

IBM Rational Developer for System z
버전 9.0.1



호스트 구성 참조 안내서

IBM Rational Developer for System z
버전 9.0.1



호스트 구성 참조 안내서

참고

이 정보를 사용하기 전에 반드시 249 페이지의 『IBM Rational Developer for System z의 문서 주의사항』에 있는 일반 정보를 읽으십시오.

제 6판(2013년 12월)

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM Rational Developer for System z 버전 9.0.1(프로그램 번호 5724-T07) 및 모든 후속 릴리스와 수정에 적용됩니다.

전화 또는 팩스로 책을 주문할 수 있습니다. IBM Software Manufacturing Solutions에서는 동부 표준시(EST) 오전 8:30과 오후 7:00 사이에 책 주문을 받습니다. 전화 번호는 (800) 879-2755입니다. 팩스 번호는 (800) 445-9269입니다. 팩스는 Attn: Publications, 3rd floor로 보내셔야 합니다.

한국 IBM 담당자 또는 해당 지역의 IBM 지방 사무소로 책을 주문할 수도 있습니다. 다음 주소에서는 책을 구비하고 있지 않습니다.

IBM은 고객의 의견을 소중하게 생각합니다. 다음 주소로 의견을 보내주십시오.

IBM Corporation
Attn: Information Development Department 53NA
Building 501 P.O. Box 12195
Research Triangle Park NC 27709-2195
USA

팩스로 의견을 보낼 수 있습니다. 1-800-227-5088(미국 및 캐나다)

IBM에 정보를 보내는 경우, IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

Note to U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© Copyright IBM Corporation 2000, 2013.

목차

그림.	vii
표.	ix
이 문서의 정보	xi
이 책의 사용자	xii
변경사항 요약	xii
문서 콘텐츠에 대한 설명.	xiv
Developer for System z 이해	xiv
보안 고려사항	xiv
TCP/IP 고려사항	xiv
WLM 고려사항.	xiv
튜닝 고려사항	xv
성능 고려사항	xv
클라이언트로 푸시 고려사항	xv
CICSTS 고려사항	xv
사용자 엑시트 고려사항	xv
TSO 환경 사용자 정의	xv
다중 인스턴스 실행.	xv
구성 문제점 해결	xvi
SSL 및 X.509 인증 설정	xvi
TCP/IP 설정.	xvi

제 1 부 IBM Rational Developer for System z 호스트 구성 참조 안내서 1

제 1 장 Developer for System z 이해	3
컴포넌트 개요	4
Java 애플리케이션으로서의 RSE.	6
태스크 소유자	7
연결 플로우.	9
I 통합 디버거	11
CARMA	12
CARMA 구성 파일	13
데이터 세트 잠금 소유자	14
잠금 해제	15
z/OS UNIX 디렉토리 구조	16
비시스템 관리자에 대한 업데이트 권한	18

제 2 장 보안 고려사항	21
인증 방법	22
사용자 ID와 비밀번호.	22
사용자 ID와 일회성 비밀번호	22

X.509 인증	22
JES 작업 모니터 인증.	23
디버그 관리자 인증.	23
연결 보안	23
지정된 포트로 외부 통신 제한	24
I SSL 또는 TLS를 사용한 통신 암호화	24
POE(Port Of Entry) 확인	25
PassTicket 사용.	25
감사 로깅	26
감사 제어	27
감사 처리	27
감사 데이터	28
JES 보안	29
작업에 대한 조치 - 대상 제한사항	29
작업에 대한 조치 - 실행 제한사항.	30
스플 파일 액세스	32
SSL/TLS 암호화된 통신	33
I 통합 디버거 암호화 통신	34
X.509 인증서를 사용한 클라이언트 인증	35
인증 기관(CA) 유효성 검증	36
(선택사항) 인증서 폐기 목록(CRL) 조회.	37
보안 소프트웨어를 사용한 인증.	37
RSE 디먼을 사용한 인증.	38
POE(Port Of Entry) 확인	39
클라이언트 기능 변경	40
OFF.REMOTECOPY.MVS.	41
클라이언트로 푸시 개발자 그룹.	41
I 디버그 보안	42
CICSTS 보안	43
CRD 저장소	43
CICS 트랜잭션	43
SSL 암호화된 통신.	43
기타 정보	43
GATE 트래싱	43
관리 ACEE	44
SCLM 보안	44
Developer for System z 구성 파일	44
JES 작업 모니터 - FEJCNFG	45
RSE - rsed.envvars	45
RSE - ssl.properties	46
RSE - pushtoclient.properties	47
보안 정의	48

요구사항 및 체크리스트	48	스레드 개수	97
보안 설정 및 클래스 활성화	50	임시 자원 사용량	100
Developer for System z 사용자에게 대한 OMVS		스토리지 사용량	101
세그먼트 정의	51	Java 힙 크기 한계	101
Developer for System z 시작 태스크 정의	51	주소 공간 크기 한계	101
RSE를 보안 z/OS UNIX 서버로 정의	53	크기 예측 가이드라인	102
RSE에 대한 MVS 프로그램 제어 라이브러리 정		샘플 스토리지 사용량 분석	103
의	54	z/OS UNIX 파일 시스템 공간 사용량	107
RSE에 대한 PassTicket 지원 정의	55	키 자원 정의	110
RSE에 대한 애플리케이션 보호 정의	56	/etc/rdz/rsed.envvars	110
JES 명령 보안 정의	56	SYS1.PARMLIB(BPXPRMxx)	111
데이터 세트 프로파일 정의	58	다양한 자원 정의	114
RSE에 대한 z/OS UNIX 프로그램 제어 파일 정		서버 JCL의 EXEC 카드	114
의	64	FEK.#CUST.PARMLIB(FEJCNFG)	114
보안 설정 확인	64	SYS1.PARMLIB(IEASYSxx)	115
제 3 장 TCP/IP 고려사항	67	SYS1.PARMLIB(IVTPRMxx)	115
TCP/IP 포트	67	SYS1.PARMLIB(ASCHPMxx)	115
외부 통신	68	모니터링	116
내부 통신	69	RSE 모니터링	116
TCP/IP 포트 예약	69	z/OS UNIX 모니터링	117
CARMA 및 TCP/IP 포트	69	네트워크 모니터링	120
LDAP 고려사항	70	z/OS UNIX 파일 시스템 모니터링	120
기본 TCP/IP 동작 대체	70	샘플 설정	120
ACK 지연	70	스레드 풀 개수	121
다중 스택(CINET)	71	최소 한계 결정	121
CARMA 및 스택 선호도	71	한계 정의	122
분산 동적 VIPA	72	모니터 자원 사용량	123
포트 선택 제한	74	제 6 장 성능 고려사항	127
샘플 설정	76	zFS 파일 시스템 사용	127
제 4 장 WLM 고려사항	79	STEPLIB 사용 방지	127
워크로드 분류	79	시스템 라이브러리에 대한 액세스 향상	128
분류 규칙	80	LE(Language Environment) 런타임 라이브러리	128
목표 설정	81	애플리케이션 개발	128
목표 선택 고려사항	82	보안 검사 성능 향상	129
STC	83	워크로드 관리	129
OMVS	84	고정 Java 힙 크기	130
JES	85	Java -Xquickstart 옵션	130
ASCH	86	JVM 간에 클래스 공유	130
CICS	86	클래스 공유 사용	131
제 5 장 튜닝 고려사항	89	캐시 크기 한계	131
자원 사용량	89	캐시 보안	131
개요	90	SYS1.PARMLIB(BPXPRMxx)	132
주소 공간 개수	91	디스크 공간	132
프로세스 개수	94	캐시 관리 유틸리티	132

제 7 장 클라이언트로 푸시 고려사항	135
소개	135
기본 시스템	136
클라이언트로 푸시 메타데이터	137
메타데이터 위치	137
메타데이터 보안	138
메타데이터 공간 사용	138
클라이언트 구성 제어	139
클라이언트 버전 제어	139
복수 개발자 그룹	140
활성화	140
그룹 연결	141
작업공간 바인딩	141
그룹 메타데이터 위치	142
설정 단계	143
LDA 기반 그룹 선택	144
LDAP 스키마	144
LDAP 서버 선택	145
LDAP 서버 위치	146
샘플 설정	146
SAF 기반 그룹 선택	150
샘플 설정	151
변경사항 거부 유예 기간	152
호스트 기반 프로젝트	153
제 8 장 CICSTS 고려사항	155
RESTful 대 웹 서비스	156
기본 대 비1차 연결 리전	156
CICS 자원 설치 로깅	157
애플리케이션 배치 관리자 보안	157
CRD 저장소 보안	157
파이프라인 보안	157
트랜잭션 보안	157
SSL 암호화된 통신	159
자원 보안	159
관리 유틸리티	159
관리 유틸리티 마이그레이션 참고사항	164
관리 유틸리티 메시지	164
I CICS 트랜잭션 디버깅	167
제 9 장 사용자 엑시트 고려사항	169
사용자 엑시트 특성	169
사용자 엑시트 활성화	169
사용자 엑시트 루틴 기록	169
콘솔 메시지	170
가변 사용자 ID를 사용하여 실행	170
사용 가능한 종료점	172

audit.action	172
logon.action	173
제 10 장 TSO 환경 사용자 정의	175
TSO 명령 서비스	175
액세스 방법	175
TSO/ISPF Client Gateway 액세스 방법 사용	176
ISPF.conf	176
기존 ISPF 프로파일 사용	177
할당 exec 사용	177
여러 개의 할당 exec 사용	178
Developer for System z 설정이 여러 개인 다 중 ISPF.conf 파일	178
제 11 장 다중 인스턴스 실행	181
sysplex에서 동일 설정	182
동일한 소프트웨어 레벨, 다른 구성 파일	182
자동화된 동기화	183
기타 모든 상황	184
제 12 장 구성 문제점 해결	189
FEKLOGS를 사용한 로그 및 설정 분석	190
로그 파일	190
JES 작업 모니터 로깅	192
RSE 디먼 및 스레드 풀 로깅	192
RSE 사용자 로깅	193
SCLM 개발자 툴킷 로깅	194
CARMA 로깅	195
fekfivpc IVP 테스트 로깅	195
fekfivpi IVP 테스트 로깅	196
fekfivps IVP 테스트 로깅	196
코드 검토 로깅	196
코드 적용 로깅	196
덤프 파일	197
MVS 덤프	197
Java 덤프	197
z/OS UNIX 덤프 위치	199
추적	199
JES 작업 모니터 추적	199
RSE 추적	199
CARMA 추적	200
오류 피드백 추적	201
z/OS UNIX 권한 비트	202
SETUID 파일 시스템 속성	202
프로그램 제어 권한	203
APF 권한 부여	204
스타키(Sticky) 비트	205

예약된 TCP/IP 포트	206
주소 공간 크기	207
시작 JCL 요구사항	208
SYS1.PARMLIB(BPXPRMxx)에 설정된 제한 사항	208
보안 프로파일에 저장된 제한사항	208
시스템 종료에 의해 강제 실행된 제한사항	208
64비트 주소 지정에 대한 제한사항	208
기타 정보	209
오류 피드백 B37 공간 이상 종료	209
시스템 한계	209
연결이 거부됨	209
OutOfMemoryError	210
호스트 연결 애플레이터	210

제 13 장 SSL 및 X.509 인증 설정	211
SSL 또는 TLS를 암호화 방법으로 사용하도록 결 정	212
개인 키 및 인증서 저장 위치 결정	212
RACF를 사용하여 키 링 작성	213
(선택사항) 서명 인증서 사용	214
기존 RSE 설정 복제	215
rsed.envvars를 업데이트하여 공존 사용	216
ssl.properties를 업데이트하여 SSL 사용	216
새 RSE 디먼을 작성하여 SSL 활성화	217
연결 테스트	218
(선택사항) X.509 클라이언트 인증 지원 추가	221
(선택사항) gskkyman을 사용하여 키 데이터베이스 작성	222

(선택사항) keytool을 사용하여 키 저장소 작성	225
---	-----

제 14 장 TCP/IP 설정	227
호스트 이름 종속성	227
분석기 이해	228
구성 정보 검색 순서 이해	228
z/OS UNIX 환경에서 사용하는 검색 순서	229
기본 분석기 구성 파일	229
변환 테이블	230
로컬 호스트 테이블	230
Developer for System z에 이 설정 정보 적용	231
호스트 주소가 올바르게 분석되지 않음	234

제 2 부 부록	237
---------------------------	-----

참고 문헌	239
참조된 서적	239
정보 서적	242

용어집	243
---------------	-----

IBM Rational Developer for System z의 문서	
주의사항	249
저작권 라이선스	252
상표 정보	252
색인	253

그림

1. 컴포넌트 개요	4	17. STCRSE의 프로세스 수	96
2. Java 애플리케이션으로서의 RSE	6	18. 클라이언트당 프로세스 수	96
3. 태스크 소유자	8	19. 최대 RSE 스레드 풀 스레드 수	99
4. 연결 플로우	9	20. 최대 JES 작업 모니터 스레드 수	99
I 5. 통합 디버거	11	21. 로그인 수가 5인 경우 자원 사용량	104
6. CARMA 플로우	12	22. 로그인 수가 5인 경우 자원 사용량(계속)	105
7. 데이터 세트 인큐 판별 플로우	14	23. PDS 멤버 편집 시 자원 사용량	106
8. z/OS UNIX 디렉토리 구조	16	24. z/OS UNIX 파일 시스템 공간 사용량	108
I 9. 디버그 관리자용 AT-TLS 정책	35	25. 샘플 설정의 자원 사용량	124
10. TCP/IP 포트	67	26. 샘플 LDAP 스키마 정의	145
11. update.sh - 방화벽이 있는 DDVIPA 설정 지 원	75	27. ADNJSPAU - CICSTS 관리 유틸리티	161
12. 분산 동적 VIPA 샘플	76	28. ADNJSPAU - CICSTS 관리 유틸리티(2/3)	162
13. WLM 분류	79	29. ADNJSPAU - CICSTS 관리 유틸리티(3/3)	163
14. 최대 주소 공간 수	92	30. RSEDSSL - SSL에 대한 RSE 디먼 사용자 작업	217
15. 클라이언트당 주소 공간 수	93	31. 호스트 인증서 가져오기 대화 상자	218
16. 최대 프로세스 수	95	32. 환경 설정 대화 상자 - SSL	220

표

1. JES 작업 모니터 콘솔 명령	29	24. 주소 공간 계수	91
2. LIMIT_COMMANDS 명령 권한 매트릭스	29	25. 주소 공간 한계	94
3. 확장 JESSPOOL 프로파일	30	26. 프로세스 개수	94
4. LIMIT_CONSOLE 콘솔 권한 매트릭스	31	27. 프로세스 한계	97
5. LIMIT_VIEW 찾아보기 권한 매트릭스	32	28. 스레드 개수	97
6. SSL 인증서 스토리지 메커니즘	33	29. 스레드 한계	100
7. 클라이언트 기능 변경에 대한 SAF 정보	40	30. 스토리지 사용량에 대한 참조 설정	103
8. 클라이언트로 푸시 SAF 정보	41	31. 로그 출력 지시문	109
9. 디버그 기능에 대한 SAF 정보	42	32. 임시 출력 지시문	110
10. 보안 설정 변수	48	33. *.enabled에 대한 클라이언트로 푸시 그룹 지 원 매트릭스	140
11. JES2 작업 모니터 운영자 명령	57	34. reject.*.updates에 대한 클라이언트로 푸시 그 룹 지원 매트릭스	140
12. JES3 작업 모니터 운영자 명령	58	35. 클라이언트로 푸시 그룹 연결	141
13. WLM 시작점 서브시스템	80	36. 클라이언트로 푸시 LDAP 정보	144
14. WLM 작업 규정자	81	37. 클라이언트로 푸시 SAF 정보	150
15. WLM 워크로드	82	38. JAVA_DUMP_TDUMP_PATTERN 변수	198
16. WLM 워크로드 - STC	83	39. SSL 인증서 스토리지 메커니즘	212
17. WLM 워크로드 - OMVS	84	40. 분석기에 사용 가능한 로컬 정의	233
18. WLM 워크로드 - JES.	85	41. 참조된 서적	239
19. WLM 워크로드 - ASCH.	86	42. 참조된 웹 사이트	241
20. WLM 워크로드 - CICS	87	43. 정보 서적.	242
21. 일반 자원 사용법	90		
22. 사용자별 필수 자원 사용량	90		
23. 사용자별 자원 사용량	91		

이 문서의 정보

이 문서는 IBM® Rational® Developer for System z® 자체 및 기타 z/OS® 컴포넌트와 제품(예: WLM, CICS®)의 다양한 구성 태스크에 대한 배경 정보를 제공합니다.

이 매뉴얼에서 사용되는 이름은 다음과 같습니다.

- *IBM Rational Developer for System z*는 *Developer for System z*라고 합니다.
- *IBM Rational Developer for System z Integrated Debugger*를 통합 디버거라고 합니다.
- *Common Access Repository Manager*의 약어는 *CARMA*입니다.
- *Software Configuration and Library Manager Developer Toolkit*은 *SCLM Developer Toolkit*이라고 하며 약어는 *SCLMDT*입니다.
- *z/OS UNIX System Services*는 *z/OS UNIX*라고 합니다.
- *Customer Information Control System Transaction Server*는 *CICSTS*라고 하며 약어는 *CICS*입니다.

이 문서는 Developer for System z 호스트 구성을 설명하는 문서 세트의 일부입니다. 이러한 각 문서는 특정 사용자를 대상으로 합니다. 모든 문서를 읽어야 Developer for System z 구성을 완료할 수 있는 것은 아닙니다.

- *Rational Developer for System z Host Configuration Guide*(SC23-7658)에서는 모든 계획 태스크, 구성 태스크 및 옵션(선택적 옵션 포함)에 대해 자세히 설명하며 대체 시나리오를 제공합니다.
- *Rational Developer for System z 호스트 구성 참조서*(SC14-7290)에서는 Developer for System z 디자인에 대해 설명하고 Developer for System z, z/OS 컴포넌트 및 Developer for System z에 관련된 다른 제품(예: WLM 및 CICS)의 다양한 구성 태스크에 대한 배경 정보를 제공합니다.
- *Rational Developer for System z 호스트 구성 빠른 시작 안내서*(GA30-4183)에서는 Developer for System z의 최소 설정에 대해 설명합니다.
- *Rational Developer for System z Host Configuration Utility*(SC14-7282)에서는 Developer for System z의 기본 및 공통 선택적 사용자 정의 단계를 안내하는 ISPF 패널 애플리케이션인 호스트 구성 유틸리티에 대해 설명합니다.

이 문서의 정보는 모든 IBM Rational Developer for System z 버전 9.0 패키지에 적용됩니다.

이 책의 사용자

이 책은 IBM Rational Developer for System z 버전 9.0.1을 구성하고 튜닝하는 시스템 프로그래머를 위한 것입니다.

다른 책에 실제 구성 단계가 설명되어 있긴 하지만 이 책에서는 튜닝, 보안 설정 등과 같은 다양한 관련 주제를 자세히 설명합니다. 이 책을 사용하려면 z/OS UNIX System Services 및 MVS™ 호스트 시스템에 대해 잘 알아야 합니다.

