermeidung von Eigentumskonflikten sollten die Befehle nur vom Clientadministrator ausgeführt werden.

```
saved_umask=$(umask)
umask 0000
cd /var/rdz/pushtoclient/grouping/
mkdir -m775 <devgroup>
cd <devgroup>
mkdir -m775 install
mkdir -m775 install/responsefiles
mkdir -m775 projects
umask $saved_umask
```

Weitere Informationen zu z/OS UNIX-Beispielbefehlen finden Sie in der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802).

Konfigurationsschritte

Zur Einrichtung der Unterstützung für mehrere Entwicklergruppen ist zwischen dem z/OS-Systemprogrammierer, dem Clientadministrator und dem Administrator der Auswahlkriterien (LDAP- oder Sicherheitsadministrator) eine gewisse Zusammenarbeit erforderlich. Im folgenden Workflow verwaltet der Sicherheitsadministrator die Auswahlkriterien:

1. Der Clientadministrator erkundigt sich beim Sicherheitsadministrator nach der bestehenden Einteilung der Entwicklergruppen. Eine Beibehaltung der vorhandenen Einteilung beschleunigt und vereinfacht die Push-to-Client-Konfiguration.
2. Der Clientadministrator entscheidet, wie er die Unterstützung für mehrere Gruppen strukturieren will und wer den einzelnen Push-to-Client-Gruppen angehören soll.

Anmerkung:

- Für die Konfiguration und die Produktaktualisierung gibt es jeweils ein Standardszenario.
 - Push-to-Client-Änderungssets können die Aktionen 'Löschen', 'Hinzufügen' und 'Überschreiben' beinhalten.
 - Push-to-Client-Änderungssets können auch leer sein.
 - Ein Entwickler kann Mitglied keiner, einer oder mehrerer Push-to-Client-Gruppen sein.
 - Der Clientadministrator muss Mitglied jeder Push-to-Client-Gruppe sein.
3. Der Client- und der Sicherheitsadministrator einigen sich über die Namen der Push-to-Client-Gruppen.
 4. Der Clientadministrator erstellt das Verzeichnis
/var/rdz/pushtoclient/grouping/<devgroup>

für jede Push-to-Client-Gruppe.

Anmerkung: Als Berechtigungsbits für dieses Verzeichnis sollte 775 (drwxrwxr-x) festgelegt sein.

5. Der Sicherheitsadministrator führt die erforderliche Erstkonfiguration für die Definition der Push-to-Client-Auswahlbedingungen aus und fügt die Push-to-Client-Gruppen der Zugriffsliste hinzu.

Anmerkung:

- Die Strukturen der Auswahlkriterien müssen definiert werden. Allerdings muss zumindest der Clientadministrator in der Zugriffsliste definiert sein, damit er die entsprechenden Push-to-Client-Metadaten erstellen kann.
 - Bei der Erstkonfiguration sollte sich nur der Clientadministrator in der Zugriffsliste für eine Push-to-Client-Gruppe befinden. Dadurch wird verhindert, dass Developer for System z-Clients Konfigurationen empfangen, die noch bearbeitet werden.
6. Der z/OS-Systemprogrammierer aktiviert die Unterstützung für mehrere Gruppen in `pushtoclient.properties`.

Anmerkung: Die `*.enabled`-Direktiven müssen aktiviert sein, damit der Clientadministrator die entsprechenden Push-to-Client-Metadaten erstellen kann.

7. Der Clientadministrator erstellt die Arbeitsbereiche der einzelnen Gruppen und exportiert diese unter Angabe der jeweiligen Gruppennamen auf den Host. Außerdem erstellt der Clientadministrator die erforderlichen Antwortdateien für gruppenspezifische Produktaktualisierungsszenarien.
8. Der Sicherheitsadministrator fügt die Entwickler den Push-to-Client-Gruppen hinzu und aktiviert Push-to-Client für die Entwickler.

LDAP-basierte Gruppenauswahl

LDAP (Lightweight Directory Access Protocol) ist zwar der Name eines TCP/IP-basierten Protokolls, häufig wird es aber zur Beschreibung einer Gruppe von Services für verteilte Verzeichnisse verwendet. Bei einem Verzeichnis handelt es sich um eine strukturierte Sammlung von Datensätzen vergleichbar mit einer Datenbank. Developer for System z kann einen LDAP-Server als einfache hierarchische Datenbank verwenden, in der Gruppen ein oder mehrere Mitglieder enthalten.

Wenn Definitionen Ihres LDAP-Servers als Auswahlmechanismus verwendet werden (der LDAP-Wert wird für Direktiven in `pushtoclient.properties` angegeben), überprüft Developer for System z die Zugehörigkeit der in Tabelle 36 aufgelisteten Gruppennamen, um festzustellen, zu welchen Entwicklergruppen ein Benutzer gehört und ob der Benutzer Aktualisierungen zurückweisen darf.

Tabelle 36. Push-to-Client-relevante LDAP-Informationen

Gruppenname (cn=)	Ergebnis
FEK.PTC.CONFIG.ENABLED.sysname.devgroup	Der Client akzeptiert Konfigurationsaktualisierungen für die angegebene Gruppe.
FEK.PTC.PRODUCT.ENABLED.sysname.devgroup	Der Client akzeptiert Produktaktualisierungen für die angegebene Gruppe.
FEK.PTC.REJECT.CONFIG.UPDATES.sysname	Der Benutzer kann Konfigurationsaktualisierungen zurückweisen.
FEK.PTC.REJECT.PRODUCT.UPDATES.sysname	Der Benutzer kann Produktaktualisierungen zurückweisen.

Der Wert von `devgroup` ist der Gruppenname, der einer Entwicklergruppe zugewiesen ist. Dieser Gruppenname ist auf Developer for System z-Clients sichtbar.

Der Wert von `sysname` ist der Systemname des Zielsystems.

LDAP-Schema

Das LDAP-Schema muss folgenden Regeln folgen:

1. Jede Push-to-Client-Gruppe muss im Schema als Gruppe definiert sein.
2. Jeder Benutzer muss im Schema als Benutzer definiert sein.
3. Ein Gruppeneintrag enthält Referenzen auf die Einträge der Benutzer, die zur jeweiligen Gruppe gehören.

Abb. 26 zeigt eine LDAP-Musterdefinition für eine Gruppe und einen Benutzer im LDIF-Format.

Anmerkung: LDAP Data Interchange Format (LDIF) ist ein Standardtextformat für die Darstellung von LDAP-Objekten und LDAP-Aktualisierungen. Dateien mit LDIF-Einträgen werden zur Übertragung von Daten zwischen Verzeichnisservern oder als Eingabe für LDAP-Dienstprogramme verwendet.

```
# Group Definition
dn: cn=FEK.PTC.CONFIG.ENABLED.CDFMVS08.GROUPA,o=PTC,c=DeveloperForZ
objectClass: groupOfUniqueNames
objectClass: top
cn: FEK.PTC.CONFIG.ENABLED.CDFMVS08.GROUPA
description: Project A
uniqueMember: uid=mborn,ou=Users,dc=example,dc=com

# User Definition
dn: uid=mborn,ou=Users,dc=example,dc=com
objectClass: organizationalPerson
objectClass: person
objectClass: inetOrgPerson
objectClass: uidObject
objectClass: top
cn: May Born
sn: Born
uid: mborn
facsimiletelephonenumber: +1 800 982 6883
givenname: May
mail: mborn@example.com
ou: Users
```

Abbildung 26. Musterdefinition für das LDAP-Schema

LDAP-Serverauswahl

Ihnen steht eine große Auswahl an kommerziellen und kostenlosen LDAP-Servern zur Verfügung, beispielsweise IBM Tivoli Directory Server (<http://www-01.ibm.com/software/tivoli/products/directory-server/>). Auch Befehlszeilen- und grafisch orientierte Tools für die Verwaltung eines LDAP-Servers werden in Hülle und Fülle angeboten.

Wie im Abschnitt „LDAP-Schema“ erwähnt, muss jeder Benutzer für den LDAP-Server definiert sein. Zur Reduzierung der Verwaltung empfiehlt es sich, das Push-to-Client-Schema auf einem LDAP-Server einzurichten, der bereits Zugriff auf die Benutzerdefinitionen hat. Sie können zum Beispiel IBM Tivoli Directory Server unter z/OS mit einer SDBM-Datenbank verwenden (die SDBM-Datenbank fungiert als Wrapper für Ihre Sicherheitsdatenbank).

Je nach standortspezifischen Richtlinien kann das Push-to-Client-Schema auf einem LDAP-Server auch vom Clientadministrator verwaltet werden. Dieses Arrangement würde Absprachen zwischen Verantwortungsbereichen und damit einhergehende Verzögerungen und Kommunikationsfehler reduzieren.

Für die LDAP-Verwaltung durch den Clientadministrator spricht, dass das Push-to-Client-Schema keine vertraulichen oder sicherheitsrelevanten Daten enthält. Wenn die Benutzerdefinitionen dem LDAP-Server bereits durch andere Schemas vorliegen, bestimmen die LDAP-Objekte von Developer for System z lediglich, welche Optionen ein Entwickler bei der Auswahl des Arbeitsbereichslayouts und bei automatischen Produktaktualisierungen für den Developer for System z-Client hat.

LDAP-Serverposition

Jeder Datenbankserver, der das LDAP-Protokoll unterstützt, kann das Push-to-Client-Schema von Developer for System z bereitstellen. Daher können Sie die Informationen zum Verbindungsaufbau mit dem LDAP-Server in Developer for System z angeben. Auch das Suffix, das die Datenbank auf dem LDAP-Server eindeutig kennzeichnet, kann angegeben werden.

rased.envvars-Direktive	Standard
_RSE_LDAP_SERVER	Lokales Hostsystem
_RSE_LDAP_PORT	389
_RSE_LDAP_PTC_GROUP_SUFFIX	"O=PTC,C=DeveloperForZ"

TCP/IP-Sicherheitsvorkehrungen wie Firewalls können das Zustandekommen der Kommunikation zwischen dem (hostbasierten) RSE-Server und dem LDAP-Server verhindern. Bitten Sie Ihren TCP/IP-Administrator daher um folgende Informationen bzw. sorgen Sie dafür, dass diese Voraussetzungen erfüllt werden, um sicherzustellen, dass der LDAP-Server erreicht werden kann:

- TCP/IP-Adresse oder DNS-Name des LDAP-Servers
- Portnummer des LDAP-Servers
- LDAP verwendet das TCP-Protokoll
- Der LDAP-Server wird vom hostbasierten RSE-Server kontaktiert
- Der RSE-Server ist in einem RSEDx-Adressraum aktiv; hierbei steht RSED für den Namen der gestarteten RSE-Task und x für eine zufällige Ziffer

Beispielkonfiguration

Developer for System z wird in einem Unternehmen auf einem System namens CDFMVS08 ausgeführt. Als LDAP-Server wird IBM Tivoli Directory Server, das auch auf CDFMVS08 ausgeführt wird, verwendet. Der LDAP-Server ist konfiguriert, wie in „Push-to-Client-Back-End zu LDAP hinzufügen“ auf Seite 138 beschrieben.

Die folgenden Benutzer verwenden Developer for System z:

- Entwickler für Finanzanwendungen mit den Benutzer-IDs BNK010 -> BNK014
- Entwickler für Versicherungsanwendungen mit den Benutzer-IDs INS010 -> INS014
- Ein Developer for System z-Clientadministrator mit der Benutzer-ID RDZADM1

Jede Entwicklergruppe benötigt spezielle Clientkonfigurationsdateien, alle Entwickler unterliegen aber der gleichen Clientversionssteuerung. Im Gegensatz zu Clientadministratoren dürfen Entwickler die von Push-to-Client bereitgestellten Änderungen nicht zurückweisen.

Der Client- und der LDAP-Administrator einigen sich für Konfigurationsaktualisierungen auf die Gruppennamen BANKING und INSURANCE.

Push-to-Client-Back-End zu LDAP hinzufügen

In diesem Beispiel wird IBM Tivoli Directory Server unter z/OS, der bislang nur eine SDBM-Datenbank (Wrapper für Sicherheitsdatenbank) verwendet hat, eine LDBM-Datenbank (z/OS UNIX-Dateien) für das Push-to-Client-Schema hinzugefügt.

1. Fügen Sie dazu den Abschnitt für das LDBM-Back-End zur LDAP-Konfigurationsdatei hinzu.

```
# filename ds.conf
# restart GLDSRV started task to pick up changes

# global section
adminDN "cn=LDAP admin"
adminPW password
listen ldap://:389
schemaPath /etc/ldap

# SDBM back-end section (RACF)
database SDBM GLDBSD31/GLDBSD64
suffix "cn=RACF,o=IBM,c=US"

# LDBM back-end section (z/OS UNIX files)
database LDBM GLDBLD31/GLDBLD64 LDBM-RDZ
suffix "o=PTC,c=DeveloperForZ"
databaseDirectory /var/ldap/ldbm/rdz
```

2. Stoppen Sie die gestartete LDAP-Task GRDSRV und starten Sie sie neu, damit die Konfigurationsänderungen wirksam werden.

3. Erstellen Sie das Verzeichnis /var/ldap/ldbm/rdz.

```
mkdir -p /var/ldap/ldbm/rdz
```

4. Fügen Sie dem LDAP-Schema das LDBM-Back-End hinzu.

```
ldapmodify -D "cn=LDAP admin" -w password -f
/usr/lpp/ldap/etc/schema.user.ldif

ldapmodify -D "cn=LDAP admin" -w password -f
/usr/lpp/ldap/etc/schema.IBM.ldif
```

5. Fügen Sie dem LDBM-Back-End den Stammeintrag hinzu.

```
ldapadd -D "cn=LDAP admin" -w password -f
/u/ibmuser/ptc_root.ldif
```

Dabei enthält /u/ibmuser/ptc_root.ldif Folgendes:

```
dn: o=PTC,c=DeveloperForZ
objectclass: top
objectclass: organization
o: PTC
```

Anfängliche LDAP-Gruppenkonfiguration

Fügen Sie die einzelnen LDAP-Gruppenobjekte zum Schema hinzu und fügen Sie den Clientadministrator danach zu jedem Gruppenobjekt hinzu. Die Benutzerdefinition für die Benutzer-ID RDZADM1 wird mittels Pull-Verfahren aus dem RACF-Schema entnommen.

```
ldapadd -D "cn=LDAP admin" -w password -f /u/ibmuser/ptc_setup.ldif
```

Dabei enthält /u/ibmuser/ptc_setup.ldif Folgendes:

```
# banking workspace configuration
dn: cn=FEK.PTC.CONFIG.ENABLED.CDFMVS08.BANKING,o=PTC,c=DeveloperForZ
objectclass: groupOfUniqueNames
cn: FEK.PTC.CONFIG.ENABLED.CDFMVS08.BANKING
description: Developer for System z push-to-client
# give client administrator access
```

```

uniqueMember: racfID=RDZADM1,profileType=user,cn=RACF,o=IBM,c=US

# insurance workspace configuration
dn: cn=FEK.PTC.CONFIG.ENABLED.CDFMVS08.INSURANCE,o=PTC,c=DeveloperForZ
objectclass: groupOfUniqueNames
cn: FEK.PTC.CONFIG.ENABLED.CDFMVS08.INSURANCE
description: Developer for System z push-to-client
# give client administrator access
uniqueMember: racfID=RDZADM1,profileType=user,cn=RACF,o=IBM,c=US

# reject configuration updates
dn: cn=FEK.PTC.REJECT.CONFIG.UPDATES.CDFMVS08,o=PTC,c=DeveloperForZ
objectclass: groupOfUniqueNames
cn: FEK.PTC.REJECT.CONFIG.UPDATES.CDFMVS08
description: Developer for System z push-to-client
# give client administrator access
uniqueMember: racfID=RDZADM1,profileType=user,cn=RACF,o=IBM,c=US

# reject product updates
dn: cn=FEK.PTC.REJECT.PRODUCT.UPDATES.CDFMVS08,o=PTC,c=DeveloperForZ
objectclass: groupOfUniqueNames
cn: FEK.PTC.REJECT.PRODUCT.UPDATES.CDFMVS08
description: Developer for System z push-to-client
# give client administrator access
uniqueMember: racfID=RDZADM1,profileType=user,cn=RACF,o=IBM,c=US

```

Entwickler zu LDAP-Gruppen hinzufügen

Entwickler werden den LDAP-Gruppenobjekten hinzugefügt. Die Benutzerdefinitionen für die Benutzer-IDs werden mittels Pull-Verfahren aus dem RACF-Schema entnommen.

```
ldapmodify -D "cn=LDAP admin" -w password -f /u/ibmuser/ptc_add.ldif
```

Dabei enthält /u/ibmuser/ptc_add.ldif Folgendes:

```

# banking workspace configuration
dn: cn=FEK.PTC.CONFIG.ENABLED.CDFMVS08.BANKING,o=PTC,c=DeveloperForZ
changeType: modify
add: uniqueMember
uniqueMember: racfID=BNK010,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=BNK011,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=BNK012,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=BNK013,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=BNK014,profileType=user,cn=RACF,o=IBM,c=US

# insurance workspace configuration
dn: cn=FEK.PTC.CONFIG.ENABLED.CDFMVS08.INSURANCE,o=PTC,c=DeveloperForZ
changeType: modify
add: uniqueMember
uniqueMember: racfID=INS010,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=INS011,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=INS012,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=INS013,profileType=user,cn=RACF,o=IBM,c=US
uniqueMember: racfID=INS014,profileType=user,cn=RACF,o=IBM,c=US

```

pushtoclient.properties

```

# BANKING and INSURANCE have different configuration needs
config.enabled=LDAP
# everyone receives product updates
product.enabled=TRUE
# only RDZADMIN can reject configuration updates
reject.config.updates=LDAP
# only RDZADMIN can reject product updates
reject.product.updates=LDAP

```

rse.envvars

Es sind keine Aktualisierungen erforderlich, da die Standardwerte verwendet werden:

- `_RSE_LDAP_SERVER=CDFMVS08.RALEIGH.IBM.COM`
- `_RSE_LDAP_PORT=389`
- `_RSE_LDAP_PTC_GROUP_SUFFIX="o=PTC,c=DeveloperForZ"`

/var/rdz/pushtoclient/*install

Beim Export der Arbeitsbereichskonfiguration der Gruppen BANKING und INSURANCE erstellt der Exportassistent die Verzeichnisse `/var/rdz/pushtoclient/grouping/<devgroup>/` sowie die darunter liegende Verzeichnisstruktur.

- `/var/rdz/pushtoclient/grouping/BANKING/*`
- `/var/rdz/pushtoclient/grouping/INSURANCE/*`

Da keine individuellen Produktaktualisierungsszenarien vorliegen, braucht der Clientadministrator die Unterverzeichnisse `install/` und `install/responsefiles/` in `/var/rdz/pushtoclient/grouping/<devgroup>` weder zu erstellen noch zu aktualisieren.

Der Clientadministrator muss die für Produktaktualisierungen erforderlichen Antwortdateien im Verzeichnis `/var/rdz/pushtoclient/install/responsefiles/` der Standardgruppe erstellen.

SAF-basierte Gruppenauswahl

SAF (Security Access Facility) ist eine Schnittstelle, über die auf jedes z/OS-Sicherheitsprodukt zugegriffen werden kann. Developer for System z kann diese Schnittstelle für die Abfrage Ihres Sicherheitsprodukts und das Abrufen Push-to-Client-relevanter Informationen verwenden.

Wenn Definitionen Ihrer Sicherheitsdatenbank als Auswahlmechanismus verwendet werden (der SAF-Wert wird für Direktiven in `pushtoclient.properties` angegeben), überprüft Developer for System z die Zugriffserlaubnis auf die in Tabelle 37 aufgelisteten Profile, um festzustellen, zu welchen Entwicklergruppen ein Benutzer gehört und ob der Benutzer Aktualisierungen zurückweisen darf.

Tabelle 37. Push-to-Client-relevante SAF-Informationen

FACILITY-Profil	Feste Länge	Erforderlicher Zugriff	Ergebnis
FEK.PTC.CONFIG.ENABLED. sysname.devgroup	23	READ	Der Client akzeptiert Konfigurationsaktualisierungen für die angegebene Gruppe.
FEK.PTC.PRODUCT.ENABLED. sysname.devgroup	24	READ	Der Client akzeptiert Produktaktualisierungen für die angegebene Gruppe.
FEK.PTC.REJECT.CONFIG. UPDATES.sysname	30	READ	Der Benutzer kann Konfigurationsaktualisierungen zurückweisen.

Tabelle 37. Push-to-Client-relevante SAF-Informationen (Forts.)

FACILITY-Profil	Feste Länge	Erforderlicher Zugriff	Ergebnis
FEK.PTC.REJECT.PRODUCT. UPDATES.sysname	31	READ	Der Benutzer kann Produktaktualisierungen zurückweisen.

Anmerkung: Developer for System z geht davon aus, dass ein Benutzer über keine Zugriffsberechtigung verfügt, wenn die Sicherheitssoftware nicht feststellen kann, ob der Benutzer über die Berechtigung für den Zugriff auf ein Profil verfügt. Dies ist zum Beispiel der Fall, wenn das Profil gar nicht definiert ist.

Der Wert von devgroup ist der Gruppenname, der einer Entwicklergruppe zugewiesen ist. Dieser Gruppenname ist auf Developer for System z-Clients sichtbar.

Der Wert von sysname ist der Systemname des Zielsystems.

In der Spalte 'Feste Länge' ist die Länge des festen Teils des zugehörigen Sicherheitsprofils angegeben.

Developer for System z erwartet standardmäßig, dass FEK.*-Profile der Sicherheitsklasse FACILITY angehören. Für Profile der Klasse FACILITY gilt eine Beschränkung auf 39 Zeichen. Falls die Länge des festen Profiltails (FEK.PTC.<key>) und die Länge des standortspezifischen Profiltails (sysname oder sysname.devgroup) diesen Wert überschreiten, können Sie die Profile in einer anderen Klasse speichern und Developer for System z anweisen, diese Klasse zu verwenden. Dazu müssen Sie in rsed.envvars das Kommentarzeichen vor _RSE_FEK_SAF_CLASS entfernen und den Namen der gewünschten Klasse angeben.

Beispielkonfiguration

Developer for System z wird in einem Unternehmen auf einem System namens CDFMVS08 ausgeführt. Die RACF-Sicherheitsdatenbank wird von mehreren Systemen gemeinsam verwendet. Darin sind folgende Gruppen definiert:

- DEVBANK: Entwickler für Finanzanwendungen
- DEVINSUR: Entwickler für Versicherungsanwendungen
- RDZADMIN: Administratoren für Developer for System z-Clients

Jede Entwicklergruppe benötigt spezielle Clientkonfigurationsdateien, alle Entwickler unterliegen aber der gleichen Clientversionssteuerung. Im Gegensatz zu Clientadministratoren dürfen Entwickler die von Push-to-Client bereitgestellten Änderungen nicht zurückweisen. Diese Zurückweisungsregel wurde als Vorbereitung auf künftige Erweiterungen für alle Systeme eingeführt.

Der Client- und der Sicherheitsadministrator einigen sich für Konfigurationsaktualisierungen auf die Gruppennamen BANKING und INSURANCE.

Sicherheitsdefinition

```
# allow RDZADMIN and DEVBANK to select push-to-client group BANKING
RDEFINE FACILITY (FEK.PTC.CONFIG.ENABLED.CDFMVS08.BANKING) -
  UACC(NONE) DATA('RATIONAL DEVELOPER FOR SYSTEM Z - PUSH-TO-CLIENT')
PERMIT FEK.PTC.CONFIG.ENABLED.CDFMVS08.BANKING CLASS(FACILITY) -
  ID(RDZADMIN DEVBANK) ACCESS(READ)
```



```
# allow RDZADMIN and DEVINSUR to select push-to-client group INSURANCE
RDEFINE FACILITY (FEK.PTC.CONFIG.ENABLED.CDFMVS08.INSURANCE) -
  UACC(NONE) DATA('RATIONAL DEVELOPER FOR SYSTEM Z - PUSH-TO-CLIENT')
PERMIT FEK.PTC.CONFIG.ENABLED.CDFMVS08.INSURANCE CLASS(FACILITY) -
  ID(RDZADMIN DEVINSUR) ACCESS(READ)

# RDZADMIN can reject configuration updates on any system
RDEFINE FACILITY (FEK.PTC.REJECT.CONFIG.UPDATES.*) -
  UACC(NONE) DATA('RATIONAL DEVELOPER FOR SYSTEM Z - PUSH-TO-CLIENT')
PERMIT FEK.PTC.REJECT.CONFIG.UPDATES.* CLASS(FACILITY) -
  ID(RDZADMIN) ACCESS(READ)

# RDZADMIN can reject product updates on any system
RDEFINE FACILITY (FEK.PTC.REJECT.PRODUCT.UPDATES.*) -
  UACC(NONE) DATA('RATIONAL DEVELOPER FOR SYSTEM Z - PUSH-TO-CLIENT')
PERMIT FEK.PTC.REJECT.CONFIG.UPDATES.* CLASS(FACILITY) -
  ID(RDZADMIN) ACCESS(READ)

# activate changes
SETROPTS RACLIST(FACILITY) REFRESH
```

pushtoclient.properties

```
# BANKING and INSURANCE have different configuration needs
config.enabled=SAF
# everyone receives product updates
product.enabled=TRUE
# only RDZADMIN can reject configuration updates
reject.config.updates=SAF
# only RDZADMIN can reject product updates
reject.product.updates=SAF
```

rsed.envvars

Es sind keine Aktualisierungen erforderlich, da die Standardwerte verwendet werden:

```
_RSE_FEK_SAF_CLASS=FACILITY
```

/var/rdz/pushtoclient/*install

Beim Export der Arbeitsbereichskonfiguration der Gruppen BANKING und INSURANCE erstellt der Exportassistent die Verzeichnisse `/var/rdz/pushtoclient/grouping/<devgroup>` sowie die darunter liegende Verzeichnisstruktur.

- `var/rdz/pushtoclient/grouping/BANKING/*`
- `/var/rdz/pushtoclient/grouping/INSURANCE/*`

Da keine individuellen Produktaktualisierungsszenarien vorliegen, braucht der Clientadministrator die Unterverzeichnisse `install/` und `install/responsefiles/` in `/var/rdz/pushtoclient/grouping/<devgroup>` weder zu erstellen noch zu aktualisieren.

Der Clientadministrator muss die für Produktaktualisierungen erforderlichen Antwortdateien im Verzeichnis `/var/rdz/pushtoclient/install/responsefiles/` der Standardgruppe erstellen.

Karenzzeit für die Zurückweisung von Änderungen

Stellen Sie sich vor, dass während der Verwendung der Musterkonfiguration ein Developer for System z-Fixpack mit wichtigen Programmkorrekturen verfügbar wird, der Zeitplan eines Finanzprojekts aber derart kritisch ist, dass einige Entwickler zu diesem Zeitpunkt nur ungern Änderungen an ihren Workstations vornehmen wollen.

Dieses Problem kann der Sicherheitsadministrator ganz einfach lösen, indem er zum Beispiel allen DEVBANK-Entwicklern eine Karenzzeit einräumt, innerhalb der sie die Aktualisierung verschieben (also zunächst zurückweisen) können.

Die Einrichtung einer Karenzzeit ist völlig unkompliziert:

```
# start of grace period
PERMIT FEK.PTC.REJECT.PRODUCT.UPDATES.* CLASS(FACILITY) -
    ID(DEVBANK) ACCESS(READ)

# activate changes
SETROPTS RACLIST(FACILITY) REFRESH
```

Am Ende der Karenzzeit kann die zusätzliche Berechtigung wieder zurückgenommen werden:

```
# end of grace period
PERMIT FEK.PTC.REJECT.PRODUCT.UPDATES.* CLASS(FACILITY) -
    ID(DEVBANK) DELETE

# activate changes
SETROPTS RACLIST(FACILITY) REFRESH
```

Hostbasierte Projekte

z/OS-Projekte können einzeln auf dem Client in der Perspektive für z/OS-Projekte erstellt werden. Sie können aber auch zentral auf dem Host erstellt werden, in welchem Fall sie auf Benutzerbasis auf den Client repliziert werden. Solche hostbasierten Projekte sind hinsichtlich Aussehen und Funktionsweise 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.

Das Basisverzeichnis für hostbasierte Projekte wird vom Clientadministrator in `/var/rdz/pushtoclient/keymapping.xml` definiert. Standardmäßig lautet es `/var/rdz/pushtoclient/projects`.

Zur Konfiguration hostbasierter Projekte definiert der Projektmanager bzw. der leitende Entwickler die folgenden Typen von Konfigurationsdateien. Bei all diesen Dateien handelt es sich um UTF-8-codierte XML-Dateien.

- Projektinstanzdateien sind spezifisch für bestimmte Benutzer-IDs und verweisen auf wiederverwendbare Projektdefinitionsdateien. Jeder Benutzer, der mit hostbasierten Projekten arbeitet, benötigt sein eigenes Unterverzeichnis `/var/rdz/pushtoclient/projects/<userid>/` mit einer Projektinstanzdatei (`*.hbpin`) für jedes herunterzuladende Projekt.
- Projektdefinitionsdateien legen die Struktur und den Inhalt eines Projekts fest und sind für mehrere Benutzer wiederverwendbar. Diese Dateien (`*.hbppd`) listen die im Projekt enthaltenen Unterprojekte auf. Sie befinden sich im Verzeichnis mit der Stammpjektdefinition oder in einem seiner Unterverzeichnisse.
- Unterprojektdefinitionsdateien legen die Struktur und den Inhalt eines Unterprojekts fest und sind für mehrere Benutzer wiederverwendbar. Diese Dateien (`*.hbpsd`) legen den für die Erstellung eines einzelnen Lademoduls erforderlichen Ressourcensatz fest. Sie befinden sich im Verzeichnis mit der Stammpjektdefinition oder in einem seiner Unterverzeichnisse.
- Unterprojekteigenschaftendateien sind Eigenschaftendateien mit Unterstützung für Variablensubstitution. Sie können daher von mehreren Unterprojekten verwendet werden. Diese Dateien (`*.hbppr`) unterstützen die Variablensubstitution und ermöglichen somit die gemeinsame Verwendung der Eigenschaftendateien

durch mehrere Benutzer. Sie befinden sich im Verzeichnis mit der Stammprojektdefinition oder in einem seiner Unterverzeichnisse.

Hostbasierte Projekte können auch in die im Abschnitt „Mehrere Entwicklergruppen“ auf Seite 131 beschriebene Konfiguration für mehrere Gruppen integriert werden. Dies bedeutet, dass hostbasierte Projekte auch in `/var/rdz/pushtoclient/grouping/<devgroup>/projects/` definiert werden können.

Wenn ein Arbeitsbereich an eine bestimmte Gruppe gebunden ist und für einen Benutzer dieser Gruppe und in der Standardgruppe Projektdefinitionen vorliegen, erhält der Benutzer die Projektdefinitionen sowohl aus der Standardgruppe als auch aus der spezifischen Gruppe.

Kapitel 8. 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 insofern, als CICS-Administratoren die Möglichkeit haben, die Standardwerte für CICS-Ressourcendefinitionen zu kontrollieren und die Anzeigemerkmale von CICS-Ressourcendefinitionsparametern mithilfe des CICS Resource Definition (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 "Application Deployment Manager (optional)" in *Hostkonfiguration* (SC12-4062).

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 CICS Resource Definition (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://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/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 CICS Resource Definition (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.

Installation von CICS-Ressourcen 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 54.