변경사항 요약

이 절에서는 *IBM Rational Developer for System z 버전 9.0 호스트 구성 참조서*, SA30-4501-05 (2013년 12월 업데이트) 변경사항을 요약합니다.

텍스트 및 그림에 대한 기술적 변경사항이나 추가사항은 변경사항 왼쪽에 세로선으로 표시됩니다.

새 정보:

- 시간소인 로그 파일 이름에 대한 정보가 추가되었습니다. 190 페이지의 『로그 파일』을 참조하십시오.
- 감사할 수 있는 새 이벤트에 대한 정보가 추가되었습니다. 감사 데이터를 참조하십시오.

이 책에는 IBM Rational Developer for System z Version 9.0 Host Configuration Reference, SC14-7290-04에 이전에 제공되었던 정보가 포함되어 있습니다.

새 정보:

- TCP/IP 포트 사용이 업데이트되었습니다. 67 페이지의 『TCP/IP 포트』을 참조하십시오.
- 2개의 RSE 디먼을 자동으로 동기화하는 샘플이 추가되었습니다. 183 페이지의 『자동화된 동기화』을 참조하십시오.
- 새 로그 파일에 대한 정보가 추가되었습니다. 190 페이지의 『로그 파일』을 참조하십시오.

이 책에서는 *IBM Rational Developer for System z 버전 8.5.1 호스트 구성 참조서*, SA30-4501-03에 이전에 제공되었던 정보가 포함되어 있습니다.

새 정보:

- 클라이언트 기능 변경을 위해 SAF 프로파일에 대한 정보가 추가되었습니다. 40 페이지의 『클라이언트 기능 변경』을 참조하십시오.
- 업데이트된 자원 사용 수입니다. 89 페이지의 제 5 장 『튜닝 고려사항』의 내용을 참조하십시오.

- 스프레드 폴당 최대 사용자 수 기본값이 업데이트되었습니다. 89 페이지의 제 5 장 『튜닝 고려사항』을 참조하십시오.

이 책에는 *IBM Rational Developer for System z Version 8.5 Host Configuration Reference*, SC14-7290-02에 이전에 제공되었던 정보가 포함되어 있습니다.

새 정보:

- JES 작업 모니터 보안 정보가 업데이트되었습니다. 21 페이지의 제 2 장 『보안 고려사항』을 참조하십시오.
- 사용자 엑시트에 대한 정보가 추가되었습니다. 169 페이지의 제 9 장 『사용자 엑시트 고려사항』을 참조하십시오.

이 책에는 *IBM Rational Developer for System z Version 8.0.3 Host Configuration Reference*, SC14-7290-01에 이전에 제공되었던 정보가 포함되어 있습니다.

새 정보:

- z/OS UNIX 디렉토리 구조가 업데이트되었습니다. 16 페이지의 『z/OS UNIX 디렉토리 구조』를 참조하십시오.
- 호스트 기반 클라이언트 제어에 대한 정보가 추가되었습니다. 135 페이지의 제 7 장 『클라이언트로 푸시 고려사항』을 참조하십시오.
- 보안 관련 클라이언트로 푸시 정보가 추가되었습니다. 41 페이지의 『클라이언트로 푸시 개발자 그룹』을 참조하십시오.
- 관리 ACEE 사용 설명. 44 페이지의 『관리 ACEE』를 참조하십시오.
- 자동화된 감사 로그 처리에 대한 정보가 추가되었습니다. 27 페이지의 『감사 처리』를 참조하십시오.
- 구성 파일의 보안 및 감사 관련 지시문에 대한 정보가 업데이트되었습니다. 44 페이지의 『Developer for System z 구성 파일』을 참조하십시오.
- 추가 TCP/IP 정보가 추가되었습니다. 67 페이지의 제 3 장 『TCP/IP 고려사항』을 참조하십시오.
- SSL 통신에 대한 인증 기관 정보가 업데이트되었습니다. 211 페이지의 제 13 장 『SSL 및 X.509 인증 설정』을 참조하십시오.
- 자원 사용이 업데이트되었습니다. 89 페이지의 『자원 사용량』을 참조하십시오.

이 책에는 *IBM Rational Developer for System z Version 8.0.1 Host Configuration Reference*, SC14-7290-00에 이전에 제공되었던 정보가 포함되어 있습니다.

새 정보:

- "Developer for System z 이해"의 CARMA 절. 12 페이지의 『CARMA』를 참조하십시오.
- 일반 TCP/IP 관련 정보. 67 페이지의 제 3 장 『TCP/IP 고려사항』을 참조하십시오.

- B37 공간 이상 종료 해결. 209 페이지의 『오류 피드백 B37 공간 이상 종료』를 참조하십시오.

제거된 정보:

- *IBM Rational Developer for System z Version 7.6.1 Host Configuration Guide(SC23-7658-04)*에 이전에 제공되었던 정보가 지금은 두 개의 문서(*IBM Rational Developer for System z Host Configuration Guide(SC23-7658)*와 *IBM Rational Developer for System z Host Configuration Reference(SC14-7290)*)로 나뉘었습니다.
- APPC 설정 관련 정보가 *Using APPC to provide TSO command services (SC14-7291)* 백서로 이동되었습니다.
- INETD 설정

문서 콘텐츠에 대한 설명

이 절에서는 이 책에 제공된 정보를 요약합니다.

Developer for System z 이해

Developer for System z 호스트는 호스트 서비스 및 데이터에 대한 클라이언트 액세스를 제공하기 위해 상호작용하는 몇 개의 컴포넌트로 구성됩니다. 이러한 구성요소의 디자인을 이해하면 올바른 구성 결정을 내리는 데 도움이 됩니다.

보안 고려사항

Developer for System z는 비메인프레임 워크스테이션 사용자에게 메인프레임 액세스를 제공합니다. 따라서 연결 요청 유효성 검증, 호스트와 워크스테이션 간의 보안 통신 제공, 권한 부여 및 감사 활동은 제품 구성의 중요한 측면입니다.

TCP/IP 고려사항

Developer for System z는 TCP/IP를 사용하여 비메인프레임 워크스테이션 사용자에게 메인프레임 액세스를 제공합니다. 또한 다양한 컴포넌트와 기타 제품 간의 통신에도 TCP/IP를 사용합니다.

WLM 고려사항

전통적인 z/OS 애플리케이션과 달리 Developer for System z는 워크로드 관리자(WLM)가 쉽게 식별할 수 있는 단일 애플리케이션이 아닙니다. Developer for System z는 상호 작용을 통해 호스트 서비스와 데이터에 대한 클라이언트 액세스 권한을 제공하는 여러 컴포넌트로 구성됩니다. 이러한 서비스 중 일부는 서로 다른 주소 공간에서 활성화되므로 여러 WLM 분류가 발생합니다.

튜닝 고려사항

RSE(Remote Systems Explorer)는 Developer for System z의 코어입니다. 클라이언트로부터의 연결과 워크로드를 관리하기 위해 RSE는 스레드 풀링 주소 공간을 제어하는 다면 주소 공간으로 구성됩니다. 다면은 연결 및 관리를 위한 포컬 포인트의 역할을 하는 반면, 스레드 풀은 클라이언트 워크로드를 처리합니다.

따라서 RSE는 Developer for System z 설정 조정의 주요 대상이 됩니다. 그러나 각각 17개 이상의 스레드, 일정 양의 스토리지, 하나 이상의 주소 공간을 사용하는 수백 명의 사용자를 유지보수하려면 Developer for System z와 z/OS를 둘 다 올바르게 구성해야 합니다.

성능 고려사항

z/OS는 사용자 정의가 매우 용이한 운영 체제이므로 시스템 변경(때때로 사소한 변경 포함) 시 전체 성능에 막대한 영향을 줄 수 있습니다. 이 장에서는 Developer for System z 성능 향상을 위해 수행할 수 있는 몇 가지 변경사항을 강조합니다.

클라이언트로 푸시 고려사항

클라이언트로 푸시 또는 호스트 기반 클라이언트 제어는 다음 항목에 대한 중앙 관리를 지원합니다.

- 클라이언트 구성 파일
- 클라이언트 제품 버전
- 프로젝트 정의

CICSTS 고려사항

이 장에는 CICS Transaction Server 관리자에게 유용한 정보가 들어 있습니다.

사용자 엑시트 고려사항

이 장에서는 종료 루틴을 기록하여 Developer for System z의 기능을 개선하는 데 필요한 정보를 제공합니다.

TSO 환경 사용자 정의

이 장에서는 Developer for System z의 TSO 환경에 DD 문과 데이터 세트를 추가하여 TSO 로그인 프로시저를 모방하는 데 필요한 정보를 제공합니다.

다중 인스턴스 실행

예를 들어, 업그레이드를 테스트하는 경우와 같이 동일한 시스템에서 여러 Developer for System z 인스턴스를 활성화하려는 경우가 있습니다. 그러나 TCP/IP 포트와 같은 일부 자원은 공유할 수 없으므로 항상 기본값을 적용할 수 있는 것은 아닙니다. 이 장

에 있는 정보를 사용하여 서로 다른 Developer for System z 인스턴스의 공존을 계획한 후 이 구성 안내서를 사용하여 사용자 정의하십시오.

구성 문제점 해결

이 장은 Developer for System z 구성 중에 발생할 수 있는 몇 가지 일반적인 문제점을 해결하기 위해 제공되며 다음 절로 구성됩니다.

- FEKLOGS를 사용한 로그 및 설정 분석
- 로그 파일
- 덤프 파일
- 추적
- z/OS UNIX 권한 비트
- 예약된 TCP/IP 포트
- 주소 공간 크기
- APPC 트랜잭션 및 TSO 명령 서비스
- 기타 정보

SSL 및 X.509 인증 설정

이 부록은 SSL(Secure Socket Layer)을 설정할 때 또는 기존 설정을 확인하거나 수정할 때 발생할 수 있는 일반적인 문제점 해결에 유용한 정보를 제공합니다. 이 부록은 사용자가 X.509 인증서를 사용하여 스스로를 인증하는 것을 지원할 수 있도록 샘플 설정도 제공합니다.

TCP/IP 설정

이 부록은 TCP/IP를 설정할 때 또는 기존 설정을 확인하거나 수정할 때 발생할 수 있는 일반적인 문제점 해결에 유용한 정보를 제공합니다.

제 1 부 IBM Rational Developer for System z 호스트 구성 참조 안내서

제 1 장 Developer for System z 이해

Developer for System z 호스트는 클라이언트에 호스트 서비스 및 데이터에 대한 액세스를 제공하기 위해 상호작용하는 몇 개의 구성요소로 이루어져 있습니다. 이러한 구성요소의 디자인을 이해하면 올바른 구성 결정을 내리는 데 도움이 됩니다.

이 장에서 다루는 주제는 다음과 같습니다.

- 4 페이지의 『컴포넌트 개요』
- 6 페이지의 『Java 애플리케이션으로서의 RSE』
- 7 페이지의 『태스크 소유자』
- 9 페이지의 『연결 플로우』
- 11 페이지의 『통합 디버거』
- 12 페이지의 『CARMA』
- 14 페이지의 『데이터 세트 잠금 소유자』
- 16 페이지의 『z/OS UNIX 디렉토리 구조』

컴포넌트 개요

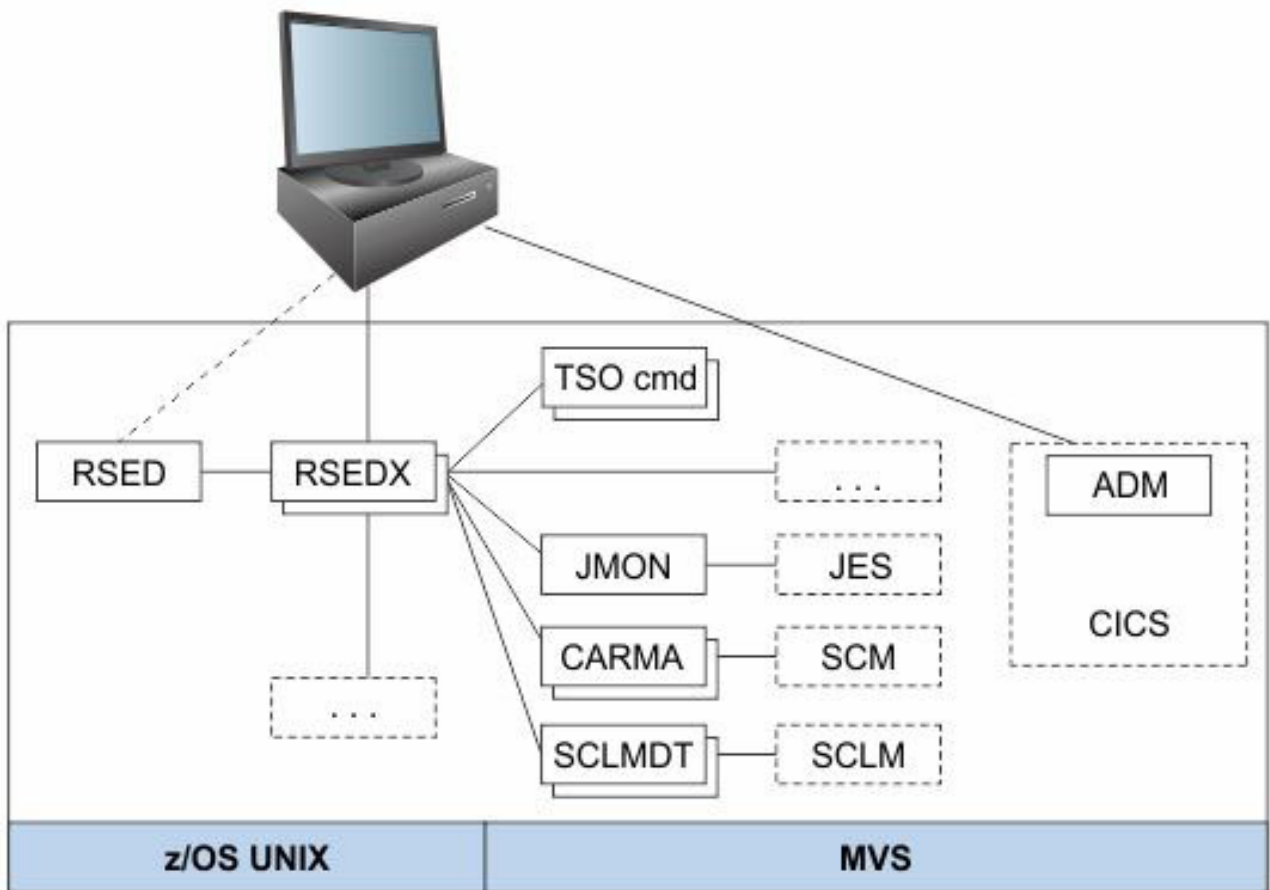


그림 1. 컴포넌트 개요

그림 1은 호스트 시스템의 Developer for System z 레이아웃 개요를 일반적으로 표시합니다.

- RSE(Remote Systems Explorer)는 호스트에 클라이언트 연결, 특정 서비스를 위한 기타 서버 시작과 같은 코어 서비스를 제공합니다. RSE는 다음 두 개의 논리 엔티티로 구성됩니다.
 - RSE 디먼(RSED) - 연결 설정을 관리합니다. RSE 디먼도 단일 서버 모드에서의 실행을 담당합니다. 이를 수행하기 위해 RSE 디먼은 RSE 스레드 풀(RSEDx)로 알려진 하나 이상의 하위 프로세스를 작성합니다.
 - RSE 서버 - 개별 클라이언트 요청을 처리합니다. RSE 서버는 RSE 스레드 풀 내에서 스레드로 활성화됩니다.
- TSO 명령 서비스(TSO cmd)는 TSO 및 ISPF 명령에 일괄처리와 유사한 인터페이스를 제공합니다.
- JES 작업 모니터(JMON)는 모든 JES 관련 서비스를 제공합니다.

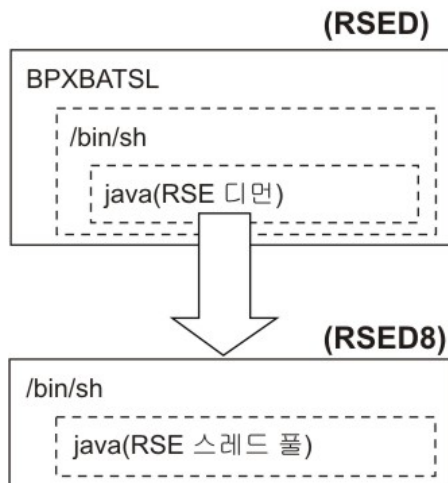
- CARMA(Common Access Repository Manager)는 CA Endeavor와 같은 소프트웨어 구성 관리자(SCM)와 상호작용하기 위한 인터페이스를 제공합니다.
- SCLM 개발자 툴킷(SCLMDT)은 SCLM을 강화하고 이와 상호작용하기 위한 인터페이스를 제공합니다.
- 애플리케이션 배치 관리자(ADM)는 다양한 CICS 관련 서비스를 제공합니다.
- Developer for System z 자체 또는 상호 필수 소프트웨어가 제공할 수 있는 더 많은 서비스를 사용할 수 있습니다.

이전 단락과 목록의 설명은 RSE에 지정된 가장 중요한 역할을 보여줍니다. 몇 가지를 제외하고 모든 클라이언트 통신은 RSE를 통해 진행됩니다. 따라서 제한된 포트 세트만 클라이언트-호스트 통신에 사용되기 때문에 보안 관련 네트워크 설정이 용이합니다.

클라이언트로부터의 연결과 워크로드를 관리하기 위해 RSE는 스레드 풀링 주소 공간을 제어하는 디먼 주소 공간으로 구성됩니다. 디먼은 연결 및 관리를 위한 포컬 포인트의 역할을 하는 반면, 스레드 풀은 클라이언트 워크로드를 처리합니다. 디먼은 `rsed.envvars` 구성 파일에 정의된 값과 실제 클라이언트 연결 수를 기반으로 다중 스레드 풀 주소 공간을 시작할 수 있습니다.

Java 애플리케이션으로서의 RSE

z/OS UNIX 프로세스



Java 스토리지 사용

시스템 - 공유
시스템 - 개인용
코드(z/OS UNIX, Java, RSE)
Java 힙
사용하지 않음

JOBNAME	상태	PID	PPID	명령
RSED	FILE SYS KERNEL WAIT	50331904	1	BPXBATSL
RSED	WAITING FOR CHILD	67109114	50331904	/bin/sh...
RSED	FILE SYS KERNEL WAIT	50331949	67109114	java...
RSED8	WAITING FOR CHILD	307	50331949	/bin/sh...
RSED8	FILE SYS KERNAL WAIT	308	307	java...

그림 2. Java 애플리케이션으로서의 RSE

그림 2는 RSE의 자원 사용량(프로세스와 스토리지)에 대한 기본 보기를 보여줍니다.