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 zur Anpassung von Transaktions-IDs finden Sie im Abschnitt "Application Deployment Manager (optional)" im Handbuch *Hostkonfiguration* (SC12-4062).

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 Definition TRANSACTION 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 CICSTS*.

- 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 TCPIPService 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 finden Sie in "Angepasstes Setup" in *Hostkonfiguration* (SC12-4062).

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. Zwischenleerschritte 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.ADNREPF0
//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 DSNAME() 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) JNLSYNCWRITE(YES)
    RECOVERY(NONE) FWDRECOVLOG(NO)
    BACKUPTYPE(STATIC)
    PASSWORD() NSRGROUP()
    CFDTPOOL() TABLENAME()

```

Abbildung 27. ADNJSPAU - CICS-Verwaltungsdienstprogramm

```

*
* 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)
           EXECKEY(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() REMOTELength(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 28. ADNJSAPU - 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) STORAGEECLEAR(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 29. ADN/JSPAU - 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 URIMAP 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 URIMAP 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 URIMAP 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 <Zeilennummer der letzten Steueranweisung>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer wurde erfolgreich beendet.

Benutzeraktion: Keine

CRAZ1801W

completed with warnings on line <Zeilennummer der letzten Steueranweisung>

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 <Zeilennummer>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat einen schwerwiegenden Fehler festgestellt.

Benutzeraktion: Überprüfen Sie die weiteren Warnungen.

CRAZ1803E

Repository open error, status=<Dateistatuscode> RC=<VSAM-Rückkehrcode> FC=<VSAM-Funktionscode> FB=<VSAM-Rückkopplungscode>

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 <Zeilennummer>

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 <Schlüsselwort> on line <Zeilennummer>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer verarbeitet die Eingabesteueranweisung (Schlüsselwort **DEFINE**).

Benutzeraktion: Keine

CRAZ1806E

Invalid manifest export rule on line <Zeilennummer>

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 <Zeilennummer>

Erläuterung: Das Verwaltungsdienstprogramm für Systemprogrammierer hat eine Steueranweisung **DEFINE DSBINDING** verarbeitet, bei der das Schlüsselwort **DSNAME** fehlt.

Benutzeraktion: Überprüfen Sie, ob die Steueranweisung **DEFINE DSBINDING** das Schlüsselwort **DSNAME** enthält.

CRAZ1808E

Invalid keyword value for keyword <Schlüsselwort> on line <Zeilennummer>

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 <Zeilennummer>

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=<Dateistatuscode>

RC=<VSAM-Rückkehrcode> FC=<VSAM-Funktionscode> FB=<VSAM-Rückkopplungscode>

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=<Dateistatuscode> RC=<VSAM-Rückkehrcode> FC=<VSAM-Funktionscode> FB=<VSAM-Rückkopplungscode>

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=<Dateistatuscode> RC=<VSAM-Rückkehrcode> FC=<VSAM-Funktionscode> FB=<VSAM-Rückkopplungscode>

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.

CICS-Transaktionsdebugging

Um ein Debugging für CICS-Transaktionen durchzuführen, benötigt Integrated Debugger die folgenden CICS-Aktualisierungen:

- CICS-JCL-Aktualisierung:
 - Definieren Sie die Ladebibliothek FEK,SFEKAUTH in der DFHRPL DD-Anweisung der Region, wenn die Bibliothek sich nicht in der LINKLIST befindet.
 - Definieren Sie die Ladebibliothek SYS1.SIEAMIGE in der STEPLIB DD-Anweisung der Region, wenn die Bibliothek sich nicht in der LINKLIST befindet.
- CICS-CSD-Aktualisierungen:

Definieren Sie den Debugger für eine CICS-Region, wie im Beispiel-CSD-Aktualisierungsjob AQECSD dokumentiert. AQECSD 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 Informationen finden Sie unter "Anpassungskonfiguration" im Handbuch *Hostkonfiguration* (IBM Form SC12-4062).

Um ein Debugging für CICS-Transaktionen durchzuführen, die in den Nur-Lese-Speicher geladen wurden, benötigt Integrated Debugger die folgenden Systemaktualisierungen:

- Integrated Debugger-Supervisoraufruf (SVC), der für Ihr System definiert wurde. Weitere Informationen finden Sie unter "PARMLIB-Änderungen" im Handbuch *Hostkonfiguration* (IBM Form SC12-4062).
- Für den SVC wird vorausgesetzt, dass Benutzer Zugriff auf ein Sicherheitsprofil haben, wenn es in einer Fehlerstatus-Umgebung (nicht autorisiert) verwendet wird. Ausführliche Informationen hierzu enthält der Abschnitt „Debug-Sicherheit“ auf Seite 39.

Beachten Sie, dass nur ein Language Environment (LE)-basierter Debugger in einer angegebenen CICS-Region aktiv sein kann. Ein deutlicher Hinweis auf einen LE-basierten Debugger ist die Bereitstellung eines CEEVDBG-Lademoduls oder -Alias, das bzw. der der Anwendung zur Verfügung stehen muss.

Kapitel 9. Hinweise zu Benutzerexits

In diesem Kapitel finden Sie Informationen dazu, wie Sie Exitroutinen schreiben können, die Developer for System z funktional erweitern.

Developer for System z stellt Exitpunkte für ausgewählte Developer for System z-Ereignisse bereit. Ein Exitpunkt ist ein bestimmter Punkt in der Verarbeitung einer Funktion, an dem die Funktion eine Exitroutine aufruft, sofern eine solche vorhanden ist. Sie können eine Exitroutine schreiben, um eine zusätzliche Verarbeitung auszuführen.

Beachten Sie, dass es bei Developer for System z-Exitpunkten anders als bei den meisten üblichen Exitpunkten nicht zulässig ist, das Verhalten der Funktion zu ändern. Falls eine Exitroutine vorhanden ist, wird diese asynchron aufgerufen, nachdem die Funktion beendet wurde. Die Verarbeitung von Developer for System z wartet nicht auf das Beenden der Exitroutine, auch wird der Fertigstellungsstatus nicht überprüft.

Merkmale von Benutzerexits

Aktivierung von Benutzerexits

Benutzerexits werden mit den `_RSE_JVAOPTS <exit_point>.action`-Variablen in `rsed.envvars` aktiviert. Dabei stellt `<exit_point>` ein Schlüsselwort dar, das einen bestimmten Exitpunkt angibt, wie in „Verfügbare Exitpunkte“ auf Seite 162 dokumentiert.

```
#_RSE_JVAOPTS="$_RSE_JVAOPTS -D<exit_point>.action=<user_exit>"
```

Alle Exitpunkte sind standardmäßig inaktiviert. Zum Aktivieren des Exitpunkts entfernen Sie die Kommentarzeichen und geben Sie den vollständigen Pfadnamen der Benutzerexitroutine an.

```
#_RSE_JVAOPTS="$_RSE_JVAOPTS -D<exit_point>.action.id=<userid>"
```

Die dem RSE-Dämon zugeordnete Benutzer-ID wird standardmäßig zum Ausführen der bereitgestellten Exitroutine verwendet. Entfernen Sie die Kommentarzeichen und geben Sie eine Benutzer-ID an, um die angegebene Benutzer-ID zum Ausführen des Benutzerexits zu verwenden. Sie müssen kein Kennwort angeben, da RSE ein PassTicket erstellt, das beim Wechsel zur angegebenen Benutzer-ID als Kennwort verwendet wird.

Benutzerexitroutine schreiben

Benutzerexitroutinen werden als z/OS UNIX-Shellbefehl mit mindestens einem Argument aufgerufen. Dies bedeutet, dass die von Ihnen entwickelte Exitroutine über die z/OS UNIX-Befehlszeile ausführbar sein muss. Zu den üblichen Codierungsverfahren gehören das z/OS UNIX-Shell-Script und die z/OS UNIX-REXX-Exec, aber auch kompilierter Code wie C/C++ ist möglich.

Im Benutzerhandbuch *UNIX System Services User's Guide* (IBM Form SA22-7801) finden Sie weitere Informationen zu z/OS UNIX-Shell-Skripts. Weitere Informationen zu z/OS UNIX-spezifischen Erweiterungen der REXX-Sprache enthält die Dokumentation *Using REXX and z/OS UNIX System Services* (IBM Form SA22-7806).

Die Exitroutine wird in der Regel von einer Benutzer-ID mit besonderen Berechtigungen ausgeführt (wie z. B. der Benutzer-ID der gestarteten RSE-Task, für die das Erstellen von PassTickets zulässig ist). Daher ist es wichtig, dass Sie die Aktualisierungsberechtigung für die Exitroutine einschränken, um Missbrauch zu vermeiden. Der folgende Beispiel-z/OS UNIX-Befehl schränkt die Schreibberechtigung auf den Eigner ein, während das Skript von allen gelesen und ausgeführt werden kann.

```
$ chmod 755 process_logon.sh
$ ls -l process_logon.sh
-rwxr-xr-x  1 IBMUSER  SYS1          2228 Feb 28 23:44 process_logon.sh
```

Definitionen in `rsed.envvars` stehen der Benutzerexitroutine als Umgebungsvariablen zur Verfügung.

RSE ruft die Benutzerexitroutine mithilfe einer einzelnen Argumentenfolge auf. Die Argumentenfolge kann aus einem einzigen Wert oder einer einzelnen Zeichenfolge bestehen, die mehrere leere Schlüsselwörter und Werte mit Begrenzern enthält. Ausführliche Informationen hierzu enthält der Abschnitt „Verfügbare Exitpunkte“ auf Seite 162.

Konsolennachrichten

Developer for System z verwendet die Konsolennachricht-ID FEK910I, um Daten anzuzeigen, die zu Benutzerexits zugehörig sind.

Der Aufruf der Exitroutine wird durch die folgende Konsolennachricht markiert:

```
FEK910I <EXIT_POINT> EXIT: invoking <exit_point> processing exit
      in thread <thread_id>
```

Sämtliche Daten, die an `stdout` (Befehl **echo** in einem Shell-Skript, Befehl **say** in einer REXX-Exec) geschrieben werden, werden an die Konsole gesendet:

```
FEK910I <EXIT_POINT> EXIT: <message>
```

Die Beendigung der Exitroutine wird durch die folgende Konsolennachricht markiert:

```
FEK910I <EXIT_POINT> EXIT: completed <exit_point> processing exit
      in thread <thread_id>
```

Ausführung mithilfe einer variablen Benutzer-ID

Developer for System z ermöglicht Ihnen, eine Exitroutine sowohl mithilfe der Benutzer-ID der gestarteten Task als auch mit einer angegebenen Benutzer-ID auszuführen. Möglicherweise möchten Sie jedoch einige Aktionen in der Exitroutine mithilfe einer weiteren Benutzer-ID ausführen, wie z. B. der Client-Benutzer-ID in der Exitroutine 'logon'. Führen Sie dies aus, indem Sie die z/OS UNIX-Standardservices verwenden, wie in den folgenden Beispielen gezeigt.

z/OS UNIX-Shell-Script

Wie in der Veröffentlichung *UNIX System Services Command Reference* (IBM Form SA22-7802) dokumentiert, stellt z/OS UNIX den Befehl **su** bereit, um die Zugriffsrechte eines Superusers oder eines anderen Benutzers zu verwenden. Für die Verwendung des Befehls **su** sind einige Dinge zu beachten.

- Die Benutzer-ID, mit der der Befehl **su** ausgeführt wird, muss für das Profil BPX.SRV.<userid> in der Klasse SURROGAT Ihres Sicherheitsprodukts über die Berechtigung READ verfügen, damit ein Wechsel zu der von <userid> ermittelten Benutzer-ID ohne Angabe eines Kennworts möglich ist.
- Mit dem Befehl **su** wird eine neue Shell gestartet, sodass die verbleibenden Befehle in Ihrem Shell-Script erst ausgeführt werden, wenn die durch den Befehl **su** gestartete Shell beendet wurde. Zum schrittweisen Ausführen von Befehlen, die in der neuen Shell, die durch den Befehl **su** gestartet wurde, ausgeführt werden sollen, können Sie den Befehl **echo** verwenden, um den gewünschten Befehl sowie das Befehlszeichen der Pipe zum Weiterleiten des Befehls an die neue Shell, wie im folgenden Beispiel gezeigt. Beachten Sie, dass zum Umgehen von Sonderzeichen Standardrichtlinien für die Shell-Scripterstellung gelten.

```
#!/bin/sh
myID=ibmuser
echo a ${id}
echo 'echo b ${id}' | su -s $myID
echo "echo c \${id}" | su -s $myID
cat /u/ibmuser/iefbr14
echo "submit /u/ibmuser/iefbr14" | su -s $myID
```

Dieser Beispiexit 'logon', der von der Benutzer-ID der gestarteten Task ausgeführt wurde, führt zur Ausgabe der folgenden Konsolennachricht:

```
+FEK910I LOGON EXIT: invoking logon processing exit in thread 411
+FEK910I LOGON EXIT: a uid=8(STCRSE) gid=1(STCGROUP)
+FEK910I LOGON EXIT: b uid=1(IBMUSER) gid=0(SYS1)
+FEK910I LOGON EXIT: c uid=1(IBMUSER) gid=0(SYS1)
+FEK910I LOGON EXIT: //IEFBR14 JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
+FEK910I LOGON EXIT: //IEFBR14 EXEC PGM=IEFBR14
$HASP100 IEFBR14 ON INTRDR FROM STC03919
IBMUSER
IRR010I USERID IBMUSER IS ASSIGNED TO THIS JOB.
+FEK910I LOGON EXIT: JOB JOB03926 submitted from path '/u/ibmuser/iefbr14'
ICH70001I IBMUSER LAST ACCESS AT 00:46:13 ON MONDAY, MARCH 19, 2012
$HASP373 IEFBR14 STARTED - INIT 2 - CLASS A - SYS CD08
IEF403I IEFBR14 - STARTED - TIME=00.46.14
+FEK910I LOGON EXIT: completed logon processing exit in thread 411
IEFBR14 IEFBR14 IEFBR14 0000
IEF404I IEFBR14 - ENDED - TIME=00.46.14
$HASP395 IEFBR14 ENDED
$HASP309 INIT 2 INACTIVE ***** C=BA
```

z/OS UNIX-REXX-Exec

Wie in *Using REXX and z/OS UNIX System Services* (IBM Form SA22-7806) dokumentiert, stellt z/OS UNIX den SYSCALL-Befehl **seteuuid** bereit, um die ausführende Benutzer-ID (UID) des aktuellen Prozesses festzulegen. Für die Verwendung des Befehls **seteuuid** sind einige Dinge zu beachten.

- Der Befehl **seteuuid** verwendet die z/OS UNIX-Benutzer-ID, nicht die MVS-Benutzer-ID. Sie müssen zunächst die Benutzer-ID (UID) der Zielbenutzer-ID ermitteln. Dies können Sie mit dem SYSCALL-Befehl **getpwnam** ausführen.

- Die Benutzer-ID, mit der der Befehl **seteuid** ausgeführt wird, muss für das Profil BPX.SRV.<userid> in der Klasse SURROGAT Ihres Sicherheitsprodukts über die Berechtigung READ verfügen, damit ein Wechsel zu der von <userid> ermittelten Benutzer-ID ohne Angabe eines Kennworts möglich ist. Wenn mehrere Benutzer-IDs dieselbe Benutzer-ID (UID) gemeinsam nutzen, müssen Sie beachten, dass es keine Möglichkeit gibt festzustellen, welche der Benutzer-IDs überprüft wird.

```
/* rexx */
myID='ibmuser'
say userid()
address SYSCALL 'getpwnam' myID 'pw.'
say pw.1 pw.2 pw.3 pw.4 pw.5
address SYSCALL 'seteuid' pw.2 /* PW_UID = 2 */
say retval errno errnojr
say userid()
```

Dieser Beispiexit 'logon', der von der Benutzer-ID der gestarteten Task ausgeführt wurde, führt zur Ausgabe der folgenden Konsolennachricht:

```
+FEK910I LOGON EXIT: invoking logon processing exit in thread 515
+FEK910I LOGON EXIT: STCRSE
+FEK910I LOGON EXIT: IBMUSER 1 0 / /bin/sh
+FEK910I LOGON EXIT: 0 0 0
+FEK910I LOGON EXIT: IBMUSER
+FEK910I LOGON EXIT: completed logon processing exit in thread 515
```

Verfügbare Exitpunkte

Folgende Exitpunkte werden von Developer for System z bereitgestellt:

- „audit.action“
- „logon.action“ auf Seite 163

audit.action

- **Zeitablauf:**

Der Benutzerexit 'audit' wird aufgerufen, wenn die aktive Prüfprotokolldatei geschlossen ist. (Die Prüfung wird fortgeführt, da RSE zu einer neuen Prüfprotokolldatei gewechselt hat.)