RSE는 Java™ 애플리케이션이며 이는 z/OS UNIX 환경에서 활성화됨을 의미합니다. 이는 다른 호스트 플랫폼으로의 간편한 이식과 Developer for System z 클라이언트(Eclipse 프레임워크를 기반으로 하는 또 다른 Java 애플리케이션)와의 직접 통신을 허용합니다. 따라서 Developer for System z를 이해하려는 경우 z/OS UNIX와 Java의 작동 원리에 대한 기본 지식이 많은 도움이 됩니다.

z/OS UNIX에서는 PID(프로세스 ID)로 식별되는 프로세스에서 프로그램이 실행됩니다. 각 프로그램은 해당 프로세스에서 활성화되므로 다른 프로그램을 호출하면 새 프로세스가 작성됩니다. 프로세스를 시작한 프로세스는 PPID(상위 PID)로 참조됩니다. 새 프로세스를 하위 프로세스라고 합니다. 하위 프로세스는 동일한 주소 공간에서 실행되거나 새 주소 공간에서 제공(작성)될 수 있습니다. 동일한 주소 공간에서 실행되는 새 프로세스는 TSO의 명령 실행과 비교할 수 있으며 새 주소 공간의 제공 프로세스는 일괄처리 작업을 제출하는 것과 유사합니다.

프로세스는 단일 스레드 또는 다중 스레드가 가능합니다. 다중 스레드 애플리케이션(예 : RSE)에서는 여러 스레드가 개별 주소 공간(오버헤드가 적음)처럼 시스템 자원을 두고 경쟁합니다.

6 페이지의 그림 2의 RSE 샘플에 이 프로세스 정보를 맵핑하면 다음과 같은 플로우가 진행됩니다.

1. RSED 태스크가 시작되면 BPXBATSL을 실행하여 z/OS UNIX를 호출하고 셸 환경을 작성합니다 - PID 50331904.
2. 이 프로세스에서는 개별 프로세스(/bin/sh)에서 실행되는 rsed.sh 셸 스크립트가 실행됩니다 - PID 67109114.
3. 셸 스크립트는 rsed.envvars에 정의된 환경 변수를 설정하고 필수 매개변수와 함께 Java를 실행하여 RSE 디먼을 시작합니다 - PID 50331949.
4. RSE 디먼은 하위 프로세스(RSED8)에 새 셸을 제공합니다 - PID 307.
5. 이 셸에서는 rsed.envvars에 정의된 환경 변수를 설정하고 필수 매개변수와 함께 Java를 실행하여 RSE 스레드 풀을 시작합니다 - PID 308.

RSE는 31비트 또는 64비트 주소 지정 모드로 실행하여 스토리지 한계가 서로 다르게 할 수 있습니다. 31비트 모드에서 사용 가능한 스토리지는 2GB로 제한되고, 64비트 모드에서는 SYS1.PARMLIB에 지정된 경우가 아니면 한계가 없습니다.

Java 애플리케이션(예: RSE)은 스토리지를 직접 할당하지 않고 Java 메모리 관리 서비스를 사용합니다. 이러한 서비스는 스토리지 할당, 스토리지 비우기, 가비지 콜렉션과 같으며 Java 힙 한계 내에서 실행됩니다. 힙의 최소, 최대 크기는 Java 시작 중에 (내재적으로 또는 명시적으로) 정의됩니다. 64비트 모드로 실행 시 Java가 2GB 막대 위에 힙을 할당하려고 하여 막대 아래 공간을 해제합니다.

이는 사용 가능한 주소 공간 크기를 가장 효과적으로 활용하려면 z/OS가 가변적인 시스템 제어 블록의 양(활성 스레드 수에 따라 다름)을 저장할 수 있는 충분한 공간을 남겨두고 큰 힙 크기를 정의하는 밸런싱 조치를 수행해야 함을 의미합니다.

태스크 소유자

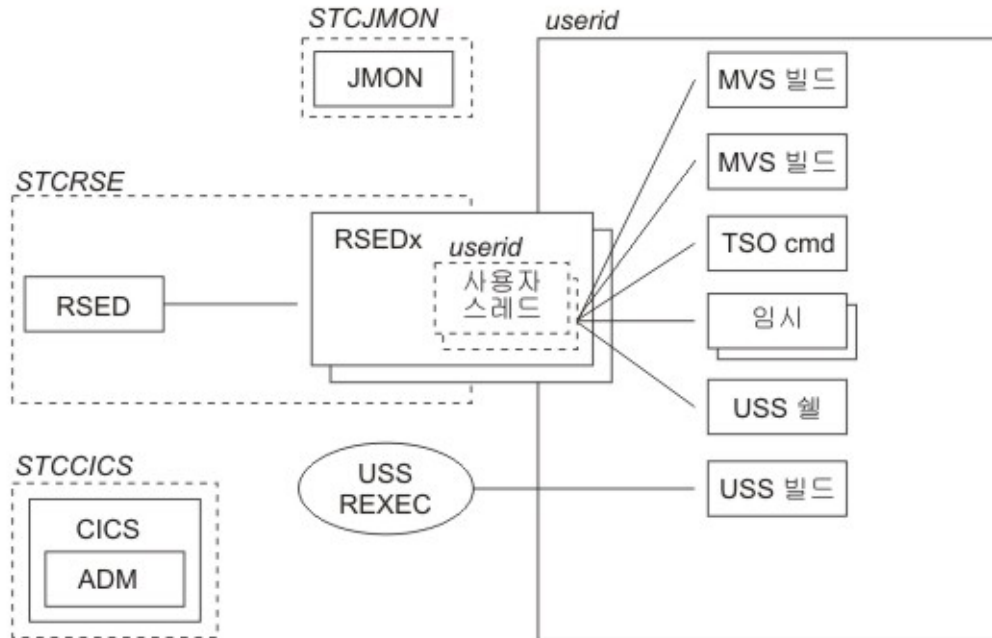


그림 3. 태스크 소유자

그림 3은 다양한 Developer for System z 태스크에 사용되는 보안 신임 정보의 소유자에 대한 기본 개요를 제공합니다.

태스크 소유권은 2개 섹션으로 분할됩니다. 시작 태스크는 보안 소프트웨어에서 시작 태스크에 지정되는 사용자 ID가 소유합니다. RSE 스레드 풀(RSEDx)을 제외한 모든 다른 태스크는 클라이언트 사용자 ID가 소유합니다.

그림 3은 Developer for System z 시작 태스크(JMON 및 RSED)와, Developer for System z가 통신하는 샘플 시작 태스크 및 시스템 서비스를 보여줍니다. 애플리케이션 배치 관리자(ADM)는 CICS 리전 내부에서 활성화됩니다. USS REXEC 태그는 z/OS UNIX REXEC(또는 SSH) 서비스를 나타냅니다.

RSE 디먼(RSED)은 하나 이상의 RSE 스레드 풀 주소 공간(RSEDx)을 작성하여 클라이언트 요청을 처리합니다. 각 RSE 스레드 풀은 여러 클라이언트를 지원하며 RSE 디먼과 동일한 사용자가 소유합니다. 각 클라이언트는 스레드 풀에 자체 스레드를 가지며 이러한 스레드는 클라이언트 사용자 ID가 소유합니다.

클라이언트가 수행하는 조치에 따라 하나 이상의 추가 주소 공간(모두 클라이언트 사용자 ID로 소유)을 시작하여 요청한 조치를 수행할 수 있습니다. 이러한 주소 공간은 MVS 일괄처리 작업, APPC 트랜잭션 또는 z/OS UNIX 하위 프로세스입니다. z/OS UNIX 하위 프로세스는 z/OS UNIX 이니시에이터(BPXAS)에서 활성화되며 하위 프로세스는 JES에서 시작 태스크로 나타납니다.

이러한 주소 공간 작성은 일반적으로 직접 또는 ISPF와 같은 시스템 서비스를 사용하여 스레드 풀의 사용자 스레드로 트리거됩니다. 그러나 주소 공간은 써드파티도 작성할 수 있습니다. 예를 들어, 파일 관리자는 Developer for System z 대신 처리해야 하는 각 데이터 세트(또는 멤버)의 새 주소 공간을 시작합니다. z/OS UNIX REXEC 또는 SSH는 z/OS UNIX에서 빌드를 시작할 때 참여합니다.

사용자별 주소 공간은 태스크 완료 시 또는 비활동 타이머가 만료될 때 종료됩니다. 시작 태스크는 활성 상태를 유지합니다. 8 페이지의 그림 3에 나열된 주소 공간은 시스템에 계속 표시됩니다. 그러나 z/OS UNIX 디자인 방식으로 인해 여러 가지 단기 임시 주소 공간도 존재합니다.

연결 플로우

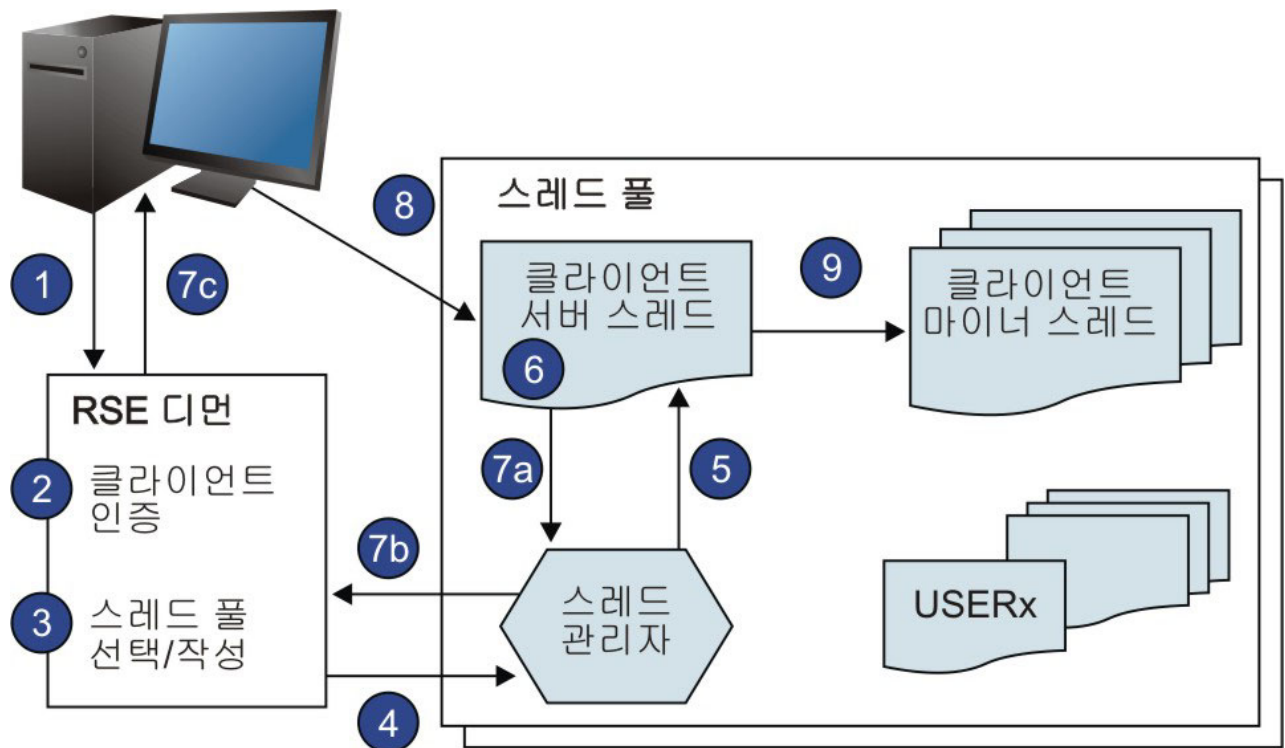


그림 4. 연결 플로우

그림 4는 클라이언트가 Developer for System z를 사용하여 호스트에 연결하는 방법에 대한 개요를 그림으로 표시합니다. PassTicket 사용 방법도 간략히 설명합니다.

1. 클라이언트가 디먼에 로그인합니다(포트 4035).
2. RSE 디먼이 클라이언트가 제공한 신임 정보를 사용하여 클라이언트를 인증합니다.
3. RSE 디먼이 기존 스레드 풀을 선택하거나 모두 가득 차 있으면 새 스레드 풀을 시작합니다.
4. RSE 디먼이 클라이언트 사용자 ID를 스레드 풀로 전달합니다.

5. 스레드 풀이 클라이언트 사용자 ID와 PassTicket을 인증에 사용하여 클라이언트 특정 RSE 서버 스레드를 작성합니다.
6. 클라이언트 서버 스레드가 향후 클라이언트 통신을 위해 포트에 바인드됩니다.
7. 클라이언트 서버 스레드가 클라이언트가 연결할 포트 번호를 리턴합니다.
8. 클라이언트가 RSE 디먼과의 연결을 끊고 제공된 포트 번호에 연결합니다.
9. 클라이언트 서버 스레드가 항상 클라이언트 사용자 ID와 PassTicket을 인증에 사용하여 기타 사용자 특정 스레드(마이너)를 시작합니다. 이 스레드는 클라이언트가 요청한 사용자 특정 서비스를 제공합니다.

위 설명은 스레드 중심 RSE 디자인을 표시합니다. 사용자당 하나의 주소 공간을 시작하는 대신 단일 스레드 풀 주소 공간이 다중 사용자에게 서비스를 제공합니다. 스레드 풀 내에서 사용자의 보안 컨텍스트가 지정된 자체 스레드에서 각 마이너(사용자 특정 서비스)가 활성화되어 보안 설정을 보장합니다. 이 디자인은 자원 사용이 제한된 아주 많은 사용자를 수용하지만 각 클라이언트가 다중 스레드(수행된 태스크에 따라 17개 이상)를 사용함을 의미합니다.

네트워크 관점에서 Developer for system z는 수동 모드의 FTP와 유사하게 작동합니다. 클라이언트는 포컬 포인트(RSE 디먼)에 연결한 후 연결을 중단하고 포컬 포인트가 제공하는 포트 번호에 다시 연결합니다. 다음 논리는 두 번째 연결에 사용되는 포트 선택을 제어합니다.

1. 클라이언트가 서브시스템 특성 탭에 0이 아닌 포트 번호를 지정한 경우, RSE 서버는 해당 포트를 바인드에 사용합니다. 이 포트를 사용할 수 없으면 연결에 실패합니다.
2. `_RSE_PORTRANGE`가 `rsed.envvars`에 지정된 경우, RSE 서버는 이 범위의 포트에 바인드합니다. 사용할 수 있는 포트가 없으면 연결에 실패합니다. RSE 서버는 클라이언트 연결 기간 동안에만 포트가 필요한 것은 아닙니다. 이는 다른 RSE 서버가 포트에 바인드할 수 없는 (서버) 바인드와 (클라이언트) 연결 사이의 기간일 뿐입니다. 이는 대부분의 연결이 범위에 있는 첫 번째 포트를 사용하고 나머지 범위는 다중 동시 로그인인 경우 버퍼가 됨을 의미합니다.
3. 제한사항이 설정되지 않은 경우, RSE 서버는 포트 0에 바인드합니다. 그 결과 TCP/IP가 포트 번호를 선택합니다.

인증이 필요한 모든 z/OS 서비스에 PassTicket을 사용하면 Developer for System z가 비밀번호를 저장하거나 사용자에게 비밀번호를 입력하도록 계속 프롬프트를 표시하지 않고 이러한 서비스를 호출할 수 있습니다. 모든 z/OS 서비스에 PassTicket을 사용하면 로그인 중에 일회성 비밀번호 및 X.509 인증서와 같은 대체 인증 방법도 사용할 수 있습니다.

통합 디버거

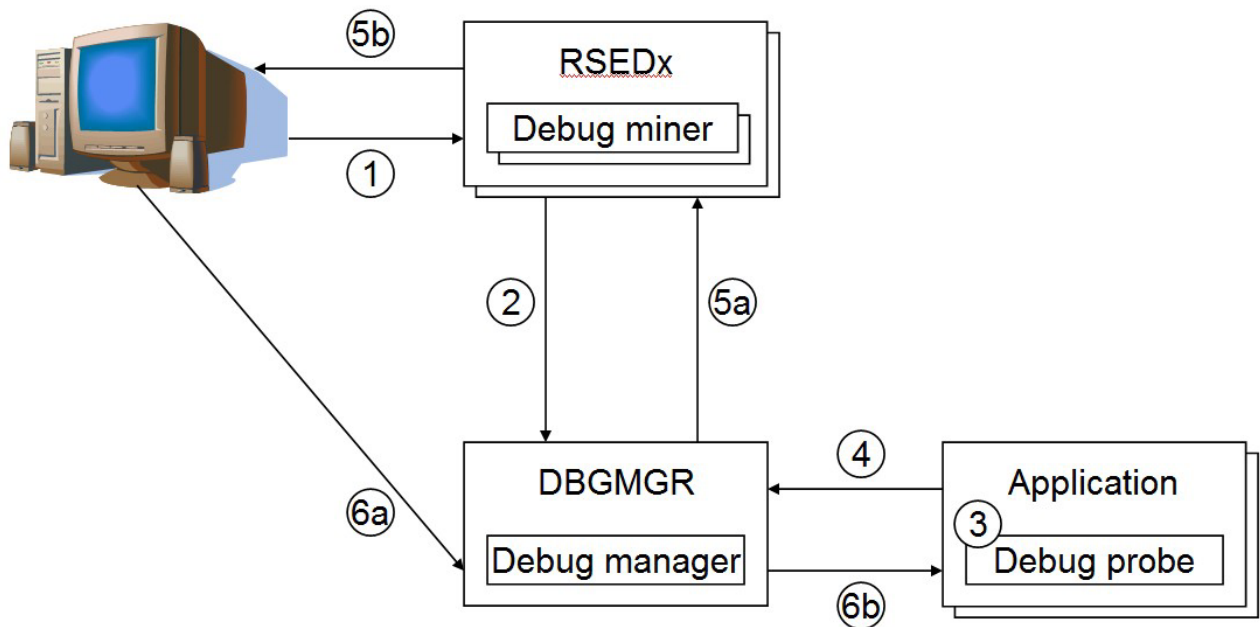


그림 5. 통합 디버거

통합 디버거는 여러 애플리케이션을 디버깅하는 데 사용됩니다. 그림 5는 Developer for System z 클라이언트가 애플리케이션을 디버깅하는 방법에 대한 구조적 개요를 표시합니다.

1. 클라이언트는 일반 Developer for System z 호스트 로그온을 사용하여 호스트에 연결합니다.
2. 로그온의 파트로서, 디버그 마이너는 디버그 관리자에 사용자를 등록하며, 이는 DBGMGR 시작 태스크 내에서 활성화됩니다.
3. 반드시 디버깅되어야 함을 나타내는 표시기와 함께 애플리케이션이 시작되면, LE(Language Environment®)는 디버그 프로브를 호출합니다.
4. 디버그 프로브는 디버그 관리자에 등록됩니다.
5. 디버그 관리자는 디버그 마이너를 사용하여 Developer for System z 클라이언트에게 이 디버그 세션을 수신할 사용자를 알립니다. 이 때 사용자가 등록되지 않으면, 디버그 세션은 휴면 상태가 되어 사용자가 디버그 관리자에 등록할 때까지 기다립니다.
6. 클라이언트 내에서 디버그 엔진은 디버그 관리자에게 문의하고, 교대로 디버그 엔진과 디버그 프로브 사이에서 앞뒤로 데이터를 전달합니다.

CARMA

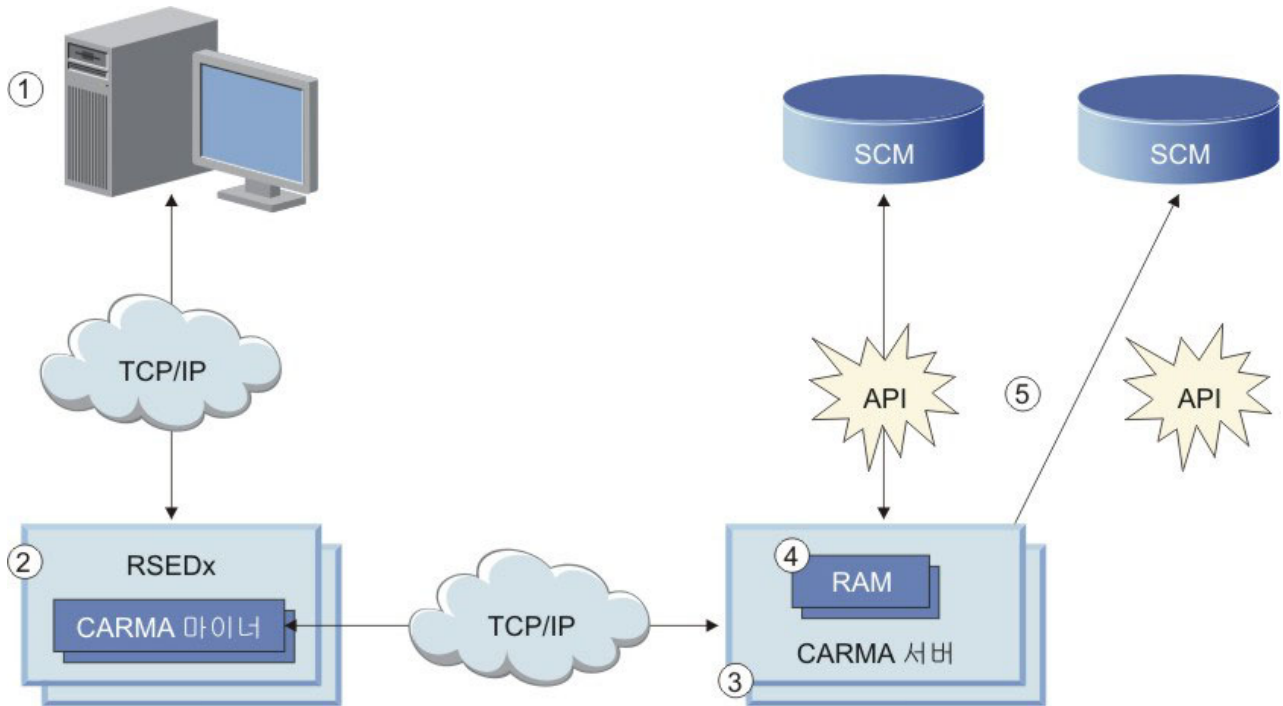


그림 6. CARMA 플로우

CARMA(Common Access Repository Manager)는 호스트 기반 소프트웨어 구성 관리자(SCM)(예: CA Endeavor® SCM)에 액세스하는 데 사용됩니다. 그림 6는 Developer for System z 클라이언트가 지원되는 호스트 기반 소프트웨어 구성 관리자(SCM)에 액세스할 수 있는 방법에 대한 개요를 그림으로 표시합니다.

1. 클라이언트에는 CARMA(Common Access Repository Manager) 플러그인이 있습니다.
2. CARMA 플러그인은 RSE 스프레드 폴(RSEDx) 내에서 사용자 특정 스프레드로 활성화된 CARMA 마이너와 통신합니다. 이 통신은 기존의 RSE 연결을 통해 수행됩니다.
3. 클라이언트가 SCM에 대한 액세스를 요청하면, CARMA 마이너가 TCP/IP 포트에 바인드하고 포트 번호를 시작 인수로 사용하여 사용자 특정 CARMA 서버를 시작합니다. 그러면 CARMA 서버가 이 포트에 연결하고 클라이언트와의 통신에 이 경로를 사용합니다. 호스트 기반 SCM에서 단일 사용자 주소 공간이 해당 서비스에 액세스한다고 예상하는데, 이는 CARMA가 사용자마다 CARMA 서버를 시작해야 합니다. 여러 사용자를 지원하는 단일 서버를 작성할 수 없습니다.
4. CARMA 서버가 요청된 SCM을 지원하는 저장소 액세스 관리자(RAM)를 로드합니다.

5. RAM은 특정 SCM과의 상호작용에 대한 기술적 세부사항을 처리하고 클라이언트에 공통 인터페이스를 제공합니다.

CARMA 구성 파일

Developer for System z는 CARMA 서버를 시작하기 위한 여러 가지 방법을 지원합니다. 방법마다 각각 장점과 단점이 있습니다. Developer for System z는 또한 여러 가지 RAM(Repository Access Manager)을 제공합니다. 이 RAM은 두 그룹(프로덕션 RAM과 샘플 RAM)으로 나눌 수 있습니다. 다양한 RAM 조합과 서버 시작 방법을 사전 구성된 설정으로 사용할 수 있습니다.

모든 서버 시작 방법은 공통 구성 파일, CRASRV.properties를 공유합니다. 이 파일은 무엇보다 사용될 시작 방법을 지정합니다.

CRASTART

"CRASTART" 메소드는 CARMA 서버를 RSE 내 하위 태스크로 시작합니다. 이 메소드는 CARMA 서버를 시작하는 데 필요한 프로그램 호출과 데이터 세트 할당을 정의하는 개별 구성 파일을 사용하여 매우 유연한 설정을 제공합니다. 이 메소드는 최고의 성능을 제공하고 가장 적은 자원을 사용하지만 CRASTART 모듈이 LPA에 있어야 합니다.

RSE는 로드 모듈 CRASTART를 호출하면 이 모듈은 crastart*.conf의 정의를 사용하여 일괄처리 TSO 및 ISPF 명령을 실행할 수 있는 올바른 환경을 작성합니다. Developer for System z는 이 환경을 사용하여 CARMA 서버, CRASERV를 사용합니다. Developer for System z는 각각 특정 RAM에 사전 구성된 여러 crastart*.conf 파일을 제공합니다.

일괄처리 제출

"일괄처리 제출" 방법은 작업을 제출하여 CARMA 서버를 시작합니다. 이 작업은 제공되는 샘플 구성 파일에서 사용되는 기본 방법입니다. 이 방법의 이점은 작업 출력에서 CARMA 로그에 쉽게 액세스할 수 있다는 것입니다. 또한 개발자 자신이 유지관리하는 각 개발자에 대한 사용자 정의 서버 JCL을 사용할 수 있습니다. 그러나 이 방법은 CARMA 서버를 시작하는 개발자당 하나의 JES 이니시에이터를 사용합니다.

RSE가 CLIST CRASUB*를 호출하면 다음으로 임베디드 JCL이 제출되어 일괄처리 TSO 및 ISPF 명령을 실행할 수 있는 올바른 환경을 작성합니다. Developer for System z는 이 환경을 사용하여 CARMA 서버, CRASERV를 실행합니다. Developer for System z는 각각 특정 RAM에 사전 구성된 여러 CRASUB* 멤버를 제공합니다.

데이터 세트 잠금 소유자

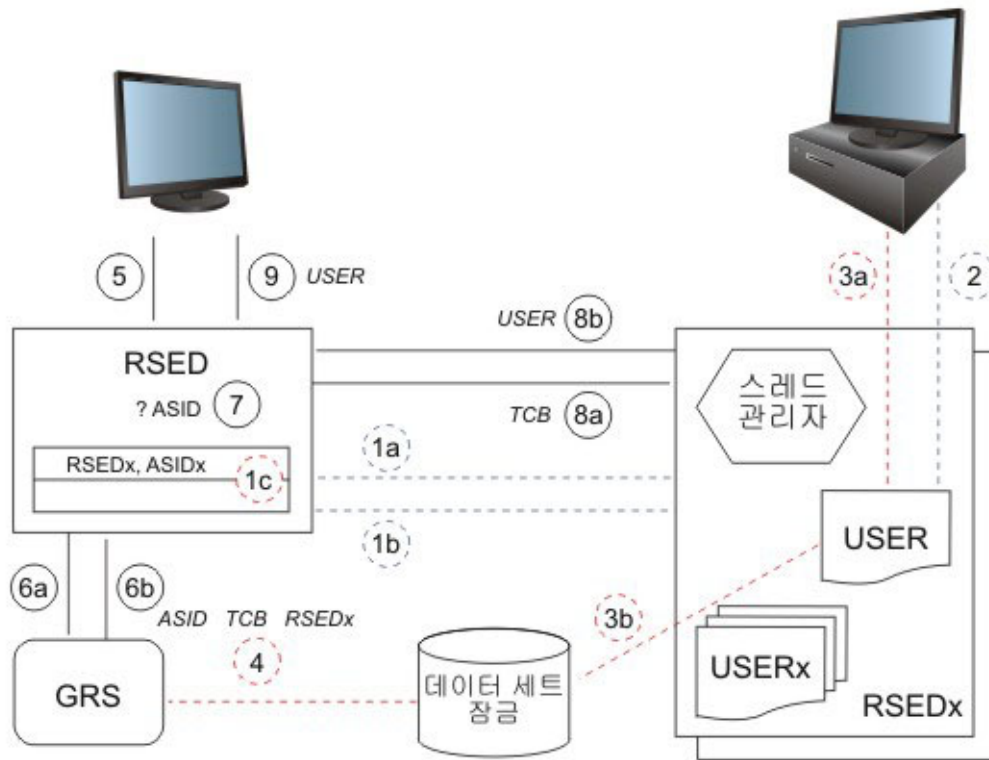


그림 7. 데이터 세트 인큐 판별 플로우

그림 7은 RSE 디먼이 데이터 세트 잠금을 소유하는 Developer for System z 클라이언트를 결정하는 방법의 개요를 그림으로 표시합니다.

1. RSE 디먼(RSED)은 스레드 풀(RSEDx)을 작성합니다. 시작이 완료되었는지 확인하기 위해, 스레드 풀은 RSE 디먼에게 ASID(Address Space Identifier)를 다시 보고하고, 이 스레드 풀 추적을 위해 작성된 제어 블록에 저장합니다.
2. 클라이언트가 로그인하여 스레드 풀(RSEDx) 내부에 사용자별 RSE 서버 스레드(USER)를 작성합니다. 스레드 각각에는 고유한 TCB(Task Control Block) ID가 있습니다.
3. 클라이언트는 데이터 세트를 편집 상태로 열고 RSE 서버에 데이터 세트에 대한 독점 잠금(큐에 넣기)을 가져오도록 지시합니다.
4. 시스템은 큐에 넣기 프로세스의 일부로 요청자의 ASID, TCB 및 태스크 이름(RSEDx)을 등록합니다. 이 정보는 글로벌 자원 직렬화(GRS) 큐에 저장됩니다.
5. 운영자는 RSE 디먼에서 데이터 세트의 잠금 상태를 조회합니다.
6. RSE 디먼이 GRS 큐를 스캔하여 데이터 세트가 잠겼는지 여부를 확인하고 잠금 소유자의 ASID, TCB 및 태스크 이름을 검색합니다.
7. 검색한 ASID는 다른 스레드 풀의 ASID와 비교합니다.

8. RSE 디먼은 ASID를 소유하는 스레드 풀에게 TCB를 소유하고 있는 사용자를 판별하도록 요청합니다.
9. 일치 항목을 찾으면 관련 클라이언트 사용자 ID가 요청자에게 리턴됩니다. 그렇지 않으면 GRS에서 검색된 태스크 이름이 리턴됩니다.

Developer for System z의 단일 서버 설정에서 단일 스레드 풀 주소 공간에 여러 사용자가 지정되는 경우, z/OS가 **DISPLAY GRS,RES=(*,dataset*)** 운영자 명령을 사용하여 데이터 세트 또는 멤버에 대한 잠금을 소유하는 사용자를 추적할 수 없습니다. 시스템 명령은 주소 공간 레벨(스레드 풀)에서 중지됩니다.

이 문제를 해결하기 위해 Developer for System z에서는 *Host Configuration Guide*(SC23-7658)의 "운영자 명령"에 설명된 대로 **MODIFY rsed APPL=DISPLAY OWNER,DATASET=dataset** 운영자 명령을 제공합니다. 운영자 명령은 ISPF와 같이 기타 제품이 수행한 잠금과 함께, RSE 사용자가 수행하는 모든 데이터 세트 및 멤버 잠금을 해결할 수 있습니다.

잠금 해제

일반적으로 클라이언트가 데이터 세트 또는 구성원을 편집 모드에서 열면 해당 데이터 세트 또는 구성원이 잠기고 클라이언트가 편집 세션을 닫으면 잠금이 해제됩니다.

특정 오류 조건 하에서는 이 메커니즘이 설계대로 작동하지 않을 수 있습니다. 이러한 경우 잠금을 유지하는 사용자를 RSE의 **modify cancel** 연산자 명령을 사용하여 취소할 수 있습니다(*Host Configuration Guide* (SC23-7658)에서 "운영자 명령"의 설명 참조). 이 사용자가 소유하는 활성 데이터 세트는 프로세스 중에 잠금이 해제됩니다.

z/OS UNIX 디렉토리 구조

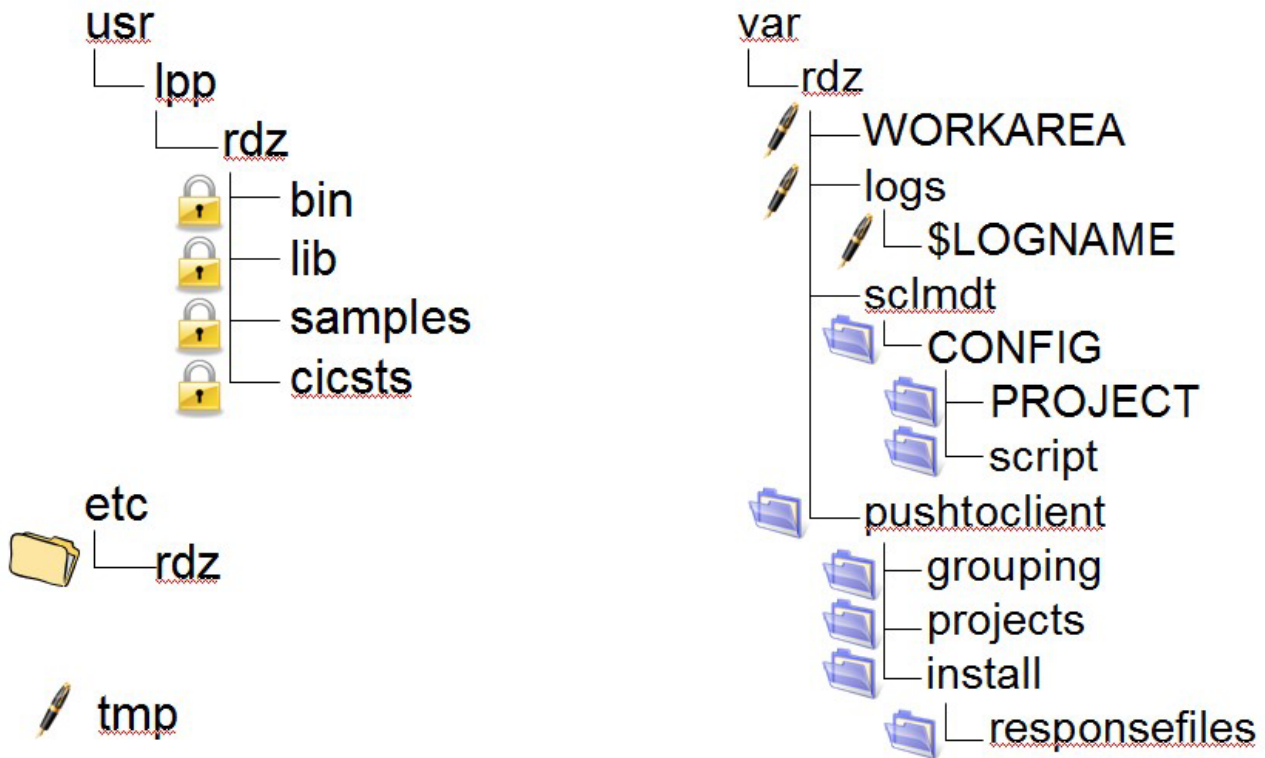


그림 8. z/OS UNIX 디렉토리 구조

그림 8은 Developer for System z에서 사용하는 z/OS UNIX 디렉토리의 개요를 보여줍니다. 다음 목록은 Developer for System z에서 사용하는 각 디렉토리, 위치 변경 방법, 해당 위치에서 데이터를 유지관리하는 사용자에게 대해 설명합니다.

- **/usr/lpp/rdz/**는 Developer for System z 제품 코드의 루트 경로입니다. 실제 위치는 RSED 시작 태스크에서 지정됩니다(변수 HOME). 해당 위치의 파일은 SMP/E로 유지관리합니다.
- **/etc/rdz/**는 RSE와 마이너 관련 구성 파일을 보관합니다. 실제 위치는 RSED 시작 태스크에서 지정됩니다(변수 CNFG). 해당 위치의 파일은 시스템 프로그래머가 유지관리합니다.
- **/tmp/**는 ISPF의 TSO/ISPF Client Gateway와 다양한 마이너가 임시 데이터를 저장하는 데 사용합니다. 일부 IVP는 해당 출력을 여기에 저장합니다. 해당 위치의 파일은 ISPF, 마이너, IVP로 유지관리합니다. 이 위치는 `rsed.envvars`의 `TMPDIR` 변수로 사용자 정의할 수 있습니다. 이 위치는 또한 Java 덤프 파일의 기본 위치이며 `rsed.envvars`의 `_CEE_DUMPTARG` 변수로 사용자 정의할 수 있습니다.

참고: **/tmp/**에는 각 클라이언트가 임시 파일을 작성하기 위해 권한 비트 마스크 777이 필요합니다.

- /var/rdz/WORKAREA/는 ISPF의 TSO/ISPF Client Gateway와 SCLMDT가 z/OS UNIX와 MVS 기반 주소 공간 사이에 데이터를 전송하는 데 사용됩니다. 실제 위치는 rsed.envvars에 지정됩니다(변수 CGI_ISPWORK). 해당 위치의 파일은 ISPF와 SCLMDT로 유지관리합니다.

참고: /var/rdz/WORKAREA/에는 각 클라이언트가 임시 파일을 작성하기 위해 권한 비트 마스크 777이 필요합니다.

- /var/rdz/logs/는 RSE 디먼과 RSE 스레드 풀 서버의 로그를 보관합니다. 실제 위치는 rsed.envvars에 지정됩니다(변수 daemon.log). 해당 위치의 파일은 RSE로 유지관리합니다.
- /var/rdz/logs/\$LOGNAME/은 RSE 서버와 마이너의 사용자별 로그를 보관합니다. 실제 위치는 rsed.envvars에 지정됩니다(변수 user.log 및 DSTORE_LOG_DIRECTORY). 해당 위치의 파일은 RSE와 마이너가 유지관리합니다.

참고: /var/rdz/logs/에는 각 클라이언트가 \$LOGNAME 디렉토리를 작성하고 사용자별 로그 파일을 저장하기 위해 권한 비트 마스크 777이 필요합니다.