- **Aufrufargumente (1):**

- <audit_log>: Vollständiger Pfadname der Prüfprotokolldatei, die geschlossen wurde

- **Beispiel:**

```
/usr/lpp/rdz/samples/process_audit.rex
```

Dieses Beispiel für einen z/OS UNIX-REXX-Exec erstellt einen Batch-Job, der das geschlossene Prüfprotokoll verarbeiten wird.

logon.action

- **Zeitablauf:**

Der Benutzerexit 'logon' wird aufgerufen, wenn ein Benutzer den Anmeldeprozess abgeschlossen hat.

- **Aufrufargumente (6):**

- -i <userid>: Client-Benutzer-ID, Groß-/Kleinschreibung wie durch den Client bereitgestellt
- -u <user_log_path>: Verzeichnis, in dem die Benutzerprotokolle dieses Clients vorhanden sind
- -s <server_log_path>: Verzeichnis, in dem die Serverprotokolle vorhanden sind
- -c <config_path>: Verzeichnis, in dem die Konfigurationsdateien vorhanden sind
- -b <binaries_path>: Verzeichnis, in dem Developer for System z installiert wurde
- -p <port>: Port des RSE-Dämons

- **Beispiel:**

/usr/lpp/rdz/samples/process_logon.sh

Mit diesem Beispiel für ein z/OS UNIX-Shell-Script wird eine Anmeldenachricht an die Konsole geschrieben.

Kapitel 10. 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.

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) Diese Methode wird nicht weiter unterstützt.

Anmerkung:

- Der TSO/ISPF-Client-Gateway-Service von ISPF ersetzt die in Version 7.1 verwendete Funktion von SCLM Developer Toolkit.
- Die APPC-Verwendung durch Developer for System z ist als nicht weiter unterstützt markiert. Die zugehörigen APPC-Informationen wurden aus dieser Veröffentlichung entfernt. Weitere Informationen finden Sie im White Paper *Using APPC to provide TSO command services* (SC14-7291), das in der Bibliothek für Developer for System z unter <http://www-01.ibm.com/support/docview.wss?uid=swg27038517> verfügbar ist.

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.

- Wenn die Anweisung `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` nicht vorhanden (oder auskommentiert) ist, wird der TSO/ISPF-Client-Gateway-Service von ISPF verwendet.
- Ist die Anweisung `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` vorhanden (und nicht auf Kommentar gesetzt), wird APPC verwendet.

TSO/ISPF-Client-Gateway als Zugriffsmethode verwenden

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.SISPLIB,FEK.SFEKPROC
isplib=ISP.SISPMENU
isptlib=ISP.SISPTEU
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`).

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_ISPF_OPTS` in `rsed.envvars`.

```
#_RSE_ISPF_OPTS="$_RSE_ISPF_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
- &SYSNAME. als Ersatz für den Systemnamen, wie im Parmlib-Member IEASYMxx angegeben

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.

Verwendung einer Zuordnungs-Exec

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 = ISP.SISPSAMP(ISPZISP2)
```

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 hierzu finden Sie in "Angepasstes Setup" in *Hostkonfiguration* (SC12-4062).

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.

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.

Mehrere 'ISPF.conf'-Dateien mit mehreren Developer for System z-Konfigurationen

Falls die zuvor beschriebenen Szenarien mit Zuordnungs-Execs Ihren Anforderungen nicht genügen, können Sie andere Instanzen des RSE-Kommunikationsservers von Developer for System z erstellen, wobei jede Instanz ihre eigene ISPF.conf-Datei verwendet. 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
-> change: _RSE_RSED_PORT=4037
-> change: CGI_ISPCONF=/etc/rdz/tso2
-> ändern: -Ddaemon.log=/var/rdz/logs/tso2
-> ändern: -Duser.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 CGI_ISPCONF in rsed.envvars muss aktualisiert werden, damit sie das neue Ausgangsverzeichnis in ISPF.conf definiert. Außerdem müssen daemon.log und user.log aktualisiert werden, um eine neue Protokollposition zu definieren. (Die Position wird automatisch erstellt, wenn sie nicht vorhanden ist.) Mit der Aktualisierung von _RSE_RSED_PORT wird sichergestellt, dass der vorhandene und der neue RSE-Dämon eindeutige Portnummern verwenden. 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 CGI_ISPWORK 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. Wenn _RSE_RSED_PORT nicht in rsed.envvars geändert wird, muss die neue gestartete Task einen neuen Port als Startargument angeben.

Weitere Informationen zu den zuvor in diesem Abschnitt beschriebenen Aktionen finden Sie im Handbuch *IBM Rational Developer for System z Hostkonfiguration* (IBM Form SC12-4062).

Kapitel 11. 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 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 (High Level Qualifier, HLQ) und der Pfad. FEK.#CUST, /etc/rdz und /var/rdz sind während der Anpassung des Produkts die verwendeten Standardpositionen. (Weitere Informationen hierzu finden Sie im Abschnitt "Angepasstes Setup" im Handbuch *Hostkonfiguration* (SC12-4062).)
- 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 verwenden. 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`

Identische Konfiguration in einem Sysplex

Konfigurationsdateien (und Code) für Developer for System z können auf verschiedenen Systemen in einem Sysplex gemeinsam genutzt werden, wobei jedes System dabei eine eigene identische Kopie von Developer for System z ausführt, sofern einige Richtlinien eingehalten werden. Beachten Sie, dass diese Informationen für eigenständige Instanzen von Developer for System z gelten. Bei der Verwendung von verteiltem dynamischen VIPA zum Gruppieren von mehreren Servern (jeweils auf einem separaten System) auf einem virtuellen Server gelten zusätzliche Regeln für die TCP/IP-Konfiguration, die unter „Verteilte dynamische VIPA“ auf Seite 66 dokumentiert sind.

- Die Protokolldateien sollten an eindeutigen Positionen gespeichert werden, damit ein System nicht die Informationen eines anderen Systems überschreibt. In-

dem 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.

- Konfigurationsähnliche Verzeichnisse wie `/etc/rdz/` und `/var/rdz/pushtoclient/` 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/`, müssen 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.
- Wenn Sie eine aktive `/etc/rdz/pushtoclient.properties`-Konfigurationsdatei gemeinsam nutzen, müssen Sie auch das zugehörige Metadatenverzeichnis `/var/rdz/pushtoclient/` gemeinsam nutzen.

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 vorangegangenen 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
  -> ändern: _RSE_RSED_PORT=4047
  -> ändern: _Ddaemon.log=/var/rdz/logs/ssl
  -> ändern: -Duser.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 Befehlen im vorherigen Beispiel werden die Konfigurationsdateien für Developer for System z, die geändert werden müssen, in ein neu erstelltes Verzeichnis `ssl` kopiert. Die Variablen `daemon.log` und `user.log` in `rsed.envvars` muss aktualisiert werden, um eine neue Protokollposition zu definieren. (Die Position wird automatisch erstellt, wenn sie noch nicht vorhanden ist.) Mit der Aktualisierung von `CLASSPATH` wird sichergestellt, 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 zuvor in diesem Abschnitt beschriebenen Aktionen enthalten die entsprechenden Abschnitte im Handbuch *IBM Rational Developer for System z Hostkonfiguration* (IBM Form SC12-4062).

Automatisierte Synchronisierung

In dem vorangegangenen SSL-Beispiel sind die Änderungen zwischen dem RSE-Dämon mit SSL-Aktivierung und dem RSE-Dämon ohne SSL minimal. Dies ermöglicht eine Automatisierung der Schritte für die Synchronisierung der entsprechenden `rsed.envvars`-Dateien. Hierdurch wird der Service-Rollout vereinfacht, da nur eine Datei `rsed.envvars` gepflegt und verwaltet werden muss.

Im folgenden Beispiel wird die RSED-Portnummer zu den Protokollverzeichnisnamen hinzugefügt. Außerdem wird der `CLASSPATH` aktualisiert, sodass Klone die übrigen Konfigurationsdateien finden können. Dann führt das Beispiel eine Erweiterung für die JCL der gestarteten Task des RSE-Dämons mit SSL-Aktivierung durch, um beim Startvorgang die Datei `rsed.envvars` des RSE-Dämons ohne SSL zu klonen und dabei die Portnummer zu aktualisieren. Da die Portnummer im Protokollverzeichnisnamen eingebettet ist, weicht sie für die beiden Dämonen jeweils ab.

1. Bereiten Sie die Masterdatei `rsed.envvars` vor.

```

$ oedit /etc/rdz/rsed.envvars
-> Ändern: -Ddaemon.log=/var/rdz/logs/$RSE_RSED_PORT
-> Ändern: -Duser.log=/var/rdz/logs/$RSE_RSED_PORT
-> Am ENDE anfügen:
# -- WIRD VON KLONEN BENÖTIGT, UM DIE ÜBRIGEN KONFIGURATIONSDATEIEN ZU FINDEN
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --

```

2. Bereiten Sie die übrigen Konfigurationsdateien vor (die keine `rsed.envvars`-Dateien sind), die von der Masterdatei (ohne SSL-Aktivierung) und dem Klon (SSL) abweichen.

```

$ mkdir /etc/rdz/ssl
$ cp /etc/rdz/ssl.properties /etc/rdz/etc/rdz/ssl
$ oedit /etc/rdz/ssl/ssl.properties
-> Ändern: nach Bedarf ändern

```

3. Erstellen Sie eine gestartete RSED-Task, die die `rsed.envvars`-Basisdatei kloniert und den RSE-Dämonport (4035 -> 4034) ändert.

```

/*
/* RSE-DÄMON - SSL
/*
//RSED      PROC IVP=,          * 'IVP' für einen IVP-Test
//          HOME='/usr/lpp/rdz',
//          CNFG='/etc/rdz/ssl'
/*
//          SET SED='"/RSED_PORT/s/4035/4034/'
//          SET FILE='rsed.envvars'
/*
/* Kopieren von /etc/rdz/rsed.envvars nach /etc/rdz/ssl/rsed.envvars
/* und Ändern von RSED_PORT
/*
//CLONE     EXEC PGM=BPXBATCH,REGION=0M,COND=(4,LT),
// PARM='SH cd &CNFG;sed &SED ../&FILE>&FILE'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
/*
/* Starten von RSED mit der neu erstellten Datei rsed.envvars
/*
//RSED      EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,COND=(4,LT),
// PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG'
//STDOUT    DD SYSOUT=*
//STDERR    DD SYSOUT=*
//          PEND
/*

```

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 *.conf
- 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 Verzeichnisse 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.
- 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* (ferne Build-prozeduren)" in *Hostkonfiguration* (SC12-4062) 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)" in *Hostkonfiguration* (SC12-4062) 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 12. 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 176
- „Speicherauszugsdateien“ auf Seite 181
- „Traceerstellung“ auf Seite 184
- „z/OS UNIX-Berechtigungsbits“ auf Seite 186
- „Reservierte TCP/IP-Ports“ auf Seite 190
- „Adressraum, Größe“ auf Seite 191
- „Sonstige Informationen“ auf Seite 193

In der Veröffentlichung *Developer for System z Messages and Codes* (IBM Form SC14-7497) werden Nachrichten und Rückkehrcodes dokumentiert, die von Developer for System z-Komponenten generiert wurden. *Developer for System z Antworten auf gängige Fragen der Hostkonfiguration und -wartung* (IBM Form SC12-4724) beschreibt verschiedene Problemszenarien und die Lösungen dazu.

Weitere Informationen sind im Bereich "Support" der Website zu Developer for System z (<http://www-03.ibm.com/software/products/us/en/developerforsystemz/>) verfügbar. Dort finden Sie die aktuellsten technischen Hinweise (Technotes) des Unterstützungsteams.

Die aktuellste Version der Dokumentation zu Developer for System z einschließlich White Papers finden Sie im Abschnitt "Library" der Website (<http://www-01.ibm.com/support/docview.wss?uid=swg27038517>).

Im Information Center für Developer for System z (<http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp>) ist der Developer for System z-Client und seine 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 finden Sie in "Angepasstes Setup" in *Hostkonfiguration* (SC12-4062).

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. Beachten Sie, dass die Ausgabedatei als VB 2051 zugeordnet werden muss (Standardwert in SDSF ist VB 240), um eine Verkürzung des Datensatzes zu vermeiden.

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
- .dstoreHashmap.* - Statische Sicht (Snapshot) der aktiven DataStore-Hashzuordnung
- .dstoreStackTrace.* - Statische Sicht (Snapshot) der aktiven DataStore-Threads und des Orts ihres Aufrufs
- 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.)

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 dem Developer for System z-Client arbeiten, wählen Sie **Fenster > Benutzervorgaben > Ferne Systeme > Dateien** aus und aktivieren Sie die Option 'Verdeckte Dateien anzeigen'.

- **daemon-home**

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

- **/tmp**

IVP-spezifische Protokolldateien (Installation Verification Program, Installationsprüfprogramm) befinden sich in dem Verzeichnis, das von `TMPDIR` referenziert wird, wenn diese Variable in `rsed.envvars` definiert ist. Wenn die Variable nicht definiert ist, werden die Dateien im Verzeichnis `/tmp` erstellt.

- `fekfivpi.log` - Protokoll des IVP-Tests `fekfivpi`
- `fekfivps.log` - Protokoll des IVP-Tests `fekfivps`
- `fekfivpc.log` - Kommunikationsprotokoll des IVP-Tests `fekfivpc`

Anmerkung: Es gibt einige Bedienerbefehle, mit denen das in einige der erwähnten Protokolldateien geschriebene Datenvolumen gesteuert werden kann. Weitere Informationen finden Sie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062).

JES Job Monitor, Protokollierung

- **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 „JES Job Monitor, Traceerstellung“ auf Seite 184.

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=*`.

- **daemon-home**

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.
- Es gibt einige Bedienerbefehle, mit denen das in einige der erwähnten Protokolldateien geschriebene Datenvolumen gesteuert werden kann. Weitere Informationen finden Sie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062).
- Die Dateien `rse*.log` können auch die Erweiterung `".last"` (statt `".log"`) haben, wenn `keep.last.log=true` in `rsed.envvars` angegeben wurde. Standardmäßig werden die Protokolldateien `".last"` nicht erstellt.
- Die Dateien `rse*.log` haben einen erweiterten Namen, wenn `keep.all.logs=true` in `rsed.envvars` angegeben wurde. Standardmäßig wird der erweiterte Name verwendet. Das folgende Beispiel ist ein erweiterter Name, bei dem "RSED" der Adressraumname des RSE-Dämons und `yyyymmddhhmmss` ein Datums- und Zeitstempel (Jahr, Monat, Tag, Stunde, Minute, Sekunde) ist: `rseserver.RSED#yyyymmddhhmmss.log`

RSE-Benutzer, Protokollierung

- **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
- .dstoreHashmap.* - Statische Sicht (Snapshot) der aktiven DataStore-Hashzuordnung
- .dstoreStackTrace.* - Statische Sicht (Snapshot) der aktiven DataStore-Threads und des Orts ihres Aufrufs
- 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.)