- /var/rdz/sclmdt/CONFIG/는 일반 SCLMDT 구성 파일을 보관합니다. 실제 위치는 rsed.envvars에 지정됩니다(변수 SCLMDT_CONF_HOME). 해당 위치의 파일은 SCLM 관리자가 유지관리합니다.
- /var/rdz/sclmdt/CONFIG/PROJECT/는 SCLMDT 프로젝트 구성 파일을 보관합니다. 실제 위치는 rsed.envvars에 지정됩니다(변수 SCLMDT_CONF_HOME). 해당 위치의 파일은 SCLM 관리자가 유지관리합니다.
- /var/rdz/sclmdt/CONFIG/script/는 다른 제품이 사용할 수 있는 SCLMDT 관련 스크립트를 보관합니다. 실제 위치는 지정되지 않습니다. 해당 위치의 파일은 SCLM 관리자가 유지관리합니다.
- /var/rdz/pushtoclient/는 호스트 연결 시 클라이언트에 푸시되는 클라이언트 구성 파일, 클라이언트 제품 업데이트 정보, 호스트 기반 프로젝트 정보를 보관합니다. 실제 위치는 pushtoclient.properties에 지정됩니다(변수 pushtoclient.folder). 해당 위치의 파일은 Developer for System z 클라이언트 관리자가 유지관리합니다.
- /var/rdz/pushtoclient/grouping/은 호스트 연결 시 클라이언트에 푸시되는 그룹별 클라이언트 구성 파일, 클라이언트 제품 업데이트 정보, 호스트 기반 프로젝트 정보를 보관합니다. 실제 위치는 pushtoclient.properties에 지정됩니다(변수 pushtoclient.folder와 접미부 /grouping). 해당 위치의 파일은 Developer for System z 클라이언트 관리자가 유지관리합니다.
- /var/rdz/pushtoclient/projects/는 호스트 기반 프로젝트 정의 파일을 보관합니다. 실제 위치는 /var/rdz/pushtoclient/keymapping.xml에 지정됩니다. 이 파일은 Developer for System z 클라이언트 관리자가 작성, 유지관리합니다. 해당 위치의 파일은 프로젝트 관리자 또는 리드 개발자가 유지관리합니다.

- /var/rdz/pushtoclient/install/은 호스트에 대한 연결 시 클라이언트 제품 버전을 업데이트하는 데 사용되는 구성 파일을 보관합니다. 실제 위치는 /var/rdz/pushtoclient/keymapping.xml에 지정됩니다. 이 파일은 Developer for System z 클라이언트 관리자가 작성, 유지관리합니다. 해당 위치의 파일은 클라이언트 관리자가 유지관리합니다.
- /var/rdz/pushtoclient/install/responsefiles/는 호스트에 대한 연결 시 클라이언트 제품 버전을 업데이트하는 데 사용되는 구성 파일을 보관합니다. 실제 위치는 /var/rdz/pushtoclient/keymapping.xml에 지정됩니다. 이 파일은 Developer for System z 클라이언트 관리자가 작성, 유지관리합니다. 해당 위치의 파일은 클라이언트 관리자가 유지관리합니다.

비시스템 관리자에 대한 업데이트 권한

/var/rdz/pushtoclient/의 데이터는 프로젝트 관리자처럼 z/OS UNIX의 업데이트 권한이 많지 않은 비시스템 관리자가 유지관리합니다. 따라서 실행 가능하면서도 안전한 설정을 유지하려면 z/OS UNIX에서 파일 작성 중에 액세스 권한을 설정하는 방법을 이해해야 합니다.

UNIX 표준에는 3가지 사용자 유형(소유자, 그룹, 기타)에 대한 권한을 설정할 수 있는 것으로 규정되어 있습니다. 각 유형마다 개발적으로 읽기, 쓰기, 실행 권한을 설정할 수 있습니다.

z/OS UNIX는 파일 작성 시 UID(사용자 ID)와 GID(그룹 ID)를 다음 값으로 설정합니다.

- UID는 작성 스레드의 유효 UID로 설정됩니다.
- GID는 소유 디렉토리의 GID로 설정됩니다. 보안 프로파일 FILE.GROUPOWNER.SETGID가 UNIXPRIV 클래스에 정의되는 경우에는 대신 작성 스레드의 유효 GID가 기본적으로 사용됩니다. 세부사항은 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.

각 사이트는 자체 기본 액세스 권한 마스크를 설정할 수 있지만 공통 마스크는 소유자에 대한 읽기/쓰기 권한과 그룹 등에 대한 읽기 권한을 허용합니다.

/var/rdz/pushtoclient/의 데이터는 pushtoclient.properties의 file.permission 지시문에 정의된 액세스 권한 마스크를 사용하여 작성됩니다. 기본 값은 소유자와 그룹에 대한 읽기, 쓰기 권한과 다른 대상에 대한 읽기 권한을 허용합니다. 실행 권한은 모든 파일 또는 디렉토리에 있습니다. 최종 액세스 권한은 모두에게 읽기, 실행 권한을 허용하고 데이터를 유지관리하는 Developer for System z 클라이언트 관리자에게는 쓰기 권한을 허용해야 합니다.

/var/rdz/pushtoclient/projects/의 데이터는 특정 액세스 권한 마스크를 사용하지 않고 작성됩니다. 최종 액세스 권한은 모두에게 읽기 권한을 허용하고 데이터를 유지관리하는 프로젝트 관리자에게는 쓰기 권한을 허용해야 합니다.

유용한 보안 명령

프로젝트 관리자 또는 Developer for System z 클라이언트 관리자가 이러한 디렉토리의 데이터를 관리하려면 보안 관리자가 올바른 해당 OMVS 세그먼트를 갖는 그룹을 작성해야 합니다. 이 그룹은 일반적으로 관련 사용자 ID의 기본 그룹입니다. 다음 샘플 RACF® 명령에 대한 자세한 정보는 *Security Server RACF Command Language Reference*(SA22-7687)를 참조하십시오.

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

유용한 z/OS UNIX 명령

다음 샘플 z/OS UNIX 명령에 대한 자세한 정보는 *UNIX System Services Command Reference*(SA22-7802)를 참조하십시오.

- 다음 z/OS UNIX **ls** 명령을 사용하면 디렉토리 내 모든 파일을 표시할 수 있습니다.

```
ls -lR /var/rdz/pushtoclient/
```

- 다음 z/OS UNIX **chown** 명령을 사용하면 디렉토리와 디렉토리 내 모든 파일의 소유자를 변경할 수 있습니다.

```
chown -R IBMUSER /var/rdz/pushtoclient/
```

- 다음 z/OS UNIX **chgrp** 명령을 사용하면 디렉토리와 디렉토리 내 모든 파일에 그룹을 지정할 수 있습니다.

```
chgrp -R RDZPROJ /var/rdz/pushtoclient/
```

- 다음 z/OS UNIX **chmod** 명령을 사용하면 디렉토리와 디렉토리 내 모든 파일에 소유자 및 그룹 쓰기 권한을 부여할 수 있습니다. 읽기 권한은 다른 파일 또는 디렉토리에 있습니다. 실행 권한은 모든 파일 또는 디렉토리에 있습니다.

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

샘플 설정

다음 시나리오에서는 Developer for System z 클라이언트 관리자를 포함하여 세 명의 개발 프로젝트 관리자가 모두 태스크를 수행해야 합니다.

보안 관리자는 이미 팀에 대해 고유 그룹 ID(1200)를 갖는 기본 그룹(RDZPROJ)을 지정했습니다. 해당 사용자 ID에는 z/OS UNIX의 특수 권한(예: UID 0)이 없습니다. 보안 관리자는 FILE.GROUPOWNER.SETGID 프로파일을 정의하지 않았으므로 z/OS UNIX는

새 파일을 작성할 때 디렉토리의 그룹 ID를 사용합니다. 시스템 프로그래머가 /var/rdz/pushtoclient 디렉토리를 작성하기 위해 사용자 ID IBMUSER(UID는 0이고 기본 그룹은 SYS1)를 사용했습니다.

1. 시스템 프로그래머는 소유자 및 그룹에 대한 /var/rdz/pushtoclient 쓰기 권한을 제한합니다.

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

참고: 사용자 정의 설정 중에 사용된 FEKSETUP 작업은 이미 이 단계를 수행합니다.

2. 시스템 프로그래머는 RDZPROJ를 소유 그룹으로 지정합니다.

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

이렇게 하면 /var/rdz/pushtoclient 쓰기 권한을 시스템 프로그래머(IBMUSER)와 프로젝트 관리자(RDZPROJ)로 제한하는 데 필요한 설정이 완료됩니다.

제 2 장 보안 고려사항

Developer for System z는 비메인프레임 워크스테이션 사용자에게 메인프레임 액세스를 제공합니다. 따라서 연결 요청 유효성 검증, 호스트와 워크스테이션 간의 보안 통신 제공, 권한 부여 및 감사 활동은 제품 구성의 중요한 측면입니다.

Developer for System z 서버 및 서비스에서 사용하는 보안 메커니즘은 보안 상태로 상주하는 파일 시스템 및 데이터 세트에 따라 다릅니다. 이는 신뢰할 수 있는 시스템 관리자만 프로그램 라이브러리와 구성 파일을 업데이트할 수 있어야 함을 의미합니다.

이 장에서 다루는 주제는 다음과 같습니다.

- 22 페이지의 『인증 방법』
- 23 페이지의 『연결 보안』
- 25 페이지의 『PassTicket 사용』
- 26 페이지의 『감사 로깅』
- 29 페이지의 『JES 보안』
- 33 페이지의 『SSL/TLS 암호화된 통신』
- 35 페이지의 『X.509 인증서를 사용한 클라이언트 인증』
- 39 페이지의 『POE(Port Of Entry) 확인』
- 43 페이지의 『기타 정보』
- 40 페이지의 『클라이언트 기능 변경』
- 41 페이지의 『클라이언트로 푸시 개발자 그룹』
- 42 페이지의 『디버그 보안』
- 43 페이지의 『CICSTS 보안』
- 44 페이지의 『SCLM 보안』
- 44 페이지의 『Developer for System z 구성 파일』
- 48 페이지의 『보안 정의』

참고: 호스트에 클라이언트 연결과 같은 코어 서비스를 제공하는 원격 시스템 탐색기(RSE)는 다음과 같은 2가지 논리 엔티티로 구성됩니다.

- 연결 설정을 관리하는 RSE 디먼은 시작 태스크 또는 장기 실행 사용자 작업으로 시작됩니다.
- 개별 클라이언트 요청을 처리하는 RSE 서버는 RSE 디먼에 의한 하나 이상의 하위 프로세스에서 스레드로 시작됩니다.

Developer for System z의 기본 디자인 개념에 대해 알아보려면 3 페이지의 제 1 장 『Developer for System z 이해』를 참조하십시오.

인증 방법

Developer for System z는 연결 시 클라이언트가 제공하는 사용자 ID를 인증하는 여러 가지 방법을 지원합니다.

- 사용자 ID와 비밀번호
- 사용자 ID와 일회성 비밀번호
- X.509 인증

참고: 클라이언트가 제공하는 인증 데이터는 초기 연결 설정 중에 한 번만 사용됩니다. 사용자 ID를 인증한 후에는 인증이 필요한 모든 작업에 사용자 ID와 자체 생성된 PassTicket이 사용됩니다.

사용자 ID와 비밀번호

클라이언트는 연결 시 사용자 ID와 해당 비밀번호를 제공합니다. 사용자 ID와 비밀번호는 보안 제품에 해당 사용자를 인증하는 데 사용됩니다.

사용자 ID와 일회성 비밀번호

고유 토큰에 따라 써드파티 제품으로 일회성 비밀번호를 생성할 수 있습니다. 고유 토큰은 사용자 지식 없이 복사, 사용할 수 없고 인터셉트된 비밀번호는 일회에 한해서만 유효하여 쓸모가 없으므로 일회성 비밀번호는 보안 설정을 향상시킵니다.

클라이언트는 연결 시 사용자 ID와, 써드파티에서 제공하는 보안 종료로 해당 사용자 ID를 인증하는 데 사용되는 일회성 비밀번호를 제공합니다. 이 보안 종료는 정상 처리 중에 인증 요청을 충족하는 데 사용되는 PassTicket을 무시합니다. PassTicket은 보안 소프트웨어로 처리해야 합니다.

X.509 인증

써드파티에서 사용자를 인증하는 데 사용할 수 있는 하나 이상의 X.509 인증을 제공할 수 있습니다. X.509 인증을 보안 장치에 저장하면 사용자가 쉽게 사용할 수 있도록 보안 설정이 결합되어 사용자 ID 또는 비밀번호가 필요하지 않습니다.

연결 시 클라이언트에서 선택한 인증서를 제공하고, 경우에 따라 보안 제품을 사용하여 사용자 ID를 인증하는 데 사용되는 선택된 확장을 제공합니다.

참고: 이 인증 방법은 RSE 디먼 연결 방법만 지원하므로 SSL(Secure Socket Layer) 통신을 사용해야 합니다.

JES 작업 모니터 인증

클라이언트 인증은 클라이언트 연결 요청의 일부로 RSE 디먼(또는 REXEC/SSH)에서 수행됩니다. 사용자가 인증되면 JES 작업 모니터에 대한 자동 로그온을 포함하여 모든 향후 인증 요청에 자체 생성 PassTicket이 사용됩니다.

JES 작업 모니터에서 RSE에 표시된 PassTicket과 사용자 ID의 유효성을 검증하려면 JES 작업 모니터에서 PassTicket을 평가할 수 있어야 합니다. 이는 다음과 같은 의미를 갖습니다.

- 로드 모듈 FEJMON(기본 위치는 로드 라이브러리 FEK.SFEKAUTH)은 APF 인증을 받아야 합니다.
- RSE와 JES 작업 모니터 모두 동일한 애플리케이션 ID(APPLID)를 사용해야 합니다. 기본적으로 두 서버 모두 FEKAPPL을 APPLID로 사용하지만 `rsed.envvars`(RSE의 경우)와 `FEJCNFG`(JES 작업 모니터의 경우)의 APPLID 지시문으로 이를 변경할 수 있습니다.

참고: 이전 클라이언트(버전 7.0 이하)는 JES 작업 모니터와 직접 통신합니다. 이러한 연결의 경우 사용자 ID와 비밀번호 인증 방법만 지원됩니다.

디버그 관리자 인증

클라이언트 인증은 클라이언트 연결 요청의 일부로 RSE 디먼(또는 REXEC/SSH)에서 수행됩니다. 사용자가 인증되면 디버그 관리자에 대한 자동 로그온을 포함하여 모든 향후 인증 요청에 자체 생성 PassTicket이 사용됩니다.

디버그 관리자에서 RSE에 표시된 PassTicket과 사용자 ID의 유효성을 검증하려면 디버그 관리자에서 PassTicket을 평가할 수 있어야 합니다. 즉 로드 모듈 AQEZPCM(기본 위치: 로드 라이브러리 FEK.SFEKAUTH)이 APF 인증을 받아야 합니다.

연결 보안

RSE에서는 서로 다른 레벨의 통신 보안을 지원하며, 이는 대부분의 Developer for System z 서비스와 클라이언트 사이의 통신을 제어합니다.

- 외부(클라이언트-호스트) 통신을 지정된 포트로 제한할 수 있습니다. 이 기능은 기본적으로 사용할 수 없습니다.
- 외부(클라이언트-호스트) 통신은 SSL 또는 TLS를 사용하여 암호화될 수 있습니다. 이 기능은 기본적으로 사용할 수 없습니다.
- Port Of Entry(POE) 검사를 사용하여 신뢰할 수 있는 TCP/IP 주소에 대해서만 호스트 액세스를 허용할 수 있습니다. 이 기능은 기본적으로 사용할 수 없습니다.

일부 선택적 Developer for System z 서비스는 별도의 외부(클라이언트-호스트) 통신 경로를 사용합니다.

- 통합 디버거 통신은 TLS를 사용하여 암호화될 수 있습니다.
- 애플리케이션 배치 관리자 통신은 웹 서비스 인터페이스를 사용하는 경우 SSL을 사용하여 암호화될 수 있습니다.

Developer for System z에서는 TN3270 서버와 같은 써드파티 제품을 사용하여 일부 서비스를 제공합니다. 연결 보안 옵션에 대해서는 관련 제품 문서를 참조하십시오.

지정된 포트로 외부 통신 제한

시스템 프로그래머는 RSE 서버가 클라이언트와 통신할 수 있는 포트를 지정할 수 있습니다. 기본적으로 사용 가능한 모든 포트를 사용합니다. 이 포트 범위는 RSE 디먼 포트와 연결되지 않습니다.

포트 사용법에 대한 이해를 돕기 위해 RSE의 연결 프로세스에 대한 간략한 설명이 아래에 나와 있습니다.

1. 클라이언트가 호스트 포트 4035, RSE 디먼에 연결됩니다.
2. RSE 디먼은 RSE 서버 스레드를 작성합니다.
3. RSE 서버는 연결할 클라이언트의 호스트 포트를 엽니다. 이 포트 선택은 사용자가 서브시스템 특성 탭의 클라이언트에서(권장되지 않음) 또는 `rsed.envvars`의 `_RSE_PORTRANGE` 정의를 통해 구성할 수 있습니다.
4. RSE 디먼은 포트 번호를 클라이언트로 리턴합니다.
5. 클라이언트는 호스트 포트에 연결됩니다.

참고: 이 프로세스는 REXEC/SSH를 사용하는 (선택적) 대체 연결 방법의 경우와 유사합니다(*Host Configuration Guide* (SC23-7658)의 "(선택사항) REXEC(또는 SSH) 사용" 설명 참조).

SSL 또는 TLS를 사용한 통신 암호화

RSE를 통해 전달되는 모든 외부 Developer for System z 데이터 스트림은 SSL(Secure Socket Layer) 또는 TLS(Transport Layer Security)를 통해 암호화할 수 있습니다. 암호화된 통신 사용은 33 페이지의 『SSL/TLS 암호화된 통신』에서 설명한 대로 `ssl.properties` 구성의 설정으로 제어합니다. `rsed.envvars`의 `_RSE_JAVAOPTS` 지시문에서 `DSTORE_SSL_ALGORITHM` 변수를 사용하면 암호화 메소드로서 SSL과 해당 계승자 TLS 중에 선택할 수 있습니다. 이 내용은 *Host Configuration Guide*(SC23-7658)에 있는 "`_RSE_JAVAOPTS`로 기타 Java 시작 매개변수 정의"에서 설명합니다.

클라이언트의 통합 디버거 엔진은 호스트의 디버그 관리자에 연결됩니다. SSL 또는 TLS 사용은 AT-TLS(Application Transparent TLS) 정책으로 제어합니다.

클라이언트의 호스트 연결 에뮬레이터는 호스트의 TN3270 서버에 연결됩니다. SSL 또는 TLS의 사용은 *Communications Server IP Configuration Guide*(SC31-8775)에서 설명한 대로 TN3270에 의해 제어됩니다.

z/OS UNIX 서브프로젝트에서 원격(호스트 기반) 조치는 호스트에서 REXEC 또는 SSH 서버를 사용합니다. SSH 통신은 항상 SSL을 사용하여 암호화됩니다.