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 dem Developer for System z-Client arbeiten, wählen Sie **Fenster > Benutzervorgaben > Ferne Systeme > Dateien** aus und aktivieren Sie die Option 'Verdeckte Dateien anzeigen'.
- Die Erstellung der .dstore*-Protokolldateien wird von den Java-DDSTORE_*-Startoptionen gesteuert. Lesen Sie hierzu die Informationen im Abschnitt "Zusätzliche Java-Startparameter mit _RSE_JAVAOPTS definieren" im Handbuch *Hostkonfiguration* (SC12-4062).
- 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.
- Im Gegensatz zu allen Dateien vom Typ *.log werden Protokolldateien des Typs .dstore* nicht automatisch bei der Wiederherstellung der Verbindung zum Client entfernt. Das Entfernen dieser Dateien erfolgt manuell.
- Es gibt einige Bedienerbefehle, mit denen das in einige der erwähnten Protokolldateien geschriebene Datenvolumen gesteuert werden kann. Weitere Informationen finden Sie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062).
- Die Dateien ffs*.log, lock.log und rsecomm.log können auch die Erweiterung ".last" (statt ".log") haben, wenn keep.last.log=true in rsed.envvars angegeben wurde. Standardmäßig werden die Protokolldateien ".last" nicht erstellt.
- Die Dateien ffs*.log, lock.log und rsecomm.log haben einen erweiterten Namen, wenn keep.all.logs=true in rsed.envvars angegeben wurde. Standardmäßig wird der erweiterte Name verwendet. Das folgende Beispiel ist ein erweiterter Name, bei dem "RSEDx" der Adressraumname des Thread-Pools ist, in dem der Benutzer aktiv ist, und yyyymmddhhmmss ein Datums- und Zeitstempel (Jahr, Monat, Tag, Stunde, Minute, Sekunde) ist: ffs.RSEDx#yyyymmddhhmmss.log

SCLM Developer Toolkit, Protokollierung

- **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 DD-Karte des Server-Jobs enthält die CARMA-Protokollierung, sofern nicht die DD-Anweisung "CARMALOG" definiert ist.

- **SYSTSPRT DD**

Die SYSTSPRT DD-Karte des Server-Jobs enthält die Systemnachrichten (TSO) für den Start des CARMA-Servers.

- **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.

Protokollierung des IVP-Tests "fekfivpc"

- **/tmp/fekfivpc.log**

Der Befehl fekfivpc (zu CARMA gehörender IVP-Test) erstellt die Datei fekfivpc.log, um die Kommunikation zwischen RSE und CARMA zu dokumentieren. Das Protokoll wird in dem Verzeichnis erstellt, das von TMPDIR referenziert wird, wenn diese Variable in rsed.envvars definiert ist. Wenn die Variable nicht definiert ist, wird die Datei im Verzeichnis /tmp erstellt.

fekfivpi, Protokollierung des IVP-Tests

- **/tmp/fekfivpi.log**

Ausgabe des Befehls fekfivpi -file (zu TSO/ISPF-Client-Gateway gehörender IVP-Test). Das Protokoll wird in dem Verzeichnis erstellt, das von TMPDIR refe-

renziert wird, wenn diese Variable in `rsed.envvars` definiert ist. Wenn die Variable nicht definiert ist, wird die Datei im Verzeichnis `/tmp` erstellt.

Protokollierung des IVP-Tests fekfivps

- `/tmp/fekfivps.log`

Ausgabe des Befehls `fekfivps -file` (zu SCLMDT gehörender IVP-Test). Das Protokoll wird in dem Verzeichnis erstellt, das von `TMPDIR` referenziert wird, wenn diese Variable in `rsed.envvars` definiert ist. Wenn die Variable nicht definiert ist, wird die Datei im Verzeichnis `/tmp` erstellt.

Protokollierung der Codeüberprüfung

- SYSTSPRT DD

Das Element SYSTSPRT DD des Schritts, mit dem die Codeüberprüfungsprozedur aufgerufen wird, enthält die Nachrichten des Front-Ends, das den Codeanalyseprozess vorantreibt.

- WORKSPCE DD

Das Element WORKSPCE DD des Schritts, mit dem die Codeüberprüfungsprozedur aufgerufen wird, enthält die Protokollnachrichten vom Eclipse-Arbeitsbereich des Codeanalyseprozesses.

- ERRMSGs DD

Das Element ERRMSGs DD des Schritts, mit dem die Codeüberprüfungsprozedur aufgerufen wird, enthält die `stderr`-Ausgabe des Codeanalyseprozesses.

Protokollierung der Codeabdeckung

- SYSTSPRT DD

Das Element SYSTSPRT DD des Schritts, mit dem die Codeüberprüfungsprozedur aufgerufen wird, enthält die Nachrichten des Front-Ends, das den Codeanalyseprozess vorantreibt.

- WORKSPCE DD

Das Element WORKSPCE DD des Schritts, mit dem die Codeüberprüfungsprozedur aufgerufen wird, enthält die Protokollnachrichten vom Eclipse-Arbeitsbereich des Codeanalyseprozesses.

- ERRMSGs DD

Das Element ERRMSGs DD des Schritts, mit dem die Codeüberprüfungsprozedur aufgerufen wird, enthält die `stderr`-Ausgabe des Codeanalyseprozesses.

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. Eventuell werden die Speicherauszüge gar nicht oder nicht an den in den folgenden Abschnitten beschriebenen Positionen erstellt.

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_JVAOPTS` 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 Speicherausügen 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.

Anmerkung: Mit `JAVA_DUMP_TDUMP_PATTERN` können Variablen verwendet werden, die zum Zeitpunkt der Erstellung des Transaktionsspeicherauszugs in einen tatsächlichen Wert umgesetzt werden.

Tabelle 38. Variablen für `JAVA_DUMP_TDUMP_PATTERN`

Variable	Verwendung
<code>%uid</code>	Benutzer-ID
<code>%job</code>	Jobname
<code>%y</code>	Jahr (2-stellig)
<code>%m</code>	Monat (2-stellig)
<code>%d</code>	Tag (2-stellig)
<code>%H</code>	Stunde (2-stellig)
<code>%M</code>	Minute (2-stellig)
<code>%S</code>	Sekunde (2-stellig)

CEEDUMP

LE-Speicherauszug (Language Environment). Dies ist ein Systemspeicherauszug in einer formatierten Zusammenfassung, die die Stack-Traces für je-

den 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“ 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“ beschrieben.

Beachten Sie, dass Developer for System z einen Operatorbefehl bereitstellt, um diesen Speicherauszug auszulösen. Weitere Details finden Sie im Kapitel zu den Operatorbefehlen in der Veröffentlichung *Hostkonfiguration* (IBM Form SC12-4062).

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“ beschrieben.

Beachten Sie, dass Developer for System z einen Operatorbefehl bereitstellt, um diesen Speicherauszug auszulösen. Weitere Details finden Sie im Kapitel zu den Operatorbefehlen in der Veröffentlichung *Hostkonfiguration* (IBM Form SC12-4062).

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 zuvor genannten Verzeichnisse gespeichert werden kann, wird er an stderr gesendet.

Traceerstellung

JES Job Monitor, Traceerstellung

Die Traceerstellung für JES Job Monitor wird, wie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062) beschrieben, vom Systembediener gesteuert.

- Wenn Sie die gestartete Task JMON mit dem Parameter PRM=-TV starten, wird der ausführliche Trace-Modus aktiviert.
- Mit den Bedienerbefehlen **modify trace** und **modify message** können Sie die gewünschte Detailstufe (Detaillierungsgrad) für Protokollnachrichten auswählen.

RSE, Traceerstellung

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. Weitere Details finden Sie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062) und im Abschnitt "RSE-Traceerstellung (optional)" im Handbuch *Hostkonfiguration* (SC12-4062).

Die Erstellung der .dstore*-Protokolldateien wird von den Java-DDSTORE_*-Startoptionen gesteuert. Lesen Sie hierzu die Informationen im Abschnitt "Zusätzliche Java-Startparameter mit _RSE_JAVAOPTS definieren" im Handbuch *Hostkonfiguration* (SC12-4062).

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 dem Developer for System z-Client arbeiten, wählen Sie **Fenster > Benutzervorgaben > 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.
- Im Gegensatz zu allen Dateien vom Typ *.log werden Protokolldateien des Typs .dstore* nicht automatisch bei der Wiederherstellung der Verbindung zum Client entfernt. Das Entfernen dieser Dateien erfolgt manuell.

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 `rzed.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. Weitere Details finden Sie im Abschnitt "Bedienerbefehle" im Handbuch *Hostkonfiguration* (SC12-4062) und im Abschnitt "RSE-Traceerstellung (optional)" im Handbuch *Hostkonfiguration* (SC12-4062).

Die Dateien `serverlogs.count`, `stderr*.log` und `stdout*.log` werden nur erstellt, wenn die Anweisung `enable.standard.log` in `rzed.envvars` aktiv ist oder wenn die Funktion mit dem Bedienerbefehl **modify rsestandardlog on** dynamisch aktiviert wurde.

CARMA, Traceerstellung

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 176.

Fehlerrückmeldungen, Trace

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* (ferne Buildprozeduren)" in *Hostkonfiguration* (SC12-4062).
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 diesem Projekt die Dateien SIDEFILE1->4 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 her, indem Sie die Änderung rückgängig machen (die Zeichenfolge "MAXTRACE" aus den Kompilierungsoptionen entfernen) oder die Sicherungskopie zurückschreiben.
8. Senden Sie die gesammelten Dateien (SIDEFILE1->4 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.

SETUID, Dateisystemattribut

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. Es gibt weitere Anzeichen für dieses Setup-Problem:

- Konsolennachricht "FEK999E The module, fekfomvs must be marked as APF-authorized"
- PassTicket-IVP schlägt mit "ICH409I 282-010 ABEND DURING RACHECK PROCESSING" fehl

Ähnliche Fehler (wie beispielsweise die Nachrichten BPXP014I und BPXP015I) können auftreten, wenn die Dateisysteme, auf denen sich die Java- oder z/OS UNIX-Binärdateien befinden, mit dem Parameter NOSETUID angehängt werden.

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 2, „Sicherheitsaspekte“, auf Seite 19. 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
 - libicudata50.1.dll
 - libicudata50.dll
 - libicudata64.50.1.dll
 - libicudata64.50.dll
 - libicudata64.dll

- libicuuc.dll
- libicuuc50.1.dll
- libicuuc50.dll
- libicuuc64.50.1.dll
- libicuuc64.50.dll
- libicuuc64.dll

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 wie die Verbindung vom Client zum Host bereitstellt. 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.

Folgende Dateien von Developer for System z müssen APF-autorisierte Dateien sein:

- /usr/lpp/rdz/bin/
 - BWBTSOW
 - CRASTART
 - fekfomvs
 - fekfriwp

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 hierzu 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-Ladmodule 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-Ladmodul 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/
 - AZUTSTRN
 - BWBTSOW
 - BWBTRANT
 - 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-Eingabeaufforderungen):

```
$ 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 port1** (TSO oder z/OS UNIX) aufgelistet werden. Die erstellte Ausgabe entspricht in etwa dem folgenden Beispiel:

```

MVS TCP/IP NETSTAT CS VxRy      TCPIP Name: TCPIP      17:08:32
Port# Prot User   Flags   Range      IP Address
-----
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.

Adressraum, Größe

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.

Sonstige Informationen

Fehlerrückmeldung B37 - Abbruch aufgrund fehlenden Speicherplatzes

Wenn ein Benutzer Fehlerrückmeldungen während einer Kompilierungsaktion auswählt, werden von Developer for System z mehrere temporäre Dateien erstellt. Wenn für eine dieser Dateien der Speicherplatz zu gering wird, enden die Kompilierungsjobs mit einem Fehler "B37-04", das heißt einem Abbruch aufgrund fehlenden Speicherplatzes.

Wenn dieser Fehler bei Ihren Benutzern auftritt, passen Sie die Speicherplatzzuordnung in FEK.SFEKPROC(FEKFERRF) an. Der Standardwert ist SPACE(200,40) TRACKS.

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.

OutOfMemoryError

Ein RSE-Thread-Pool kann mit einer OutOfMemoryError-Fehlernachricht fehlschlagen, die protokolliert wird. Dieser Fehler bezieht sich auf die Größe des Java-Heapspeichers und kann auftreten, wenn die in diesem Thread-Pool aktiven Benutzer mehr Ressourcen verwenden als vorhergesehen. Folgendes sind häufige Ursachen dieses Fehlers:

- Erweiterung großer Dateifilter in Remote Systems Explorer
- Öffnen von PDS(E)-Dateien mit einer großen Anzahl an Mitgliedern
- Öffnen von großen Mitgliedern oder sequenziellen Dateien

Dieses Problem können Sie folgendermaßen lösen:

- Erhöhen Sie die Anweisung `-Xmx` in `rsed.envvars`, da diese die maximale Größe des Java-Heapspeichers steuert. Beachten Sie, dass der Java-Heapspeicher sich den Begrenzungen des Adressraums anpassen muss.
- Verringern Sie die Anweisung `-Dmaximum.clients` in `rsed.envvars`, da dadurch gesteuert wird, wie viele Benutzer in einem einzelnen Thread-Pool enthalten sein können (und somit einen einzelnen Java-Heapspeicher gemeinsam nutzen).

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 13. 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.

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.

- „Auswahl von SSL oder TLS als Verschlüsselungsverfahren“ auf Seite 196
- „Speicherpositionen für private Schlüssel und Zertifikate festlegen“ auf Seite 196
- „Schlüsseldatei mit RACF erstellen“ auf Seite 197
- „Vorhandene RSE-Konfiguration klonen“ auf Seite 199
- „Koexistenz durch Aktualisieren von rsed.envvars aktivieren“ auf Seite 199
- „Aktualisierung von ssl.properties durchführen, um SSL zu aktivieren“ auf Seite 200
- „Neuen RSE-Dämon erstellen, um SSL zu aktivieren“ auf Seite 200
- „Verbindung testen“ auf Seite 201
- „Unterstützung der X.509-Clientauthentifizierung hinzufügen (optional)“ auf Seite 204
- „Schlüsseldatenbank mit gskkyman erstellen (optional)“ auf Seite 204
- „Keystore mit keytool erstellen (optional)“ auf Seite 207

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 in den folgenden Abschnitten beschriebenen 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.

Auswahl von SSL oder TLS als Verschlüsselungsverfahren

Die Variable `DSTORE_SSL_ALGORITHM` in der Anweisung `_RSE_JAVA_OPTS` von `rsed.envvars` ermöglicht Ihnen, zwischen SSL und dem Nachfolgeprotokoll TLS (Transport Layer Security) als Verschlüsselungsmethode zu wählen. Dies wird im Abschnitt "Zusätzliche Java-Startparameter mit `_RSE_JAVA_OPTS` definieren" im Handbuch *Hostkonfiguration* (IBM Form SC12-4062) dokumentiert.

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 39. Mechanismen für den SSL-Zertifikatsspeicher

Zertifikatsspeicher	Erstellt und verwaltet von	RSE-Dämon	RSE-Server
Schlüsseldatei	SAF-kompatibles Sicherheitsprodukt	unterstützt	unterstützt
Schlüsseldatenbank	z/OS UNIX gskkyman	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. Bibliotheken 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-Dämonsschlüsseldatenbank 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 vorangehende 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))
```

Es kann auch vorkommen, dass das zur Signierung Ihres Zertifikats verwendete CA-Zertifikat von einem anderen, höheren CA-Zertifikat signiert worden ist. In diesem Fall muss dieses höhere CA-Zertifikat ebenfalls zur Schlüsseldatei hinzugefügt werden. Dies trifft auf sämtliche Ebenen der zur Signierung des Zertifikats verwendeten CA-Zertifikate bis zum Root-CA-Zertifikat zu, das auf jeden Fall ein selbst signiertes Zertifikat ist.

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 im Verzeichnis `/etc/rdz/` befinden. Dies ist die im Abschnitt "Angepasstes Setup" im Handbuch *Hostkonfiguration* (SC12-4062) verwendete Standardposition.

```
$ cd /etc/rdz  
$ mkdir ssl  
$ cp rsed.envvars ssl  
$ cp ssl.properties ssl  
$ ls ssl  
rsed.envvars    ssl.properties
```

Die im vorangehenden Beispiel 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
-> ändern: _RSE_RSED_PORT=4047
-> ändern: -Ddaemon.log=/var/rdz/logs/ssl
-> ändern: -Duser.log=/var/rdz/logs/ssl
-> am ENDE hinzufügen:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

Die im vorangegangenen Beispiel 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 im vorangegangenen Beispiel 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.

System SSL (vom Dämon verwendet) verwendet sofern verfügbar immer ICSF, die Schnittstelle zur Verschlüsselungshardware von System z. Für die gemeinsame Verwendung der Dämondefinitionen mit dem Server bei der Verwendung von ICSF muss der Server-Keystore-Typ JCECCARACFKS angegeben sein. Als Keystore für öffentliche Schlüssel wird hier auch eine SAF-konforme Schlüsseldatei verwendet, der private Schlüssel wird aber in ICSF gespeichert. Zur Steuerung der Verwendung von Verschlüsselungsschlüsseln und Verschlüsselungsservices verwendet ICSF, wie im Handbuch *Cryptographic Services ICSF Administrator's Guide* (IBM Form SA22-7521) beschrieben, Profile der Sicherheitsklassen CSFKEYS und CSFSERV.

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 sich die Beispiel-JCL in FEK.#CUST.PROCLIB(RSED) befindet. Dies ist die im Abschnitt "Angepasstes Setup" im Handbuch *Hostkonfiguration* (SC12-4062) 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 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 TMPDIR=,
//          PORT=,
//          IVP=,                * 'IVP' für einen IVP-Test
//          CNFG='/etc/rdz/ssl',
//          HOME='/usr/lpp/rdz'
//*
//RSED      EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
// PARM='PGM &HOME./bin/rsed.sh &IVP -C&CNFG -P&PORT -T&TMPDIR'
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//          PEND
//*
//RSED      EXEC RSED
//*
```

Abbildung 30. 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 PTKTDATA-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(RSEDSSL) 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 der Developer for System z-Client das Hostzertifikat oder die signierende CA nicht erkennt, fragt der Developer for System z-Client beim Benutzer an, ob dieses Zertifikat vertrauenswürdig ist.

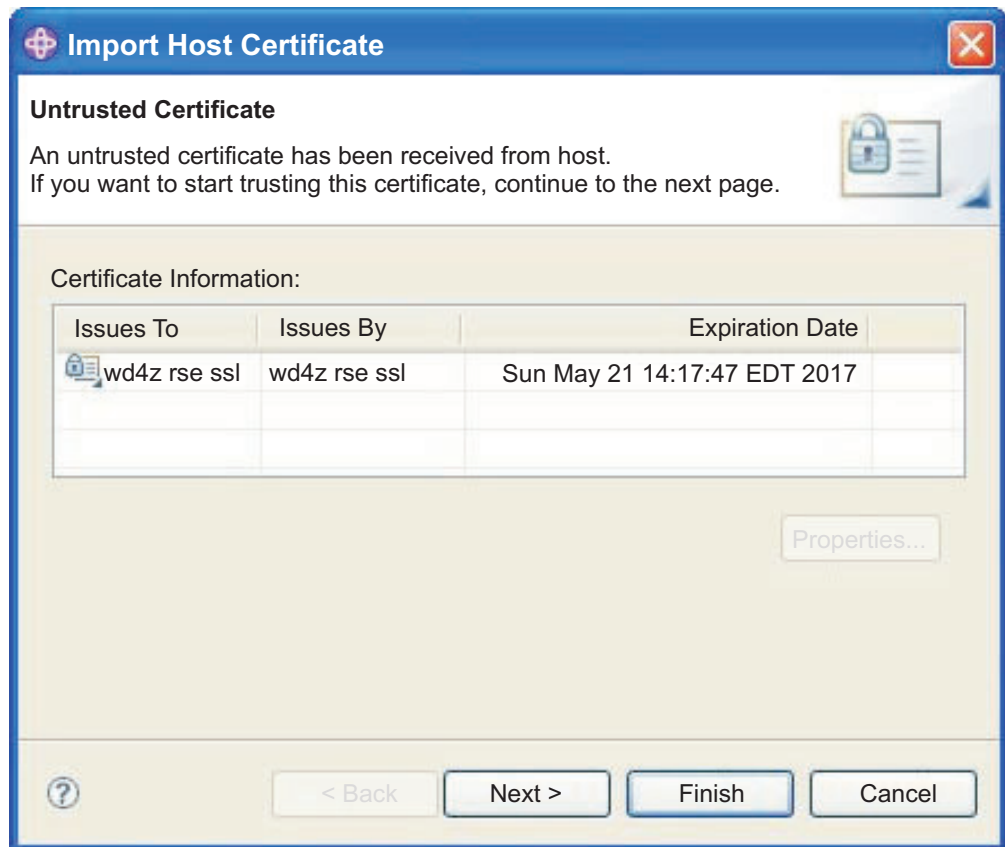


Abbildung 31. Dialog 'Hostzertifikat importieren'

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:

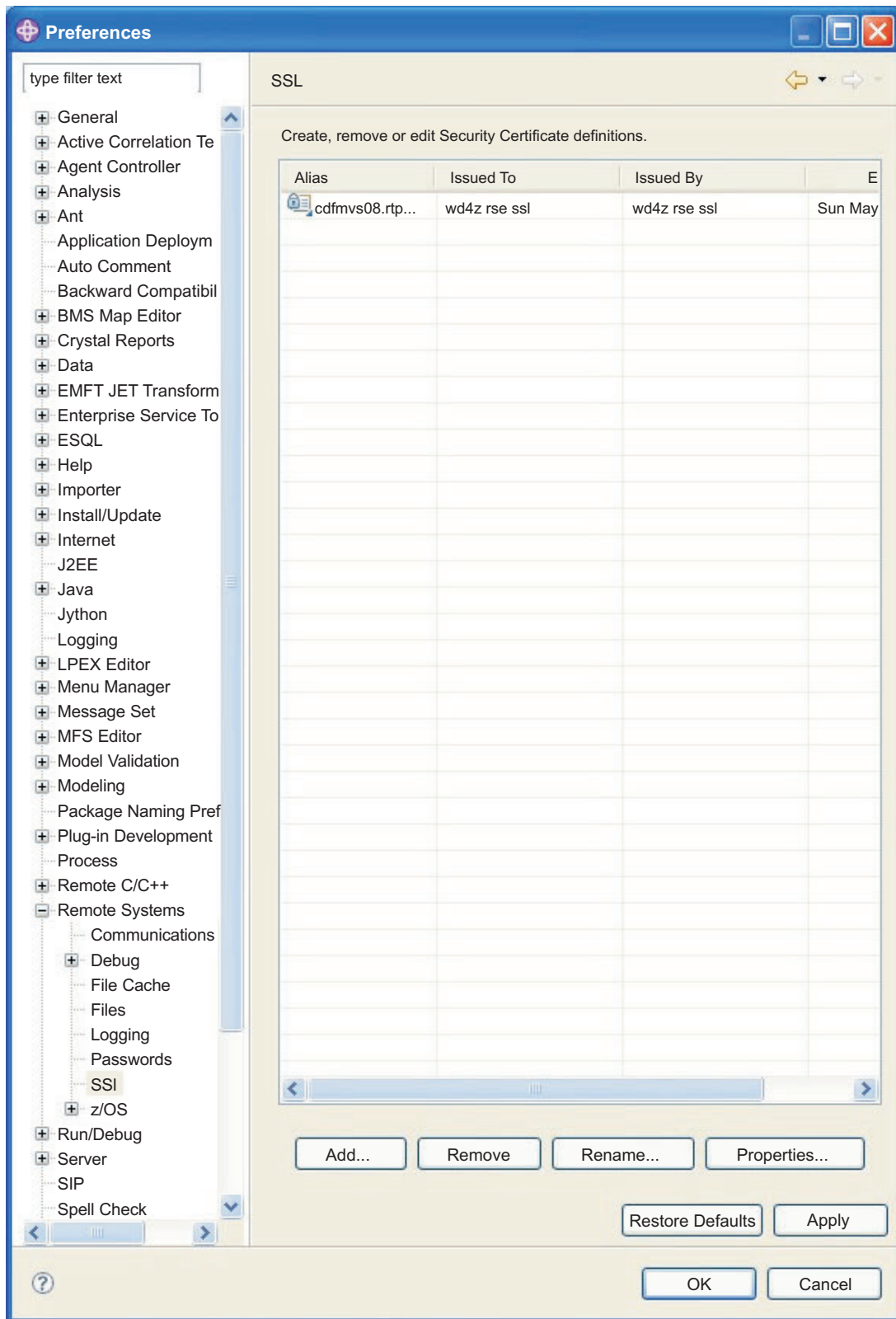


Abbildung 32. Vorgabendialog - SSL

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 177 und „RSE-Benutzer, Protokollierung“ auf Seite 178.

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 32. 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 Zertifikationsauthentifizierung 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.

```
SETOPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)  
oder  
SETOPTS 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 vorangegangene Beispiel beginnt mit der Erstellung der 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 im vorangegangenen Beispiel 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.

Kapitel 14. 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).

Hostnamen, Abhängigkeit

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 Stack 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 210.) 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 210.)

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 Domännennamensserver 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 Domännennamensserver zu verwenden sind. Außerdem kann mit der Anweisung LOOKUP TCPIP.DATA gesteuert werden, wie Domännennamensserver 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. Wert der Umgebungsvariable **X_SITE**
Der Wert der Umgebungsvariable ist der Name der Informationsdatei HOSTS.SITEINFO, die mit dem TSO-Befehl **MAKESITE** erstellt wurde.
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 TCP/IP 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 Resolverkonfigurationsdateien 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.

```
//TCP/IP    PROC  PARMS='CTRACE(CTIEZB00)',PROF=TCPPROF,DATA=TCPDATA
//*
//* TCP/IP NETWORK
//*
//TCP/IP    EXEC  PGM=EZBTCP/IP,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. SYS1.TCPPARMS(TCPDATA) können Sie 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 Sie, dass die DD-Anweisung SETUP nicht verwendet wird. Wie Sie aus dem Abschnitt „Wissenswertes zu Resolvern“ auf Seite 210 wissen, bedeutet dies, dass GLOBALTCPIPDATA 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 Sie davon ausgehen, dass die Umgebungsvariable RESOLVER_CONFIG nicht gesetzt ist, können Sie Tabelle 40 auf Seite 215 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 211 erwähnt, enthält die Basiskonfigurationsdatei TCPIP.DATA-Anweisungen. Wenn der Systemname CDFMVS08 lautet, sehen Sie, 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. Tabelle 40 auf Seite 215 können Sie außerdem entnehmen, dass in Ermangelung anderer Angaben standardmäßig die ASCII-EBCDIC-Umsetztabelle 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 vorangegangenen Beispiel ist sowohl cdfmvs08 als auch cdfmvs08.raleigh.ibm.com als gültiger Name für die IP-Adresse des z/OS-Systems definiert.

Wenn ein Domänen Namensserver (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 40. 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 40 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 "Installationsprüfung" in *Hostkonfiguration* (SC12-4062) 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>` bzw. `<Resolver-Daten>` repräsentieren hier den Namen Ihrer IP-Resolver-Datei:

```
RESOLVER_CONFIG=<Resolver-Datei>
```

oder

```
RESOLVER_CONFIG='<Resolver-Daten>'
```

Literaturübersicht

Referenzierte Veröffentlichungen

In diesem Dokument werden die folgenden Veröffentlichungen referenziert:

Tabelle 41. Referenzierte Veröffentlichungen

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
Program Directory for IBM Rational Developer for System z	GI11-8298	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Program Directory for IBM Rational Developer for System z Host Utilities	GI13-2864	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Voraussetzungen	SC23-7659	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Leitfaden für den Schnelleinstieg in die Hostkonfiguration	GI11-3191	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Hostkonfiguration	SC12-4062	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Hostkonfigurationsreferenz	SC12-4489	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Host Configuration Utility	SC14-7282	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Messages and Codes	SC14-7497	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z - Antworten auf gängige Fragen der Hostkonfiguration und -wartung	SC12-4724	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Common Access Repository Manager Developer's Guide	SC23-7660	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Voraussetzungen	SC23-7659	Developer for System z	http://www.ibm.com/software/rational/products/developer/systemz/library/index.html
Rational Developer for System z Leitfaden für den Schnelleinstieg in die Hostkonfiguration	GI11-3191	Developer for System z	http://www.ibm.com/software/rational/products/developer/systemz/library/index.html
SCLM Developer Toolkit Administrator's Guide	SC23-9801	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Using APPC to provide TSO command services	SC14-7291	White Paper	http://www-01.ibm.com/support/docview.wss?uid=swg27038517

Tabelle 41. Referenzierte Veröffentlichungen (Forts.)

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
Using ISPF Client Gateway to provide CARMA services	SC14-7292	White Paper	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Communications Server IP Configuration Guide	SC31-8775	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Configuration Reference	SC31-8776	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Diagnosis Guide	GC31-8782	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP System Administrator's Commands	SC31-8781	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA Network Implementation Guide	SC31-8777	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server SNA Operations	SC31-8779	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Cryptographic Services System SSL Programming	SC24-5901	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
DFSMS Macro Instructions for Data Sets	SC26-7408	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
DFSMS Using data sets	SC26-7410	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Language Environment Customization	SA22-7564	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Language Environment Debugging Guide	GA22-7560	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Diagnosis: Tools and Service Aids	GA22-7589	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL Reference	SA22-7597	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning: APPC/MVS Management	SA22-7599	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning Workload Management	SA22-7602	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS System Commands	SA22-7627	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E Customization	SA22-7783	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

Tabelle 41. Referenzierte Veröffentlichungen (Forts.)

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
TSO/E REXX Reference	SA22-7790	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Command Reference	SA22-7802	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Planning	GA22-7800	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services User's Guide	SA22-7801	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Using REXX and z/OS UNIX System Services	SA22-7806	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Java™ Diagnostic Guide	SC34-6650	Java 6.0	http://www.ibm.com/developerworks/java/jdk/diagnosis/
Java SDK and Runtime Environment User Guide	/	Java 6.0	http://www-03.ibm.com/servers/eserver/zseries/software/java/
Resource Definition Guide	SC34-6430	CICS TS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
Resource Definition Guide	SC34-6815	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
Resource Definition Guide	SC34-7181	CICSTS 4.2	https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
RACF Security Guide	SC34-6454	CICS TS 3.1	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
RACF Security Guide	SC34-6835	CICSTS 3.2	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html
RACF Security Guide	SC34-7003	CICSTS 4.1	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
RACF Security Guide	SC34-7179	CICSTS 4.2	https://publib.boulder.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html
Language Reference	SC27-1408	Enterprise COBOL für z/OS	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html

In diesem Dokument werden die folgenden Websites referenziert:

Tabelle 42. Referenzierte Websites

Beschreibung	Referenzwebsite
Information Center für Developer for System z	http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp
Developer for System z Library	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Homepage von Developer for System z	http://www-03.ibm.com/software/products/us/en/developerforsystemz/
Developer for System z Recommended service	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335

Tabelle 42. Referenzierte Websites (Forts.)

Beschreibung	Referenzwebsite
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
IBM Tivoli Directory Server	http://www-01.ibm.com/software/tivoli/products/directory-server/
Plug-ins für Tools zur Fehlerbestimmung	http://www-01.ibm.com/software/awdtools/deployment/pdtpplugins/
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 liefern, die bei der Konfiguration der als Voraussetzung erforderlichen Komponenten des Hostsystems auftauchen.