애플리케이션 배치 관리자 클라이언트는 CICS TS 웹 서비스 또는 RESTful 인터페이스를 사용하여 애플리케이션 배치 관리자 호스트 서비스를 호출합니다. SSL 사용은 CICS TS로 제어합니다(CICS TS용 RACF 보안 안내서의 설명 참조).

POE(Port Of Entry) 확인

Developer for System z는 신뢰 TCP/IP 주소에 대한 호스트 액세스만 허용하는 POE(Port Of Entry) 확인을 지원합니다. POE 사용은 보안 소프트웨어의 특정 프로파일 정의와 rsed.envvars의 enable.port.of.entry 지시문으로 제어됩니다(39 페이지의 『POE(Port Of Entry) 확인』의 설명 참조).

POE를 활성화하면 POE 확인을 지원하는 다른 TCPIP 애플리케이션(예: INETD)에 영향을 줍니다.

PassTicket 사용

로그온한 후 PassTicket을 사용하여 RSE 서버 내에서 스프레드 보안을 설정합니다. 이 기능은 사용 불가능으로 설정할 수 없습니다. PassTicket은 수명이 약 10분인 시스템 생성 비밀번호입니다. 생성된 PassTicket은 DES 암호화 알고리즘, 사용자 ID, 애플리케이션 ID, 시간 및 날짜 소인, 비밀 키를 기반으로 합니다. 이 비밀 키는 보안 소프트웨어에 정의되어야 하는 64비트 숫자(16진수 문자)입니다. 추가 보안을 위해 z/OS 보안 소프트웨어는 기본적으로 PassTicket을 일회용 비밀번호로 사용합니다.

PassTicket 사용법에 대한 이해를 돕기 위해 RSE의 보안 프로세스에 대한 간략한 설명이 아래에 나와 있습니다.

1. 클라이언트가 호스트 포트 4035, RSE 디먼에 연결됩니다.
2. RSE 디먼은 클라이언트가 제공하는 신임 정보를 사용하여 클라이언트를 인증합니다.
3. RSE 디먼은 고유 클라이언트 ID와 RSE 서버 스프레드를 작성합니다.
4. RSE 서버는 PassTicket을 생성하고 PassTicket을 비밀번호로 사용하여 클라이언트의 보안 환경을 작성합니다.
5. 클라이언트는 RSE 디먼이 리턴하는 호스트 포트에 연결됩니다.
6. RSE 서버는 클라이언트 ID를 사용하여 클라이언트 유효성을 검증합니다.
7. RSE 서버는 새로 생성된 PassTicket을 비밀번호가 필요한 모든 향후 조치에 대한 비밀번호로 사용합니다.

참고: 디버그 관리자와의 보안 연결을 설정하는 데에 유사한 메커니즘이 사용됩니다.

SAF 준수 보안 제품은 PassTicket과 일반 비밀번호를 모두 평가할 수 있으므로 초기 인증 후에는 클라이언트의 실제 비밀번호가 더 이상 필요하지 않습니다. RSE 서버는 비밀번호가 필요할 때마다 PassTicket을 생성하고 사용하므로 클라이언트에 유효한 (임시) 비밀번호가 생성됩니다.

PassTicket을 사용하면 모든 사용자 ID와 비밀번호를 테이블에 저장하지 않고(절충 가능) RSE가 사용자별 보안 환경을 원하는 대로 설정할 수 있습니다. 또한 X.509 인증서와 같이 재사용 가능 비밀번호를 사용하지 않는 클라이언트 인증 방법을 허용합니다.

APPL 및 PTKTDATA 클래스의 보안 프로파일은 PassTicket을 사용할 수 있어야 합니다. 이러한 프로파일은 애플리케이션에 따라 다르므로 현재 시스템 설정에 영향을 주지 않습니다.

PassTicket이 애플리케이션에 따라 다르다는 것은 RSE와 JES 작업 모니터 모두 동일한 애플리케이션 ID(APPLID)를 사용해야 함을 의미합니다. 기본적으로 두 서버 모두 FEKAPPL을 APPLID로 사용하지만 rsed.envvars(RSE의 경우)와 FEJJCNFG(JES 작업 모니터의 경우)의 APPLID 지시문으로 이를 변경할 수 있습니다.

대부분의 z/OS UNIX 애플리케이션 비밀 키를 열기 때문에 OMVSAPPL을 애플리케이션 ID로 사용해서는 안됩니다. 사용자 일괄처리 작업을 포함하여 대부분의 MVS 애플리케이션 비밀 키를 열기 때문에 기본 MVS 애플리케이션 ID(MVS 다음에 시스템의 SMF ID가 옴)도 사용해서는 안됩니다.

PassTicket 시간소인의 최소 단위는 1초입니다. 이는 동일한 사용자 ID에 대해 동일한 애플리케이션이 1초 이내에 생성한 모든 PassTicket은 동일함을 의미합니다. 이는 PassTicket을 일회용 비밀번호로 처리하는 z/OS 보안 소프트웨어와 함께 로그인 중에 Developer for System z 관련 문제점을 야기합니다. 이는 1초 이내에 여러 PassTicket이 필요하기 때문입니다. 따라서 Developer for System z는 생성된 PassTicket을 재사용할 수 있도록 허용하는 플래그를 PassTicket 정의에 설정해야 합니다.

경고: PassTicket이 올바르게 설정되지 않으면 클라이언트 연결 요청이 실패합니다.

감사 로깅

Developer for System z는 RSE 디먼이 관리하는 조치에 대한 감사 로깅을 지원합니다. 감사 로그는 CSV(Comma Separated Value) 형식을 사용하여 디먼 로그 디렉토리에 텍스트 파일로 저장됩니다.

감사 제어

`rsed.envvars`의 여러 옵션이 감사 기능에 영향을 줍니다(*Host Configuration Guide* (SC23-7658)의 "`_RSE_JAVAOPTS`를 사용하여 추가 Java 시작 매개변수 정의" 설명 참조).

- 감사 기능은 `enable.audit.log` 옵션으로 사용/사용하지 않을 수 있습니다.
- 감사 기본값은 `audit.*` 옵션으로 제어합니다.
- 감사 로그 파일 위치는 `daemon.log` 옵션으로 제어합니다.
- 감사 로그 작성에 사용되는 코드 페이지는 `_RSE_HOST_CODEPAGE` 지시문으로 제어됩니다(*Host Configuration Guide* (SC23-7658)의 "`rsed.envvars`, RSE 구성 파일" 참조).

modify switch 연산자 명령은 새 감사 로그 파일로 수동 전환하는 데 사용할 수 있습니다(*Host Configuration Guide* (SC23-7658)의 "운영자 명령" 참조).

감사 로그 파일을 보유하는 파일 시스템이 여유 공간에서 많이 실행되지 않으면 콘솔로 경고 메시지를 보냅니다. 이 콘솔 메시지(FEK103E)는 공간 부족 문제가 해결될 때까지 주기적으로 반복됩니다. RSE에서 생성되는 콘솔 메시지의 목록은 *Host Configuration Guide* (SC23-7658)의 "콘솔 메시지"를 참조하십시오.

감사 처리

사전 결정된 시간이 경과하거나 **modify switch** 연산자 명령이 실행되면 새 감사 로그 파일이 시작됩니다. 이전 로그 파일은 `audit.log.yyyymmdd.hhmmss`로 저장됩니다. 여기서 `yyymmdd.hhmmss`는 해당 로그를 닫은 날짜/시간소인입니다. 파일에 지정된 시스템 날짜/시간소인은 로그 파일 작성을 나타냅니다. 두 날짜의 조합은 이 감사 로그 파일이 적용되는 기간을 보여줍니다.

`rsed.envvars`의 `audit.action*` 지시문을 사용하면 감사 로그가 닫힐 때 RSE에서 호출되는 사용자 엑시트(z/OS UNIX 셸 스크립트, z/OS UNIX REXX 또는 z/OS UNIX 프로그램을)를 지정할 수 있습니다. 이 사용자 엑시트는 감사 로그 내 데이터를 처리할 수 있습니다.

감사 로그 파일에는 권한 비트 마스크 640 (-rw-r-----)가 있습니다(`rsed.envvars`의 `audit.log.mode` 지시문으로 변경되지 않은 경우). 이는 소유자(RSE 디먼 z/OS UNIX UID)에게 읽기 및 쓰기 액세스 권한이 있고 소유자의 (기본) 그룹에 읽기 액세스 권한이 있음을 의미합니다. 다른 모든 액세스 시도는 무시됩니다(수퍼유저(UID 0) 또는 UNIXPRIV 보안 클래스의 SUPERUSER.FILESYS 프로파일에 대한 충분한 권한이 있는 사용자가 수행하지 않는 경우).

감사 데이터

로그되는 조치는 다음과 같습니다.

- 시스템 액세스(연결, 연결 끊기)
- JES 스푼 액세스(제출, 표시, 보류, 해제, 취소, 제거)
- 데이터 세트 액세스(읽기, 쓰기, 작성, 삭제, 이름 바꾸기, 압축, 마이그레이션, 재호출)
- 파일 액세스(읽기, 쓰기, 작성, 삭제, 이름 바꾸기)
- TSO 및 z/OS UNIX 명령 실행

로그된 각 조치는 CSV(Comma Separated Value) 형식을 사용하여 날짜/시간소인과 함께 저장됩니다(자동화 또는 데이터 분석 도구를 사용하여 읽을 수 있음). 예를 들면, 다음과 같습니다.

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

파일을 열면 데이터 세트 및 멤버 통계도 로그됩니다. 통계는 READ 조치 완료를 설명하는 행에 추가되며 필드는 %n으로 구분됩니다. 예를 들면, 다음과 같습니다.

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

다음 속성은 나열된 순서로 로그됩니다.

- 작성 날짜 및 시간(mm/dd/yyyy hh:mm)
- 마지막 수정 날짜 및 시간(mm/dd/yyyy hh:mm:ss)
- 마지막 액세스 날짜 및 시간(mm/dd/yyyy hh:mm:ss)
- 레코드 형식(RECFM)
- SCLM 개정 표시기(N = 개정 번호가 설정됨, D = 개정 번호가 설정되지 않음)
- SCLM 개정 번호
- "잘못된 16진" 문자 포함(Y = 예, N = 아니오)

참고: "잘못된 16진" 문자의 경우 코드 페이지 불일치로 인해 클라이언트 왕복이 불가능하기 때문에 Developer for System z 맵핑 서비스가 필요합니다.

- 논리적 레코드 길이(LRECL)
- 파일 크기
- 향후 사용을 위해 예약됨
- 향후 사용을 위해 예약됨
- 사용자 ID
- 이 데이터 세트 또는 멤버에 대한 소유자 잠금(큐에 넣기)
- CR(캐리지 리턴), LF(줄 바꾸기), NL(줄 바꾸기) 호스트 코드 포인트 및 대체 문자 (클라이언트 버전 8.0.3 이상을 사용하는 경우에만 사용 가능)

JES 보안

Developer for System z는 클라이언트가 JES 작업 모니터를 통해 JES 스펴에 액세스할 수 있도록 허용합니다. 서버는 기본 액세스 제한사항을 제공하며 이 제한사항은 보안 제품의 표준 스펴 파일 보호 기능으로 확장될 수 있습니다. 스펴 파일에 대한 운영자 조치(유지, 릴리스, 취소, 영구 제거)는 조건부 허가를 설정해야 하는 EMCS 콘솔을 통해 수행됩니다.

작업에 대한 조치 - 대상 제한사항

JES 작업 모니터는 Developer for System z 사용자에게 JES 스펴에 대한 전체 운영자 액세스를 제공하지 않습니다. 기본적으로 사용자가 소유한 스펴 파일에 대해서만 보류, 해제, 취소, 제거 명령만 사용할 수 있습니다. 클라이언트 메뉴 구조에서 해당 옵션을 선택하면 이러한 명령이 실행됩니다(명령 프롬프트 없음) 보안 프로파일을 사용하여 명령을 사용할 수 있는 작업을 정의하여 명령 범위를 확장할 수 있습니다.

SDSF SJ 조치 문자와 유사하게 JES 작업 모니터도 선택된 작업 출력을 작성한 JCL을 검색하고 편집기 창에 표시하도록 JCL 표시 명령을 지원합니다. JES 작업 모니터는 JES에서 JCL을 검색하며, 이는 원래 JCL 멤버를 쉽게 찾을 수 없는 경우에 유용한 기능입니다.

표 1. JES 작업 모니터 콘솔 명령

조치	JES2	JES3
보류	\$Hx(jobid) x = {J, S 또는 T}	*F,J=jobid,H
해제	\$Ax(jobid) x = {J, S 또는 T}	*F,J=jobid,R
취소	\$Cx(jobid) x = {J, S 또는 T}	*F,J=jobid,C
제거	\$Cx(jobid),P x = {J, S 또는 T}	*F,J=jobid,C
JCL 표시	not applicable	not applicable

표 1에 나열된 사용 가능한 JES 명령은 기본적으로 사용자가 소유한 작업으로 제한됩니다. *Host Configuration Guide* (SC23-7658)의 "FEJJCNFG, JES 작업 모니터 구성 파일"에 설명된 대로 LIMIT_COMMANDS 지시문을 사용하여 이를 변경할 수 있습니다.

표 2. LIMIT_COMMANDS 명령 권한 매트릭스

	작업 소유자	
LIMIT_COMMANDS	사용자	기타
USERID(기본값)	허용	허용되지 않음

표 2. *LIMIT_COMMANDS* 명령 권한 매트릭스 (계속)

	작업 소유자	
LIMITED	허용	보안 프로파일에 명시적으로 허용되는 경우에만 허용
NOLIMIT	허용	보안 프로파일을 통해 또는 JESSPOOL 클래스가 활성화되지 않은 경우 허용

JES는 JESSPOOL 클래스를 사용하여 SYSIN/SYSOUT 데이터 세트를 보호합니다. SDSF와 유사하게 JES 작업 모니터는 JESSPOOL 클래스 사용을 확장하여 작업 자원도 보호합니다.

LIMIT_COMMANDS가 USERID가 아닌 경우, 다음 표에 표시된 대로 JES 작업 모니터는 JESSPOOL 클래스의 관련 프로파일에 대한 권한을 조회합니다.

표 3. 확장 JESSPOOL 프로파일

명령	JESSPOOL 프로파일	필수 액세스 권한
보류	nodeid.userid.jobname.jobid	ALTER
해제	nodeid.userid.jobname.jobid	ALTER
취소	nodeid.userid.jobname.jobid	ALTER
제거	nodeid.userid.jobname.jobid	ALTER
JCL 표시	nodeid.userid.jobname.jobid.JCL	READ

이전 표에서 다음을 대체하여 사용하십시오.

nodeid	대상 JES 서브시스템의 NJE 노드 ID
userid	작업 소유자의 로컬 사용자 ID
jobname	작업 이름
jobid	JES 작업 ID

JESSPOOL 클래스가 활성화되지 않은 경우, *Host Configuration Guide* (SC23-7658)의 "FEJJCNFG, JES 작업 모니터 구성 파일"의 "LIMIT_COMMANDS 명령 권한 매트릭스 테이블"에 설명된 대로 LIMIT_COMMANDS의 LIMITED 및 NOLIMIT 값이 다르게 동작합니다. 프로파일이 정의되어 있지 않으면 클래스가 기본적으로 권한을 거부하기 때문에 이 동작은 JESSPOOL가 활성화된 경우 동일합니다.

작업에 대한 조치 - 실행 제한사항

허용 대상을 지정한 후 JES 스푼 명령 보안의 두 번째 단계는 운영자 명령을 실제로 실행하는 데 필요한 허용을 포함합니다. 이 실행 권한은 z/OS 및 JES 보안 검사를 통해 강제 실행됩니다.

JCL 표시는 기타 JES 작업 모니터 명령(보류, 해제, 취소, 제거)과 같은 운영자 명령이 아니므로 추가 보안 검사가 없기 때문에 다음 목록에 있는 제한사항이 적용되지 않습니다.

JES 작업 모니터는 *Host Configuration Guide* (SC23-7658)의 "FEJJC�FG, JES 작업 모니터 구성 파일"에 설명된 대로 `CONSOLE_NAME` 지시문을 사용하여 이름이 제어되는 확장 MCS(EMCS) 콘솔을 통해 사용자가 요청한 모든 JES 운영자 명령을 실행합니다.

JES 작업 모니터를 통해 *Host Configuration Guide*(SC23-7658)의 "FEJJC�FG, JES 작업 모니터 구성 파일"에 설명된 대로 `LIMIT_CONSOLE` 지시문을 사용하여 EMCS 콘솔에 부여되는 권한의 양을 정의할 수 있습니다.

표 4. `LIMIT_CONSOLE` 콘솔 권한 매트릭스

<code>LIMIT_CONSOLE</code>	<code>OPERCMDS</code> 클래스의 활성 프로파일	<code>OPERCMDS</code> 클래스의 비활성 프로파일
LIMITED(기본값)	허용, 보안 프로파일이 허용하는 경우	허용되지 않음
NOLIMIT	허용, 보안 프로파일이 허용하는 경우	허용

이 설정을 통해 보안 관리자는 `OPERCMDS` 및 `CONSOLE` 클래스를 사용하여 세부 단위의 명령 실행 허용을 정의할 수 있습니다.

- EMCS 콘솔을 사용하려면 사용자에게 `OPERCMDS` 클래스의 `MVS.MCSOPER.console-name` 프로파일에 대한 (최소한) `READ` 권한이 있어야 합니다. 프로파일이 정의되어 있지 않으면 시스템이 권한 요청을 부여합니다.
- JES 운영자 명령을 실행하려면 사용자에게 `OPERCMDS` 클래스의 `JES%.**`(또는 보다 특정한) 프로파일에 대한 충분한 권한이 있어야 합니다. 프로파일이 정의되어 있지 않거나 `OPERCMDS` 클래스가 활성화되어 있지 않으면 `LIMIT_CONSOLE=LIMITED`가 `FEJJC�FG`에 정의된 경우 JES가 명령에 실패합니다.
- 또한 보안 관리자는 **PERMIT** 정의에 `WHEN(CONSOLE(JMON))`을 지정하여 운영자 명령을 실행할 때 사용자가 반드시 JES 작업 모니터를 사용하도록 요구할 수 있습니다. 이 설정이 작동하려면 `CONSOLE` 클래스가 활성화되어 있어야 합니다. `CONSOLE` 클래스가 활성화되어 있으면 충분합니다. EMCS 콘솔의 경우 프로파일을 검사하지 않습니다.

TSO 세션에서 `JMON` 콘솔을 작성하여 JES 작업 모니터 서버의 ID를 가정하는 것은 보안 소프트웨어에서 금지됩니다. 콘솔을 작성할 수 있지만 진입점이 다릅니다(JES 작업 모니터 대 TSO). 이 책에 설명된 대로 보안이 설정되고 사용자에게 다른 방법을 통해 JES 명령에 대한 권한이 없으면 이 콘솔에서 실행된 JES 명령은 보안 검사에 실패합니다.

콘솔 이름이 이미 사용 중이면 명령이 실행되어야 할 때 JES 작업 모니터가 콘솔을 작성할 수 없습니다. 이를 방지하기 위해 시스템 프로그래머는 JES 작업 모니터 구성 파일에 GEN_CONSOLE_NAME=ON 지시문을 설정하거나 보안 관리자는 TSO 사용자가 콘솔을 작성하지 못하도록 보안 프로파일을 정의할 수 있습니다. 다음 샘플 RACF 명령은 허용된 사용자를 제외한 모든 사람이 TSO 또는 SDSF 콘솔을 작성하지 못하게 합니다.

- 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)

참고: 이러한 운영자 명령에 대한 권한이 없더라도 사용자는 이러한 자원(JESINPUT, JESJOBS, JESSPOOL 클래스의 자원)을 보호하는 가능한 프로파일에 대한 충분한 권한이 있으면 JES 작업 모니터를 통해 여전히 작업을 제출하고 작업 출력을 읽을 수 있습니다.

운영자 명령 보호에 대한 자세한 정보는 *Security Server RACF Security Administrator's Guide*(SA22-7683)를 참조하십시오.

스플 파일 액세스

JES 작업 모니터는 기본적으로 모든 스플 파일에 대한 찾아보기 액세스를 허용합니다. *Host Configuration Guide* (SC23-7658)의 "FEJCNFG, JES 작업 모니터 구성 파일"에 설명된 대로 LIMIT_VIEW 지시문을 사용하여 이를 변경할 수 있습니다.

표 5. LIMIT_VIEW 찾아보기 권한 매트릭스

LIMIT_VIEW	작업 소유자	
	사용자	기타
USERID	허용	허용되지 않음
NOLIMIT(기본값)	허용	보안 프로파일을 통해 또는 JESSPOOL 클래스가 활성화되지 않은 경우 허용

사용자를 JES 스플의 자체 작업으로 제한하려면 JES 작업 모니터 구성 파일인 FEJCNFG에 "LIMIT_VIEW=USERID" 문을 정의하십시오. 사용자가 광범위한 작업에 액세스해야 하지만 모든 작업에 액세스하지는 않아도 되는 경우, 보안 제품의 표준 스플 파일 보호 기능(예: JESSPOOL 클래스)을 사용하십시오.

추가 보호를 정의하는 경우, JES 작업 모니터는 SAPI(SYSOUT 애플리케이션 프로그램 인터페이스)를 사용하여 스플에 액세스함을 기억하십시오. 이는 사용자에게 최소한 스플 파일에 대한 UPDATE 액세스 권한이 있어야 함을 의미하며 찾아보기 기능의 경우

에도 마찬가지입니다. z/OS 1.7(JES3의 경우 z/OS 1.8) 이상을 실행하는 경우에는 이 필수 조건이 적용되지 않습니다. 여기서는 찾아보기 기능에 READ 권한이면 충분합니다.

JES 스펴 파일 보호에 대한 자세한 정보는 *Security Server RACF Security Administrator's Guide*(SA22-7683)를 참조하십시오.

SSL/TLS 암호화된 통신

RSE를 사용하는 외부(클라이언트-호스트) 통신은 SSL(Secure Socket Layer) 또는 TLS(Transport Layer Security)를 사용하여 암호화할 수 있습니다. 이 기능은 기본적으로 사용하지 않으며 `ssl.properties`의 설정으로 제어됩니다. *Host Configuration Guide* (SC23-7658)의 "(선택사항) ssl.properties, RSE SSL 암호화"를 참조하십시오.

RSE 디먼과 RSE 서버는 상호 간의 아키텍처 차이로 인해 다른 인증서 저장 메커니즘을 지원합니다. 이는 RSE 디먼과 RSE 서버에 모두 SSL 정의와 인증서가 필요함을 의미합니다. RSE 디먼과 RSE 서버가 동일한 인증서 관리 방법을 사용하는 경우에는 공유 인증서를 사용할 수 있습니다.

표 6. SSL 인증서 스토리지 메커니즘

인증서 스토리지	작성자 및 관리자	RSE 디먼	RSE 서버
키 링	SAF 준수 보안 제품	지원됨	지원됨
키 데이터베이스	z/OS UNIX gskkyman	지원됨	/
키 저장소	Java의 keytool	/	지원됨

참고: SAF 준수 키 링은 인증서 관리를 위해 선호하는 방법입니다.

SAF 준수 키 링은 인증서의 개인 키를 보안 데이터베이스에 저장하거나 System z 암호 하드웨어의 인터페이스인 ICSF(Integrated Cryptographic Service Facility)를 사용하여 저장할 수 있습니다.

ICSF는 비ICSF 개인 키 관리보다 안전한 솔루션이므로 디지털 인증서와 연관된 개인 키를 저장하는 데 권장됩니다. ICSF를 사용하면 ICSF 마스터 키로 개인 키를 암호화하고 CSFKEYS 및 CSFSERV 보안 클래스의 일반 자원으로 개인 키에 대한 액세스를 제어합니다. 또한 ICSF는 암호 코프로세서 하드웨어를 이용하므로 작동 성능이 향상됩니다. ICSF와 암호 키 및 서비스를 사용할 수 있는 사용자를 제어하는 방법에 대한 세부사항은 *Cryptographic Services ICSF Administrator's Guide*(SA22-7521)를 참조하십시오.

RSE 디먼은 시스템 SSL 기능을 사용하여 SSL 암호화된 통신을 관리합니다. 이는 SYS1.SIEALNKE는 보안 소프트웨어로 제어하는 프로그래머가 하며 `rsed.envvars`의 STEPLIB 지시문 또는 LINKLIST를 통해 RSE가 사용할 수 있어야 함을 의미합니다.

SAF 준수 키 링을 RSE 디먼 또는 RSE 서버에 사용하는 경우 RSE 사용자 ID(다음 샘플 명령의 경우 stcrse)는 키 링과 관련 인증서에 액세스할 수 있는 권한이 필요합니다.

- 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

rsed.envvars의 _RSE_JAVAOPTS 지시문에서 DSTORE_SSL_ALGORITHM 변수를 사용하면 암호화 메소드로서 SSL과 해당 계승자 TLS 중에 선택할 수 있습니다. 이 내용은 *Host Configuration Guide*(SC23-7658)에 있는 "_RSE_JAVAOPTS로 기타 Java 시작 매개변수 정의"에서 설명합니다.

Developer for System z에 SSL 활성화에 대한 세부사항은 211 페이지의 제 13 장 『SSL 및 X.509 인증 설정』을 참조하십시오.

통합 디버거 암호화 통신

선택적 디버거 관리자를 사용한 외부(클라이언트-호스트) 통신은 SSL 또는 TLS를 사용하여 암호화될 수 있습니다. 이러한 방법으로 암호화를 수행하려면 외부 통신을 위해 디버거 관리자가 사용하는 포트(기본값 5335)의 AT-TLS 정책을 작성하십시오. 샘플 정책은 35 페이지의 그림 9에 제공됩니다. AT-TLS(Application Transparent TLS) 설정에 대한 자세한 내용은 *Communications Server IP Configuration Guide*(SC31-8775)를 참조하십시오.

```

| 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
| {
|   TLSEnabled                          On
|   Trace                              2
| }

```

그림 9. 디버그 관리자용 AT-TLS 정책

X.509 인증서를 사용한 클라이언트 인증

RSE 디먼은 사용자가 X.509 인증서를 사용하여 스스로를 인증하는 것을 지원합니다. SSL 암호화된 통신은 SSL에서 사용되는 인증서를 사용하여 호스트 인증을 확장한 것이기 때문에 이 기능이 작동하려면 SSL 암호화된 통신을 사용하는 것이 전제조건입니다.

RSE 디먼은 클라이언트 유효성 인증서의 유효성을 검증하여 클라이언트 인증 프로세스를 시작합니다. 검사되는 몇 가지 주요 측면은 인증서의 유효 날짜와 인증서에 서명하는 데 사용되는 인증 기관(CA)의 신뢰도입니다. 선택적으로 (짜드파티) 인증서 폐기 목록(CRL)도 참고할 수 있습니다.

RSE 디먼이 인증서의 유효성을 검증한 후 인증을 위해 인증서가 처리됩니다. rsed.envvars 지시문 enable.certificate.mapping이 false로 설정(이 시점에서 RSE 디먼이 인증을 수행함)되어 있지 않으면 인증을 위해 인증서가 보안 제품에 전달됩니다.

성공하면, 인증 프로세스가 이 세션에 사용될 사용자 ID를 결정합니다. RSE 디먼은 이 사용자 ID를 테스트하여 RSE 디먼이 실행 중인 호스트 시스템에서 사용할 수 있는지 확인합니다.

마지막 검사(X.509 인증서뿐만 아니라 모든 인증 메커니즘에 대해 수행됨)는 사용자 ID가 Developer for System z를 사용할 수 있는지 확인합니다.

TCP/IP가 사용하는 SSL 보안 분류에 대해 잘 알고 있는 경우, 이러한 유효성 검증 단계를 조합하면 “레벨 3 클라이언트 인증” 스펙(사용 가능한 최상위 레벨)과 일치합니다.

인증 기관(CA) 유효성 검증

인증서 유효성 검증 프로세스의 일부에는 사용자가 신뢰하는 인증 기관(CA)에서 인증서에 서명했는지를 검사하는 과정이 포함됩니다. 이를 수행하려면 RSE 디먼에 CA를 식별하는 인증서에 대한 액세스 권한이 있어야 합니다.

SSL 연결에 **gskkyman** 키 데이터베이스를 사용하는 경우, 키 데이터베이스에 CA 인증서를 추가해야 합니다.

SAF 키 링을 사용하는 경우(권장 방법임), 다음 샘플 RACF 명령에 표시된 대로 TRUST 또는 HIGHTRUST 속성을 가진 CERTAUTH 인증서로 보안 데이터베이스에 CA 인증서를 추가해야 합니다.

- RACDCERT CERTAUTH ADD(dsn) HIGHTRUST WITHLABEL('label')

대부분의 보안 제품에는 데이터베이스에서 사용 가능한 NOTRUST 상태의 잘 알려진 CA에 대한 인증서가 이미 있습니다. 다음 샘플 RACF 명령을 사용하여 기존 CA 인증서를 나열하고 지정된 레이블을 기반으로 신뢰할 수 있는 인증서로 표시하십시오.

- RACDCERT CERTAUTH LIST
- RACDCERT CERTAUTH ALTER(LABEL('HighTrust CA')) HIGHTRUST

참고: 인증서의 HostIdMappings 확장을 기반으로 사용자를 인증하는 RACF에 의존하는 경우 HIGHTRUST 상태가 필요합니다. 자세한 정보는 37 페이지의 『보안 소프트웨어를 사용한 인증』을 참조하십시오.

보안 데이터베이스에 CA 인증서가 추가되면 다음 샘플 RACF 명령에 표시된 대로 RSE 키 링에 연결되어야 합니다.

- RACDCERT ID(stcrse) CONNECT(CERTAUTH LABEL('HighTrust CA')
RING(rdzssl.racf))

RACDCERT 명령에 대한 자세한 정보는 *Security Server RACF Command Language Reference*(SA22-7687)를 참조하십시오.

주의: 보안 소프트웨어 대신 RSE 디먼을 사용하여 사용자를 인증하는 경우, SAF 키 링 또는 **gskkyman** 키 데이터베이스에서 TRUST 및 HIGHTRUST 상태의 CA를 혼합하지 않도록 주의해야 합니다. RSE 디먼은 두 상태를 구별할 수 없으므로 TRUST 상태의 CA에서 서명한 인증서는 사용자 ID 인증용으로 올바릅니다.

(선택사항) 인증서 폐기 목록(CRL) 조회

원하는 경우 RSE 디먼에 하나 이상의 인증서 폐기 목록(CRL)을 선택하여 유효성 검증 프로세스에 추가 보안을 추가하도록 지시할 수 있습니다. 이를 수행하려면 rsed.envvars에 CRL 관련 환경 변수를 추가해야 합니다.

- GSK_CRL_SECURITY_LEVEL
- GSK_LDAP_SERVER
- GSK_LDAP_PORT
- GSK_LDAP_USER
- GSK_LDAP_PASSWORD

이들 변수와 z/OS 시스템 SSL이 사용하는 기타 환경 변수에 대한 자세한 정보는 *Cryptographic Services System Secure Sockets Layer Programming(SC24-5901)*을 참조하십시오.

참고: rsed.envvars에 다른 z/OS 시스템 SSL 환경 변수(GSK_*)를 지정하면 RSE 디먼이 SSL 연결과 인증서 인증을 처리하는 방식이 변경될 수 있으므로 유의해야 합니다.

보안 소프트웨어를 사용한 인증

RACF는 인증서를 인증하고 연관된 사용자 ID를 리턴하기 위해 몇 가지 검사를 수행합니다. 다른 보안 제품에서는 다르게 수행될 수 있습니다. 인증을 수행(조회 모드)하는 데 사용되는 initACEE 함수에 대한 자세한 정보는 보안 제품 문서를 참조하십시오.

1. RACF는 인증서가 DIGTCERT 클래스에 정의되어 있는지 여부를 확인합니다. 정의되어 있는 경우, RACF는 인증서가 RACF 데이터베이스에 추가될 때 이 인증서와 연관된 사용자 ID를 리턴합니다.

다음 예제에서와 같이 인증서는 RACDCERT 명령을 사용하여 RACF에 정의됩니다.

```
RACDCERT ID(userid) ADD(dsn) TRUST WITHLABEL('label')
```

2. 인증서가 정의되어 있지 않은 경우, RACF는 DIGTNMAP 또는 DIGTCRIT 클래스에 정의된 일치하는 인증서 이름 필터가 있는지 여부를 확인합니다. 있는 경우, 가장 명확하게 일치하는 필터와 연관된 사용자 ID를 리턴합니다.

참고: 이러한 필터는 모든 인증서를 단일 사용자 ID에 맵핑하므로 Developer for System z에서 사용하는 인증서에 이름 필터를 사용하지 않는 것이 좋습니다. 그 결과 모든 Developer for System z 사용자가 동일한 사용자 ID를 사용하여 로그인합니다.

3. 일치하는 이름 필터가 없는 경우, RACF는 HostIdMappings 인증서 확장을 찾고 임베디드 사용자 ID와 호스트 이름 쌍을 추출합니다. 확장을 찾아 유효성을 검증한 경우, RACF는 HostIdMappings 확장 내에 정의된 사용자 ID를 리턴합니다.

다음 조건이 모두 참이면 사용자 ID와 호스트 이름 쌍이 올바릅니다.

- 이 인증서에 서명하는 데 사용되는 CA 인증서는 DIGTCERT 클래스에 HIGHTRUST로 표시됩니다.
- 확장에 저장된 사용자 ID의 길이가 올바릅니다(1 - 8자).
- RSE 디먼에 지정된 사용자 ID에 SERVAUTH 클래스의 IRR.HOST.hostname 프로파일에 대한 (최소한) READ 권한이 있습니다(여기서 hostname은 확장에 저장된 호스트 이름임). 일반적으로 이는 CDFMVS08.RALEIGH.IBM.COM과 같은 도메인 이름입니다.

ASN.1 구문에서 HostIdMappings 확장 정의는 다음과 같습니다.

```
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
}
```

참고: HostIdMappings 확장이 포함된 인증서의 유효 기간이 시작된 후에 대상 사용자 ID가 작성된 경우 HostIdMappings 확장이 적용되지 않습니다. 따라서 특히 HostIdMappings 확장이 포함된 인증서의 사용자 ID를 작성하는 경우 인증서 요청을 제출하기 전에 사용자 ID를 작성하십시오.

X.509 인증서, RACF가 이 인증서를 관리하는 방법, 인증서 이름 필터 정의 방법에 대한 자세한 정보는 *Security Server RACF Security Administrator's Guide*(SA22-7683)를 참조하십시오. **RACDCERT** 명령에 대한 자세한 정보는 *Security Server RACF Command Language Reference*(SA22-7687)를 참조하십시오.

RSE 디먼을 사용한 인증

Developer for System z는 보안 제품에 의존하지 않고 기본 X.509 인증서 인증을 수행할 수 있습니다. RSE 디먼이 수행하는 인증은 인증서 확장에 사용자 ID와 호스트 이름이 정의되어 있어야 하며 rsed.envvars의 enable.certificate.mapping 지시문이 FALSE로 설정된 경우에만 활성화됩니다.

이 기능은 보안 제품이 X.509 인증서를 기반으로 한 사용자 인증을 지원하지 않거나 인증서가 보안 제품이 수행하는 테스트에 실패할 경우(예를 들어, 인증서의 HostIdMappings 확장 ID가 잘못되었거나 DIGTCERT에 이름 필터 또는 정의가 없음) 사용됩니다.

클라이언트는 사용자에게 사용할 확장 ID(OID)(기본적으로 HostIdMappings OID {1 3 18 0 2 18 1}임)를 조회합니다.

RSE 디먼은 HostIdMappings 확장 형식을 사용하여 사용자 ID와 호스트 이름을 추출합니다. 이 형식은 37 페이지의 『보안 소프트웨어를 사용한 인증』에 설명되어 있습니다.

다음 조건이 모두 참이면 사용자 ID와 호스트 이름 쌍이 올바릅니다.

- 확장에 저장된 사용자 ID의 길이가 올바릅니다(1 - 8자).
- RSE 디먼에 지정된 사용자 ID에 SERVAUTH 클래스의 IRR.HOST.hostname 프로파일에 대한 (최소한) READ 권한이 있습니다(여기서 hostname은 확장에 저장된 호스트 이름임). 일반적으로 이는 CDFMVS08.RALEIGH.IBM.COM과 같은 도메인 이름입니다.

경고: RSE 디먼은 클라이언트 인증서에 서명한 CA가 신뢰성이 높는지 또는 단지 신뢰할 수 있는 정도인지 여부를 확인할 수 없기 때문에 RSE 디먼에 알려진 모든 CA가 신뢰성이 높는지 확인하는 것은 보안 관리자의 책임입니다. 액세스 가능한 CA 인증서에 대한 자세한 정보는 36 페이지의 『인증 기관(CA) 유효성 검증』을 참조하십시오.

POE(Port Of Entry) 확인

Developer for System z는 신뢰 TCP/IP 주소에 대한 호스트 액세스만 허용하는 POE(Port Of Entry) 확인을 지원합니다. 이 기능은 기본적으로 사용하지 않으며 BPX.POE 보안 프로파일의 정의가 필요합니다(다음 샘플 RACF 명령 참조).

- RDEFINE FACILITY BPX.POE UACC(NONE)
- PERMIT BPX.POE CLASS(FACILITY) ACCESS(READ) ID(STCRSE)
- SETROPTS RACLIST(FACILITY) REFRESH

참고:

- rsed.envvars에서 "enable.port.of.entry=true" 옵션의 주석을 해제하여 RSE 에서 POE를 사용하도록 구성해야 합니다(*Host Configuration Guide* (SC23-7658)의 "_RSE_JVAOPTS를 사용하여 추가 Java 시작 매개변수 정의" 설명 참조).
- 이 프로파일이 정의되지 않고 rsed.envvars에서 PEO 확인을 사용하는 경우 RSE 사용자 ID STCRSE에는 UID(0)가 필요합니다.