Tabelle 43. Veröffentlichungen mit weiteren Informationen

Titel der Veröffentlichung	IBM Form	Bezug	Referenzwebsite
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	http://www.redbooks.ibm.com/
System Programmer's Guide to: Workload Manager	SG24-6472	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	http://www.redbooks.ibm.com/
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	http://www.redbooks.ibm.com/
Tivoli Directory Server for z/OS	SG24-7849	Redbook	http://www.redbooks.ibm.com/

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 Java EE entsprechen. Java EE-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 Anwendungsserverseite 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, numerischer Richtungswechsel und symmetrischer Richtungswechsel

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 Java EE 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 Prozessstyps (und der zugehörigen Aktivitätsinstanzen) werden in derselben Repository-Datei gespeichert. Es können Datensätze für mehrere Prozessstypen 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

Bemerkungen zur Dokumentation für IBM Rational Developer for System z

© Copyright IBM Corporation 2009, 2013.

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

*IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France*

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die 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 werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen 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
5 Technology Park Drive
Westford, MA 01886
U.S.A.*

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 geschätzt. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

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 dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

Copyrightlizenz

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind und Programmier- und Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme in beliebiger Form kopieren, ändern und verteilen, ohne dass dafür Zahlungen an IBM anfallen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die 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.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. 2009, 2013.

Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der International Business Machines Corp. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite www.ibm.com/legal/copytrade.shtml.

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Windows ist eine Marke der Microsoft Corporation in den USA und/oder anderen Ländern.

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

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein.

Index

Sonderzeichen

.dstoreMemLogging 176
.dstoreTrace 176
_RSE_PORTRANGE 22
/var/rdz/pushtoclient/*install 140, 142

A

Abfrage einer Zertifikatswiderrufsliste (CRL)
 CRL-Umgebungsvariablen 33
 rse.envvars 33
Abhängigkeit vom Hostnamen 209
ACEE, verwaltet 41
ACK, verzögert 64
ADNJSPAU, Verwaltungsdienstprogramm 149
Adressraum, Größe 191
Adressräume, Begrenzung für die Größe 95
Aktionen für Beschränkungen der Jobausführung 28
Aktivierung 131
Aktivierung von Benutzerexits 159
Aktualisierungsberechtigungen für Benutzer ohne Systemadministratorrechte 17
Anfängliche LDAP-Gruppenkonfiguration 138
Anforderungen an die Start-JCL 191
Anpassung der TSO-Umgebung 165
Anpassung von Application Deployment Manager 145
Anpassung von ISPF.conf 166
Anwendungsentwicklung 120
Anwendungsschutz für RSE definieren 51
Anzahl der Adressräume 85
Anzahl der Prozesse 88
Anzahl der Threads 91
APF-Autorisierung 188
 FEK.SFEKAUTH 54
Application Deployment Manager (ADM) 4
Application Deployment Manager, CICS Resource Definition-Editor 145
Application Deployment Manager, CICS Resource Definition-Server 145
Application Deployment Manager, Sicherheit 147
Application Deployment Manager anpassen 145
AQEZPCM 21
Arbeitsbereichsbindung 133
ASCHPMxx
 MAX 109
Aspekte der Leistung 119
Aspekte der Sicherheit 19
ASSIZEMAX 48
audit.action, Benutzerexit 162
audit.log 177

Ausführung mehrerer Instanzen 169
Auslastungsverwaltung 121
Authentifizierung, Debug Manager 21
Authentifizierung durch JES Job Monitor 20
Authentifizierung durch RSE-Dämon 35
Authentifizierung durch Sicherheitssoftware 34
Authentifizierung konfigurieren, SSL und X.509 195
Authentifizierungsmethoden 20
Automatisierte Synchronisierung 171

B

Basiskonfigurationsdateien des Resolvers 211
Bedingte Aktionen für Jobs 26
Bedingter Zugriff auf Spooldateien 29
Befehle für Sicherheitsfunktion, nützliche
 ADDGROUP 17
 ALTUSER 17
 CONNECT 17
Befehlssicherheit definieren, JES 52
Begrenzung für die Größe der Adressräume 95
Begrenzung für die Größe des Java-Heapspeichers 95
Beispielanalyse der Speicherbelegung 97
Beispielkonfiguration 113
 Anzahl der Thread-Pools 114
 Grenzwerte definieren 115
 minimale Grenzwerte bestimmen 114
Benutzer, RSE-Protokollierung 178
Benutzer-ID, variable, Ausführung mithilfe von 160
Benutzer-ID und Kennwort 20
Benutzer-ID und Kennwort für einmaliges Anmelden 20
Benutzer ohne Systemadministratorrechte, Aktualisierungsberechtigungen 17
Benutzerexit, Hinweise xv, 159
Benutzerexit, Konsolennachrichten 160
Benutzerexitpunkte, verfügbare 162
Benutzerexitroutine schreiben 159
Benutzerexits, Merkmale 159
Berechtigungsbits, z/OS UNIX 186
Beschränkung der externen Kommunikation auf angegebene Ports 22
Beschränkungen der Ausführung, Aktionen für Jobs 28
BPXPRMxx 115
 INADDRANYCOUNT 107
 MAXASSIZE 48, 106, 191
 MAXFILEPROC 106
 MAXMMAPAREA 106
 MAXPROCSYS 105, 193
 MAXPROCUSER 105, 193
 MAXSOCKETS 107
 MAXTHREADS 105
 MAXTHREADTASKS 105

BPXPRMxx (Forts.)
 MAXUIDS 106, 193

C

Cachegrößenbegrenzung, Java Virtual Machines (JVMs) 123
Cachesicherheit, Java Virtual Machines (JVMs) 123
CARMA, Traceerstellung 185
CARMA-Protokollierung
 rsecomm.log 180
CARMA und TCP/IP-Ports 63
CEE.SCEELPA
 SYS1.PARMLIB(LPALSTxx) 120
CICS Resource Definition-Editor (CRD-Editor), Application Deployment Manager 145
CICS Resource Definition-Server (CRD-Server), Application Deployment Manager 145
CICS-Ressourcendefinitionen, Administrator 145
CICS-Ressourcendefinitionen, Entwickler 145
CICS-Ressourceninstallation protokollieren 147
CICS-Transaktionen 40
CICS-Transaktionsdebugging 157
CICSplex SM Business Application Services (BAS) 146
CICSTS-Aspekte 145
CICSTS-Sicherheit 40
CLASSPATH 171
Clientauthentifizierung unter Verwendung von X.509-Zertifikaten 32
Clientfunktionen ändern 36
Clientkonfigurationssteuerung 130
Clientversionssteuerung 131
COBOL
 ferne Prüfung 185
Codeabdeckung, Protokollierung 181
Codeüberprüfung, Protokollierung 181
Common Access Repository Manager, Protokollierung 180
CRD-Repository 40

D

Dateiprofile definieren 54
Dateisystem-Speicherbelegung, z/OS UNIX 101
Dateisystemattribut SETUID 186
Dateisysteme, zFS 119
Debug Manager-Authentifizierung 21
Debug-Sicherheit 39
Debugger, Integrated 10
Debugging, CICS-Transaktion 157
Definitionen, Sicherheit 44
Definitionen für den Resolver 215

Definitionen von verschiedenen Ressourcen 108

EXEC-Karte, Server-JCL 108
FEJJCENFG 108
SYS1.PARMLIB(ASCHPMxx) 109
SYS1.PARMLIB(IEASYsxx) 108
SYS1.PARMLIB(IVTPRMxx) 108

Developer for System z, gestartete Tasks definieren 47

Developer for System z, Komponentenübersicht
grafische Darstellung 4

Developer for System z, Wissenswertes 3

Dienstprogramme für Cacheverwaltung, Java Virtual Machines (JVMs) 124

Durchsatz der Sicherheitsprüfung verbessern 121

E

Einführung, Push-to-Client-Aspekte 127

Eingangsport überprüfen 23, 36

Einstellungen und Klassen für Sicherheit aktivieren 46

Empfangsbestätigung, verzögert 64

Emulator, Host-Connect 194

Entwicklung von Anwendungen 120

Exitpunkte, verfügbare 162

Externe Kommunikation 62

Externe Kommunikation auf angegebene Ports beschränken 22

F

fa.log 176

Fehler aufgrund abnormaler Speicherbedingungen 194

Fehlerrückmeldungen, Trace 185

FEJJCENFG 62, 115, 173

CONSOLE_NAME 28

MAX_THREADS 108

FEJJCENFG, JES Job Monitor 41

FEKAPPL 20

fekfivpc.log 177

fekfivpi, Protokollierung des IVP-Tests
fekfivpi.log 180

fekfivpi.log 177

fekfivpi.log, Protokollierung des IVP-Tests 180

fekfivps.log 177

fekfivps.log, Protokollierung des IVP-Tests 181

FEKLOGS, Protokoll- und Konfigurationsanalyse 175

FEKRACF, Sicherheitsdefinitionen 44

fekrivp 188

Feste Java-Heapgröße 121

ffs.log 176

ffsget.log 176

ffsput.log 176

Fremdanbieter und X.509-Zertifikat 20

Für den Resolver verfügbare lokale Definitionen 215

G

GATE, Überlastung (Thrashing) 40

Gemeinsame Klassennutzung, in Java Virtual Machines (JVMs) aktivieren 123

Gemeinsame Klassennutzung aktivieren, Java Virtual Machines (JVMs) 123

Gemeinsame Klassennutzung durch mehrere Java Virtual Machines (JVMs) 122

Gestartete Tasks, Definieren für Developer for System z

JMON, gestartete Tasks 47

RSED, gestartete Tasks 47

Grenzwerte des Systems 193

Größe des Adressraums 191

Größenschätzungen, Richtlinien 96

Gruppenauswahl, LDAP-basiert 135

Gruppenauswahl, SAF-basiert 140

Gruppenverkettungen 132

gskkyman, Schlüsseldatenbank erstellen 204

H

Heapspeicher, Begrenzung für die Größe, Java 95

Hinweise zu WLM xiv, 73

Host-Adresse, nicht aufgelöst durch TCP/IP-Resolver

lock.log 216

Host-Connect-Emulator 194

Hostbasierte Projekte 143

Hostnamen, Abhängigkeit 209

Hostnamen in Developer for System z anwenden 213

Hosttabellen, lokal 212

I

Identische Konfiguration in einem Sysplex 169

Identische Softwareversionen mit unterschiedlichen Konfigurationsdateien 170

IEASYSxx 116

MAXUSER 108, 193

Installation protokollieren, CICS-Ressourcen 147

Integrated Debugger 10

verschlüsselte Kommunikation 31

Interne Kommunikation 62

ISP.SISPLOAD

TSO/ISPF-Client-Gateway von ISPF 49

ISPF, mehrere Zuordnungs-Execs verwenden 167

ISPF.conf, Basisanpassung 166

ISPF.conf-Dateien mit mehreren Konfigurationen 168

IVP-Test, Protokollierung

fekfivpi.log 180

fekfivps.log 181

IVP-Test, Protokollierung von fekfivpc 180

IVTPRMxx

ECSA MAX 108

FIXED MAX 108

J

JAVA_DUMP_TDUMP_PATTERN 182

Java-Heapgröße, fest 121

Java-Option 'Xquickstart' 122

Java-Speicherauszüge 182

Java Virtual Machines (JVMs), gemeinsame Klassennutzung 122

JES-Befehlssicherheit definieren 52

JES JMON

GEN_CONSOLE_NAME 29

JES Job Monitor, Authentifizierung 20

JES Job Monitor, FEJJCENFG 41

JES Job Monitor (JMON) 4

JES Job Monitor, Protokollierung 177

JES Job Monitor, Traceerstellung 184

JES Job Monitor-Konfiguration

GEN_CONSOLE_NAME 29

JES-Sicherheit 26

JMON 52, 173

Jobs, bedingte Aktionen 26

JVMs, gemeinsame Klassennutzung 122

K

Karenzzeit, Zurückweisung von Änderungen 142

Kennwort für einmaliges Anmelden und Benutzer-ID 20

Kennwort und Benutzer-ID 20

Keystore mit keytool erstellen 207

keytool, Keystore erstellen 207

Klassifikation für Verarbeitungsprozesse, WLM 73

Klassifikationsregeln, WLM 74

Klonen der vorhandenen RSE-Konfiguration 199

Koexistenz, rsed.envvars zum Aktivieren der Koexistenz aktualisieren 199

Kommunikation, extern 62

Kommunikation, intern 62

Kommunikation, mit SSL/TLS verschlüsselt 30

Kommunikation, mit SSL verschlüsselt 149

Kommunikation mit SSL verschlüsseln 22

Kommunikation mit TLS verschlüsseln 22

Komponentenübersicht, Developer for System z

grafische Darstellung 4

Konfiguration, identisch in einem Sysplex 169

Konfigurationsdateien, Developer for System z 41

Konfigurationsdateien, unterschiedliche in identischen Softwareversionen 170

Konfigurationsdateien des Resolvers 211

Konfigurationsdaten, Suchreihenfolge 210

Konfigurationsprobleme lösen 175

Konfigurationsschritte 134

Konsolennachrichten, Benutzerexit 160

L

Language Environment, Laufzeitbibliotheken 120
Laufzeitbibliotheken, Language Environment 120
LDAP-Aspekte 63
LDAP-Gruppen, Entwickler hinzufügen 139
LDAP-Gruppenkonfiguration, anfängliche 138
LDAP-Schema 136
LDAP-Serverauswahl 136
LDAP-Serverposition 137
Leistungsaspekte 119
LIMIT_COMMANDS 27
LIMIT_VIEW 29
lock.log 176
logon.action, Benutzerexit 163
Lokale Hosttabellen 212
LPALSTxx 120

M

Mehrere Entwicklergruppen 131
Mehrere Instanzen ausführen 169
Mehrere ISPF.conf-Dateien 168
Mehrere Konfigurationen für Developer for System z durch Verwendung mehrerer ISPF.conf-Dateien 168
Mehrere Zuordnungs-Execs, TSO/ISPF 167
Metadaten, Position 129
Metadaten, Push-to-Client 129
Metadaten, Sicherheit 129
Methoden zur Authentifizierung 20
Migrationshinweise, Verwaltungsdienstprogramm 154
Mit SSL/TLS verschlüsselte Kommunikation 30
Mit SSL verschlüsselte Kommunikation 40, 149
Musterkonfiguration, LDAP-Gruppenauswahl 137
Musterkonfiguration, SAF-basierte Gruppenauswahl 141
MVS-Bibliotheken für RSE definieren 49
MVS-Speicherauszüge 181

N

Nachrichten des Verwaltungsdienstprogramms 154
netstat 190
Netz überwachen 113

O

OFF.REMOTECOPY.MVS 37
OMVS-Segment definieren 47
Optimierungsaspekte 83
OutOfMemoryError 194

P

PassTicket-Unterstützung für RSE definieren 50
PassTickets verwenden 23
Pipelinesicherheit 147
Plattenspeicherplatz, Java Virtual Machines (JVMs) 124
POE-Überprüfung 23, 36
Portauswahl beschränken 67
PORTRANGE 190
Portreservierung, TCP/IP 63
Ports, externe Kommunikation auf angegebene Ports beschränken 22
Ports, TCP/IP 61
Ports, TCP/IP und CARMA 63
Position der Gruppen-Metadaten 133
Primäre und nicht primäre Verbindungsregionen 146
Primäres System 128
Private Schlüssel und Zertifikate, Speicherpositionen festlegen 196
Profile für Datei definieren 54
Programmgesteuerte MVS-Bibliotheken für RSE definieren 49
Programmgesteuerte UNIX-Dateien für RSE definieren 58
Programmgesteuerte z/OS UNIX-Dateien für RSE definieren 58
Programmsteuerung autorisieren 187
Projekte, hostbasiert 143
Protokoll- und Konfigurationsanalyse mit FEKLOGS 175
Protokolldateien
 .dstoreMemLogging 176
 .dstoreTrace 176
 audit.log 176
 fa.log 176
 fekfivpi.log 176
 fekfivps.log 176
 ffs.log 176
 ffsget.log 176
 ffsput.log 176
 lock.log 176
 rmt_class_loader.cache.jar 176
 rsecomm.log 176
 rsedaemon.log 176
 rseserver.log 176
 serverlogs.count 176
 stderr.log 176
 stdout.log 176
Protokollierung, Codeabdeckung 181
Protokollierung, Codeüberprüfung 181
Protokollierung, SCLM Developer Toolkit 180
Protokollierung des IVP-Tests "fekfivpc" fekfivpc.log 180
Protokollierung des IVP-Tests fekfivpi 180
Protokollierung des RSE-Benutzers 178
Protokollierung des RSE-Dämons 177
Protokollierung des Thread-Pools 177
Protokollierung von CARMA 180
Protokollierung von JES Job Monitor 177
Prüfdaten
 protokollierte Aktionen 25

Prüfprotokollierung, vom RSE-Dämon verwaltet 24
Prüfprozesse
 modify switch 25
Prüfung der Zertifizierungsstelle
 gskkyman 33
 SAF-Schlüsseldatei 33
 TRUST, HIGHTRUST 33
Push-to-Client 38
Push-to-Client-Aspekte 127
Push-to-Client-Back-End, zu LDAP hinzufügen 138
Push-to-Client-Metadaten 129
pushtoclient.properties 139, 142

R

RACF
 Berechtigungen 55
RACF, Schlüsseldatei erstellen 197
Referenzierte Veröffentlichungen 217
Repositorysicherheit, CRD 147
Reservierte Ports, TCP/IP 190
Reservierte TCP/IP-Ports 190
Reservierung, TCP/IP-Port 63
Resolver, verfügbare lokale Definitionen 215
Resolver, Wissenswertes 210
Ressourcendefinitionen, verschiedene 108
Ressourceninstallation protokollieren, CICS 147
Ressourcennutzung, temporäre 94
Ressourcennutzung, Überblick 84
Ressourcennutzung optimieren 83
Ressourcensicherheit 149
RESTful-Schnittstelle 146
RESTful-Schnittstelle oder Web-Service-Schnittstelle 146
rmt_class_loader.cache.jar 176
RSE, Anwendungsschutz definieren 51
RSE, PassTicket-Unterstützung definieren 50
RSE, programmgesteuerte MVS-Bibliotheken definieren 49
RSE, programmgesteuerte z/OS UNIX-Dateien definieren 58
RSE, pushtoclient.properties 43
RSE, rsed.envvars
 _RSE_JAVAOPTS 42
RSE, ssl.properties 43
RSE, Traceerstellung 184
RSE, Überprüfung des Eingangsports definieren 36
RSE als Java-Anwendung
 grafische Darstellung 5
RSE als sicheren sicheren z/OS UNIX-Server definieren 49
RSE-Benutzer, Protokollierung
 .dstoreMemLogging 178
 .dstoreTrace 178
 ffs.log 178
 ffsget.log 178
 ffsput.log 178
 lock.log 178
 rmt_class_loader.cache.jar 178
 rsecomm.log 178

- RSE-Benutzer, Protokollierung (*Forts.*)
 - stderr.log 178
 - stdout.log 178
- RSE-Dämon 62
- RSE-Dämon, Authentifizierung 35
- RSE-Dämon, Protokolldateien
 - audit.log 178
 - rsedaemon.log 178
 - rseserver.log 178
 - serverlogs.count 178
 - stderr*.log 178
 - stdout*.log 178
- RSE-Dämon, Protokollierung 177
- RSE-Dämon (RSED) 4
- RSE-Dämon und Prüfprotokollierung 24
- RSE-Konfiguration klonen 199
- RSE-Server 62
- RSE-Server als sicheren z/OS UNIX-Server definieren 49
- RSE-Thread-Pool, Protokolldateien
 - audit.log 178
 - rsedaemon.log 178
 - rseserver.log 178
 - serverlogs.count 178
 - stderr*.log 178
 - stdout*.log 178
- RSE überwachen 109
- rsecomm.log 176
 - SCLM Developer Toolkit, Protokollierung 180
- rsecomm.properties 185
- rsed.envvars 103, 140, 142, 171
 - _CMDSESV_CONF_HOME 168
 - _RSE_JAVAOPTS 165, 182
 - _RSE_PORTRANGE 22
 - Dmaximum.clients 104
 - Dmaximum.threadpool.process 104
 - Dmaximum.threads 104
 - Dminimum.threadpool.process 104
 - DSTORE_LOG_DIRECTORY 180, 184
 - STEPLIB 31
 - Xms 104
 - Xmx 104
- rsed.envvars, zum Aktivieren der Koexistenz aktualisieren 199
- rsedaemon.log 176, 177
- rseserver.log 176, 177

S

- Schlüsseldatei mit RACF erstellen 197
- Schlüsseldatenbank mit gskkyman erstellen 204
- Schnellstart, Java-Option (-Xquickstart) 122
- SCLM Developer Toolkit 50
- SCLM Developer Toolkit, Protokollierung
 - rsecomm.log 180
- SCLM Developer Toolkit (SCLMDT) 4
- SCLM-Sicherheit 41
- Secure Sockets Layer, Verbindung in der Hostkonfiguration testen 201
- Secure Sockets Layer konfigurieren 195
- Secure Sockets Layer zur Verschlüsselung der Kommunikation 22
- Segment definieren, OMVS 47
- Serversauswahl, LDAP 136

- serverlogs.count 176
- Serverposition, LDAP 137
- SETUID, Dateisystemattribut 186
- Sicherer z/OS UNIX-Server, RSE definieren 49
- Sicherheit, CICSTS 40
- Sicherheit, Debug 39
- Sicherheit, JES 26
- Sicherheit, Pipeline 147
- Sicherheit, Ressourcen 149
- Sicherheit, SCLM 41
- Sicherheit, Transaktionen 147
- Sicherheit des CRD-Repositorys 147
- Sicherheit für JES-Befehle definieren 52
- Sicherheit für Verbindungen 21
- Sicherheit von Application Deployment Manager (ADM) 147
- Sicherheitsaspekte 19
- Sicherheitsdefinition 141
- Sicherheitsdefinitionen 44
- Sicherheitsdefinitionen, Prüfliste 44
- Sicherheitseinstellungen prüfen 59
- Sicherheitseinstellungen und -klassen aktivieren 46
- Sicherheitsprofil mit gespeicherten Begrenzungen 192
- Sicherheitsprüfung, Durchsatz verbessern 121
- Sicherheitssoftware, Authentifizierung 34
- Signiertes Zertifikat, selbstsigniert oder durch Zertifizierungsstelle signiert 198
- SMP/E-Installation, Sticky Bit 189
- Softwareversionen, identische mit unterschiedlichen Konfigurationsdateien 170
- Speicherauszüge, Java 182
- Speicherauszüge, MVS 181
- Speicherauszüge, Position in z/OS UNIX 183
- Speicherauszugsdateien 181
- Speicherbelegung 95
- Speicherbelegung, Beispielanalyse 97
- Speicherbelegung, Metadaten 130
- Speicherbelegung, z/OS UNIX-Dateisystem 101
- Speicherbelegung durch Metadaten 130
- Speicherpositionen für private Schlüssel und Zertifikate 196
- Sperrdämon 13
- Sperrdämon (LOCKD) 4
- Sperrdämonflow
 - grafische Darstellung 13
- Sperren aufheben
 - RSE, Befehl 'modify cancel' 14
- Spooldateien, bedingter Zugriff 29
- SSL, Verbindung in der Hostkonfiguration testen 201
- SSL, Verschlüsselung 196
- SSL konfigurieren 195
- ssl.properties, SSL durch Erstellung eines neuen RSE-Dämons aktivieren 200
- ssl.properties für SSL-Aktualisierung aktivieren 200
- SSL-Verschlüsselung der Kommunikation 22

- Standardverhalten von TCP/IP, überschreiben 64
- Start-JCL, Anforderungen 191
- stderr*.log 176
- stderr.log 176
- stdout*.log 176
- stdout.log 176
- STEPLIB, Verwendung vermeiden 119
- Steuerung der Prüffunktion
 - _RSE_HOST_CODEPAGE 24
 - audit*-Optionen 24
 - daemon.log 24
 - enable.audit.log 24
- Sticky Bit, Verfügbarkeit des MVS-Ladmoduls für z/OS UNIX 189
- Subsystemtypen
 - ASCH 74
 - CICS 74
 - JES 74
 - OMVS 74
 - STC 74
- Suchreihenfolge für Konfigurationsdaten 210
- Suchreihenfolge in der z/OS-UNIX-Umgebung 211
- Synchronisierung, automatisierte 171
- SYS1.PARMLIB(BPXPRMxx) 115
 - MAXASSIZE 48, 191
 - MAXPROCSYS 193
 - MAXPROCUSER 193
 - MAXUIDS 193
- SYS1.PARMLIB(BPXPRMxx), Java Virtual Machines (JVMs) 123
- SYS1.PARMLIB(BPXPRMxx) mit festgelegten Begrenzungen 191
- SYS1.PARMLIB(IEASYSxx) 116
 - MAXUSER 193
- Sysplex, identische Konfiguration 169
- Systembibliotheken, Zugriff verbessern 119
- Systemexits, erzwungene Begrenzungen 192
- Systemgrenzwerte 193

T

- Tabellen für Umsetzung 212
- Taskeigner 7
- TCP/IP, für den Resolver verfügbare lokale Definitionen 215
- TCP/IP in Developer for System z anwenden 213
- TCP/IP konfigurieren 209
- TCP/IP-Portreservierung 63
- TCP/IP-Ports 61
- TCP/IP-Ports, grafische Darstellung 61
- TCP/IP-Ports, reservierte 190
- TCP/IP-Resolver, Host-Adresse nicht aufgelöst
 - lock.log 216
- TCP/IP-Verhalten, Standard überschreiben 64
- Temporäre Ressourcennutzung 94
- Test, Protokollierung von fekfivpi 180
- Testen der SSL-Verbindung in der Hostkonfiguration 201
- Thread-Pool, Protokollierung 177

- Threadsicherheit im RSE-Server
 - PassTickets 23
- TLS, Verschlüsselung 196
- TLS, Verschlüsselung der Kommunikation mit 22
- Trace für Fehlerrückmeldungen 185
- Traceerstellung 184
- Traceerstellung für CARMA 185
- Traceerstellung für JES Job Monitor 184
- Traceerstellung für RSE 184
- Transaktionssicherheit 147
- Transaktionsspeicherausgang, Strukturvariablen 182
- TSO Commands Service 4, 165
- TSO/ISPF, mehrere Zuordnungs-Execs verwenden 167
- TSO/ISPF, vorhandene ISPF-Profilen verwenden 166
- TSO/ISPF, Zuordnungs-Exec verwenden 167
- TSO/ISPF-Anpassung, ISPF.conf 166
- TSO/ISPF-Client-Gateway als Zugriffsmethode verwenden 166
- TSO/ISPF-Client-Gateway von ISPF ISPSISLOAD 49
- TSO/ISPF mit mehreren Konfigurationen verwenden 168
- TSO-Umgebung anpassen 165
- TSO-Zugriffsmethoden 165

U

- Überprüfung des Eingangsports für RSE definieren 36
- Überschreiben, TCP/IP-Standardverhalten 64
- Überwachung, Netz 113
- Überwachung, RSE 109
- Überwachung, z/OS UNIX 110
- Überwachung, z/OS UNIX-Dateisysteme 113
- Umsetztabelle 212
- UNIX-Server, RSE definieren 49
- UNIX-Speicherausgang, Position 183
- UNIX-Umgebung, Suchreihenfolge 211
- Unterschiedliche Konfigurationsdateien in identischen Softwareversionen 170
- Unterstützung der Clientauthentifizierung hinzufügen, X.509 204
- Unterstützung für RSE, PassTicket definieren 50

V

- Variable Benutzer-ID, Ausführung mithilfe von 160
- Verbesserung des Durchsatzes von Sicherheitsprüfungen 121
- Verbesserung des Zugriffs auf Systembibliotheken 119
- Verbindung verweigert 193
- Verbindungsflow 8
 - grafische Darstellung 8
- Verbindungsregionen, primäre und nicht primäre 146
- Verbindungssicherheit 21

- Veröffentlichungen, referenzierte 217
- verschlüsselte Kommunikation
 - Integrated Debugger 31
- Verschlüsselte Kommunikation, mit SSL 40, 149
- Verschlüsselte Kommunikation, mit SSL/TLS 30
- Verschlüsselung, SSL oder TLS 196
- Verschlüsselung der Kommunikation mit SSL 22
- Verschlüsselung mit TLS, Kommunikation 22
- Verteilte dynamische VIPA
 - EZBEPOR 66
 - PORT 66
 - PORTRANGE 66
 - SERVERWLM 66
 - SYSPLXEXPORTS 66
 - VIPADISTRIBUTE 66
- Verwaltung der Auslastung 121
- Verwaltungsdienstprogramm, Migrationshinweise 154
- Verwaltungsdienstprogramm, Nachrichten 154
- Verwaltungsdienstprogramm für CICS-Administratoren
 - bereitgestellte Funktionen 149
- Verweigerte Verbindung 193
- Verwendung einer Zuordnungs-Exec 167
- Verwendung von PassTickets 23
- Verwendung von STEPLIB vermeiden 119
- Verwendung vorhandener ISPF-Profilen 166
- Verzeichnisstruktur, z/OS UNIX
 - grafische Darstellung 15
- Verzögertes ACK 64
- VIPA, verteilt dynamisch 66
- Vorhandene ISPF-Profilen verwenden 166

W

- Web-Service-Schnittstelle 146
- Webverwaltungsregion 146
- Wichtige Ressourcen, Definitionen 104
 - rsed.envvars 104
 - SYS1.PARMLIB(BPXPRMxx) 105
- Wissenswertes zu Developer for System z 3
- WLM-Klassifikationsregeln 74
- Workload Manager 73

X

- X.509, Hinzufügen der Clientauthentifizierungsunterstützung 204
- X.509-Authentifizierung konfigurieren 195
- X.509-Zertifikat 20
- X.509-Zertifikate, Clientauthentifizierung 32
- Xquickstart, Java-Option 122

Z

- z/OS UNIX, Positionen für Speicherausgänge 183
- z/OS UNIX-Befehle, nützliche
 - chgrp 18
 - chmod 18
 - chown 18
 - ls 18
- z/OS UNIX-Berechtigungsbits 186
- z/OS UNIX-Dateisystem, Speicherbelegung 101
- z/OS UNIX-Dateisysteme überwachen 113
- z/OS UNIX-REXX-Exec 161
- z/OS UNIX-Server, RSE definieren 49
- z/OS UNIX-Shell-Script 161
- z/OS UNIX überwachen 110
- z/OS-UNIX-Umgebung, Suchreihenfolge 211
- z/OS UNIX-Verzeichnisstruktur
 - grafische Darstellung 15
- Zertifikat, X.509 20
- Zertifikate, Clientauthentifizierung unter Verwendung von X.509 32
- Zertifikatswiderrufsliste (CRL) abfragen
 - CRL-Umgebungsvariablen 33
 - rsed.envvars 33
- zFS-Dateisysteme verwenden 119
- Ziele festlegen, WLM 75
- Ziele in WLM festlegen 75
- Zugriff auf Spooldateien, bedingt 29
- Zugriff auf Systembibliotheken verbessern 119
- Zugriffsmethode, TSO/ISPF-Client-Gateway verwenden 166
- Zugriffsmethoden, TSO 165
- Zuordnungs-Exec verwenden 167
- Zurückweisung von Änderungen, Karenzzeit 142

Antwort

IBM Rational Developer for System z
Version 9.0.1
Hostkonfigurationsreferenz

IBM Form SC12-4489-05

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 0180 3 313233) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: translation@de.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

IBM Deutschland GmbH
TSC Germany

71083 Herrenberg



SC12-4489-05