- BPX.POE를 정의하면 POE 확인을 지원하는 다른 TC/PIP 애플리케이션(예: INETD)에 영향을 줍니다.
- POE 확인 기능을 모두 이용하려면 보안 영역(IP 주소 범위인 EZB.NETACCESS.** 프로파일)을 SERVAUTH 클래스에서 설정해야 합니다.

POE 확인을 사용한 네트워크 액세스 제어에 대한 자세한 정보는 *Communications Server IP Configuration Guide(SC31-8775)*를 참조하십시오.

클라이언트 기능 변경

Developer for System z 클라이언트 버전 8.5.1 이상은 SAF 보안 프로파일에 대한 액세스 권한을 확인할 수 있으며 그 결과를 기반으로 사용자에게 관련 기능을 사용할 수 있게 하거나 사용할 수 없게 할 수 있습니다.

Developer for System z는 표 7에 나열된 프로파일에 대한 액세스 허용을 확인하여 사용자에게 사용할 수 있게 하거나 사용할 수 없게 해야 하는 옵션을 결정합니다.

표 7. 클라이언트 기능 변경에 대한 SAF 정보

FACILITY 프로파일	고정 길이	필수 액세스 권한	결과
FEK.USR.OFF.REMOTECOPY.MVS.sysname	27	READ	클라이언트는 MVS 데이터 세트에 대해 복사 및 관련 기능을 사용할 수 없게 합니다.

참고: Developer for System z는 보안 소프트웨어가 사용자가 프로파일에 대한 액세스 권한을 갖고 있는지 여부를 결정할 수 없음을 표시하는 경우 사용자에게 액세스 권한이 없다고 가정합니다. 예를 들어, 프로파일이 정의되지 않는 경우입니다.

sysname 값은 대상 시스템의 시스템 이름을 비교합니다.

"고정 길이" 열은 관련 보안 프로파일의 고정 파트 길이를 설명합니다.

기본적으로 Developer for System z는 FEK.* 프로파일이 FACILITY 보안 클래스에 있다고 예상합니다. FACILITY 클래스의 프로파일은 39자로 제한됩니다. 고정 프로파일 파트(FEK.USR.<key>)의 길이와 사이트 특정 프로파일 파트(sysname)의 길이 합계가 이 숫자를 초과하는 경우, 다른 클래스에 프로파일을 배치하고 이 클래스를 대신 사용하도록 Developer for System z에 지시할 수 있습니다. 이를 수행하려면 rsed.envvars에서 _RSE_FEK_SAF_CLASS의 주석을 해제하고 원하는 클래스 이름을 제공합니다.

다음 샘플 보안 정의는 RESTRICT 그룹의 사용자를 제외하고 모든 사용자에게 CDFMVS08에 대한 REMOTECOPY.MVS 조치를 허용합니다.

```
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

사용자에게 FEK.USR.OFF.REMOTECOPY.MVS.sysname 프로파일에 대한 READ 액세스 권한이 있는 경우, Developer for System z 클라이언트 버전 8.5.1 이상은 MVS 데이터 세트에 대해 끌기, 복사, 다른 이름으로 저장, 오프라인 작업 조치를 사용할 수 없습니다. 그 결과 사용자는 이 시스템의 데이터 세트에 액세스할 수 있지만 워크스테이션에 데이터 세트 로컬 사본을 작성할 수 없습니다. 따라서 로컬 워크스테이션이 유실되거나 도난당할 경우 기밀 정보 노출을 방지합니다.

클라이언트로 푸시 개발자 그룹

Developer for System z 클라이언트 버전 8.0.1 이상은 연결 시 호스트에서 클라이언트 구성 파일과 업그레이드 정보를 가져올 수 있으므로 모든 클라이언트가 공통 설정을 갖고 최신 상태를 유지합니다.

버전 8.0.3부터는 클라이언트 관리자가 다양한 개발자 그룹의 요구에 맞는 여러 클라이언트 구성 세트와 여러 클라이언트 업데이트 시나리오를 작성할 수 있습니다. 따라서 사용자가 LDAP 그룹 멤버십 또는 보안 프로파일 허가와 같은 기준에 따라 사용자 정의 설정을 수신할 수 있습니다.

보안 데이터베이스의 정의를 선택 메커니즘으로 사용하는 경우(SAF 값이 pushtoclient.properties의 지시문에 대해 지정됨), Developer for System z는 표 8에 나열된 프로파일에 대한 액세스 허가를 확인하여 사용자가 속하는 개발자 그룹과 사용자가 업데이트를 거부할 수 있는지 여부를 결정합니다.

표 8. 클라이언트로 푸시 SAF 정보

FACILITY 프로파일	고정 길이	필수 액세스 권한	결과
FEK.PTC.CONFIG.ENABLED. sysname.devgroup	23	READ	클라이언트가 지정된 그룹에 대한 구성 업데이트를 허용합니다.
FEK.PTC.PRODUCT. ENABLED.sysname.devgroup	24	READ	클라이언트가 지정된 그룹에 대한 제품 업데이트를 허용합니다.
FEK.PTC.REJECT.CONFIG. UPDATES.sysname	30	READ	사용자가 구성 업데이트를 거부할 수 있습니다.
FEK.PTC.REJECT.PRODUCT. UPDATES.sysname	31	READ	사용자가 제품 업데이트를 거부할 수 있습니다.

참고: Developer for System z에서는 보안 소프트웨어가 사용자에게 프로파일에 대한 액세스 권한이 있는지 여부를 결정할 수 없음을 나타내는 경우 사용자에게 액세스 권한이 없는 것으로 가정합니다. 예를 들어, 프로파일이 정의되지 않는 경우입니다.

devgroup 값은 특정 개발자 그룹에 지정된 그룹 이름을 비교합니다. 그룹 이름은 Developer for System z 클라이언트에서 표시됩니다.

sysname 값은 대상 시스템의 시스템 이름을 비교합니다.

"Fixed length" 열에는 관련 보안 프로파일의 고정 파트 길이가 기록됩니다.

기본적으로 Developer for System z는 FEK.* 프로파일이 FACILITY 보안 클래스에 있다고 예상합니다. FACILITY 클래스의 프로파일은 39자로 제한됩니다. 고정 프로파일 파트 (FEK.PTC.<key>)의 길이와 사이트별 프로파일 파트(sysname 또는 sysname.devgroup)의 길이 합계가 이 숫자를 초과하는 경우에는 프로파일을 다른 클래스에 배치하고 Developer for System z이 대신 이 클래스를 사용하도록 지시할 수 있습니다. 이를 수행하려면 rsed.envvars에서 _RSE_FEK_SAF_CLASS의 주석을 해제하고 원하는 클래스 이름을 제공합니다.

클라이언트 관리자가 관련 클라이언트로 푸시 메타데이터를 정의하고 관리하려면 FEK.PTC.*.ENABLED.* 프로파일의 액세스 목록에 있어야 합니다. 이는 그룹 지원이 가능한 클라이언트로 푸시를 구현하려면 최소한 액세스 목록의 클라이언트 관리자로 프로파일을 정의해야 함을 의미합니다.

여러 그룹 지원 사용에 대한 자세한 정보는 *Host Configuration Guide*(SC23-7658)의 "(선택사항) pushtoclient.properties, 호스트 기반 클라이언트 제어"를 참조하십시오. 클라이언트로 푸시 개념 및 구현에 대한 자세한 정보는 135 페이지의 제 7 장 『클라이언트로 푸시 고려사항』을 참조하십시오.

디버그 보안

선택적인 통합 디버거는 읽기 전용 메모리로 로드되는 CICS 트랜잭션을 디버깅할 수 있습니다. 문제점 상태(권한 없음)에서 트랜잭션을 디버깅하는 데 사용되는 경우, 통합 디버거는 디버그 세션을 소유하는 사용자에게 이를 수행하도록 허용되는지를 검사합니다.

Developer for System z는 표 9에 나열된 프로파일에 대한 액세스를 확인하여 사용할 수 있도록 권한이 부여되는 디버그를 판별합니다.

표 9. 디버그 기능에 대한 SAF 정보

FACILITY 프로파일	필수 액세스 권한	결과
AQE.AUTHDEBUG.WRITEBUFFER	UPDATE	사용자는 읽기 전용 CICS 트랜잭션을 디버깅할 수 있습니다.

참고: Developer for System z는 보안 소프트웨어가 사용자가 프로파일에 대한 액세스 권한을 갖고 있는지 여부를 결정할 수 없음을 표시하는 경우 사용자에게 액세스 권한이 없다고 가정합니다. 예를 들어, 프로파일이 정의되지 않는 경우입니다.

다음 샘플 보안 정의에서는 RDZDEBUG 그룹에 있는 모든 사용자에게 대해 AUTHDEBUG.WRITEBUFFER 조치가 허용됩니다.

```
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 보안

선택적인 통합 디버거는 CICS 트랜잭션을 디버깅할 수 있습니다. 자세한 내용은 167 페이지의 『CICS 트랜잭션 디버깅』을 참조하십시오.

Developer for System z는 애플리케이션 배치 관리자를 통해 CICS 관리자가 개발자가 편집할 수 있는 CICS 자원 정의, 해당 기본값, CICS 자원 정의(CRD) 서버를 통한 CICS 자원 정의 표시를 제어할 수 있게 합니다. 필수 CICS TS 보안 정의에 대한 자세한 정보는 155 페이지의 제 8 장 『CICSTS 고려사항』을 참조하십시오.

CRD 저장소

CRD 서버 저장소 VSAM 데이터 세트는 모든 기본 자원 정의를 보관하므로 업데이트로부터 보호해야 하지만 개발자는 여기에 저장된 값을 읽을 수 있어야 합니다.

CICS 트랜잭션

Developer for System z는 CICS 자원을 정의하고 조회할 때 CRD 서버가 사용하는 다중 트랜잭션을 제공합니다. 트랜잭션에 접속되면, CICS 자원 보안 검사(사용 가능한 경우)가 사용자 ID에 트랜잭션 ID를 실행할 수 있는 권한이 있는지 확인합니다.

SSL 암호화된 통신

애플리케이션 배치 관리자 클라이언트는 CICS TS 웹 서비스 또는 RESTful 인터페이스를 사용하여 CRD 서버를 호출합니다. 이 통신에 대한 SSL 사용은 CICS TS TCPIP SERVICE 정의로 제어됩니다(CICS TS에 대한 RACF 보안 참조서 참조).

기타 정보

GATE 트래싱

주소 공간에서 RACF에게 DATASET결과 같이 RACLIST화(메모리에 저장)되지 않은 자원 클래스에 액세스하도록 처음 지시하면, RACF는 GATE(Generic Anchor Table

Entry)로 알려진 목록에서 사용자 주소 공간에 있는 관련된 모든 일반 프로파일을 검색하고 저장합니다. z/OS 1.12까지, RACF는 각 주소 공간에 대해 네 개의 일반 앵커를 유지보수하고, 자체 ACEE가 있는 각 MVS TCB에 대해 네 개를 유지보수합니다. 네 개 모두 사용되는 경우 새 앵커가 들어오면 RACF는 가장 오래 전에 참조된 앵커를 대체합니다.

사용자가 5개 이상의 데이터 세트 고급 규정자에 자주 액세스하는 경우, RSE 스프레드폴(사용자 고유 ACEE와 함께 스프레드를 사용하여 여러 사용자를 서빙함)은 RACF가 사용 가능한 앵커 슬롯을 통해 새 항목을 회전시켜야 하므로 GATE 트래싱을 경험할 수 있습니다.

z/OS 1.12에서, RACF는 SET 명령의 **GENERICANCHOR** 옵션을 도입했으며, 이를 사용하면 사용자가 테이블의 크기를 늘릴 수 있습니다. 이것은 각 작업 이름에 대해 또는 시스템 범용으로 설정될 수 있습니다.

관리 ACEE

Developer for System z는 pthread_security_np(), __passwd()와 같이 InitACEE 보안 서비스를 사용하는 z/OS UNIX 커널 서비스를 사용하므로 "관리 ACEE" 보안 제어 블록을 생성합니다. 관리 ACEE(Accessor Environment Element)는 보안 제품으로 캐시되며 보안 제품은 캐시 제한시간이 초과될 때까지 특정 변경사항(예: Developer for System z 외부에서의 비밀번호 변경)을 무시합니다. 제한시간 초과는 몇 분이 소요될 수 있습니다.

보안 변경 후 관리 ACEE 캐시를 새로 고치면 Developer for System z에서 새 데이터를 사용할 수 있습니다.

SCLM 보안

SCLM 개발자 툴킷 서비스는 빌드, 승격, 배치 기능에 대한 선택적 보안 기능을 제공합니다.

SCLM 관리자가 기능에 대한 보안을 사용하는 경우, SAF 호출이 작성되어 호출자 또는 대리 사용자 ID로 보호된 기능을 실행할 수 있는 권한을 확인합니다.

필수 SCLM 보안 정의에 대한 자세한 정보는 *SCLM Developer Toolkit Administrator's Guide*(SC23-9801)를 참조하십시오.

Developer for System z 구성 파일

지시문이 보안 및 감사 설정에 영향을 주는 몇 개의 Developer for System z 구성 파일이 있습니다. 이 장의 정보를 기반으로 보안 관리자와 시스템 프로그래머는 다음 지시문에 대한 설정을 결정할 수 있습니다.

JES 작업 모니터 - FEJJCNFG

- `LIMIT_COMMANDS={USERID | LIMITED | NOLIMIT}`

수행할 수 있는 작업 조치를 정의합니다(찾아보기와 제출 제외). 자세한 정보는 29 페이지의 『작업에 대한 조치 - 대상 제한사항』을 참조하십시오.

- `LIMIT_CONSOLE={LIMITED | NOLIMIT}`

조치를 실행하는 데 사용되는 EMCS 콘솔의 권한 레벨을 정의합니다. 자세한 정보는 29 페이지의 『작업에 대한 조치 - 대상 제한사항』을 참조하십시오.

- `LIMIT_VIEW={USERID | NOLIMIT}`

찾아볼 수 있는 스폴 파일을 정의합니다. 자세한 정보는 32 페이지의 『스폴 파일 액세스』를 참조하십시오.

- `LOOPBACK_ONLY={ON | OFF}`

이 z/OS 시스템 외부에서 JES 작업 모니터에 액세스할 수 있는지 여부를 정의하십시오. 자세한 정보는 *Host Configuration Guide*(SC23-7658)의 기본 사용자 정의 장에 있는 *FEJJCNFG*, *JES 작업 모니터 구성 파일* 절을 참조하십시오.

- `APPLID={FEKAPPL | *}`

PassTicket 작성/유효성 검증에 사용되는 애플리케이션 ID. 자세한 정보는 25 페이지의 『PassTicket 사용』을 참조하십시오.

참고: 이 지시문과 기타 FEJJCNFG 지시문에 대한 세부사항은 *Host Configuration Guide* (SC23-7658)의 "FEJJCNFG, JES 작업 모니터 구성 파일"에 나와 있습니다.

RSE - rsed.envvars

- `_RSE_FEK_SAF_CLASS={FACILITY | *}`

FEK.** 프로파일을 보유하는 보안 클래스. 자세한 정보는 41 페이지의 『클라이언트로 푸시 개발자 그룹』 및 40 페이지의 『클라이언트 기능 변경』의 내용을 참조하십시오.

- `(_RSE_JAVAOPTS) -DDENY_PASSWORD_SAVE={true | false}`

사용자가 클라이언트에 해당 호스트 비밀번호를 저장하지 않도록 거부합니다. 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "_RSE_JAVAOPTS를 사용하여 추가 Java 시작 매개변수 정의"를 참조하십시오.

- `(_RSE_JAVAOPTS) -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=value`

유휴 클라이언트 연결을 끊기 위한 타이머. 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "_RSE_JAVAOPTS를 사용하여 추가 Java 시작 매개변수 정의"를 참조하십시오.

- (_RSE_JAVAOPTS) -DAPPLID={FEKAPPL | *}

PassTicket 작성/유효성 검증에 사용되는 애플리케이션 ID. 자세한 정보는 25 페이지의 『PassTicket 사용』을 참조하십시오.

- (_RSE_JAVAOPTS) -Denable.port.of.entry={true | false}

POE(Port Of Entry) 확인을 사용합니다. 자세한 정보는 39 페이지의 『POE(Port Of Entry) 확인』을 참조하십시오.

- (_RSE_JAVAOPTS) -DDSTORE_SSL_ALGORITHM={TLSv1.2 | SSL}

SSL 또는 TLS를 통신 암호화 방법으로 선택하십시오. 자세한 정보는 33 페이지의 『SSL/TLS 암호화된 통신』을 참조하십시오.

- (_RSE_JAVAOPTS) -Denable.certificate.mapping={true | false}

보안 제품을 사용하여 X.509 인증서로 사용자를 인증합니다. 자세한 정보는 35 페이지의 『X.509 인증서를 사용한 클라이언트 인증』을 참조하십시오.

- GSK_CRL_SECURITY_LEVEL={LOW | MEDIUM | HIGH}

GSK_LDAP_SERVER=*
GSK_LDAP_PORT={389 | *}
GSK_LDAP_USER=*
GSK_LDAP_PASSWORD=*

X.509 인증을 위한 추가 보안 검사. 자세한 정보는 37 페이지의 『(선택사항) 인증서 폐기 목록(CRL) 조회』를 참조하십시오.

- (_RSE_JAVAOPTS) -Ddaemon.log={/var/rdz/logs | *}

감사 로그 파일의 위치. 자세한 정보는 26 페이지의 『감사 로깅』을 참조하십시오.

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

감사 로그 파일의 파일 액세스 권한 마스크. 자세한 정보는 26 페이지의 『감사 로깅』을 참조하십시오.

- (_RSE_JAVAOPTS) -Daudit.action=<shell script>

(_RSE_JAVAOPTS) -Daudit.action.id=<userid>

감사 로그를 처리하는 z/OS UNIX 기반 사용자 액시트. 자세한 정보는 26 페이지의 『감사 로깅』을 참조하십시오.

참고: 이 지시문과 기타 rsed.envvars 지시문에 대한 세부사항은 *Host Configuration Guide* (SC23-7658)의 "rsed.envvars, RSE 구성 파일"에 나와 있습니다.

RSE - ssl.properties

- daemon_keydb_file={SAF key ring name | gskkyman key database name}

RSE 디먼 인증서 위치. 자세한 정보는 33 페이지의 『SSL/TLS 암호화된 통신』을 참조하십시오.

- `daemon_key_label=certificate label`

RSE 디먼 인증서 이름. 자세한 정보는 33 페이지의 『SSL/TLS 암호화된 통신』을 참조하십시오.

- `server_keystore_file={SAF key ring name | Java key store name}`

RSE 서버 인증서 위치. 자세한 정보는 33 페이지의 『SSL/TLS 암호화된 통신』을 참조하십시오.

- `server_keystore_label=certificate label`

RSE 서버 인증서의 이름. 자세한 정보는 33 페이지의 『SSL/TLS 암호화된 통신』을 참조하십시오.

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

사용하는 키 저장소 유형(Java 키 저장소 또는 SAF 키 링). 자세한 정보는 33 페이지의 『SSL/TLS 암호화된 통신』을 참조하십시오.

참고: 이 지시문과 기타 `ssl.properties` 지시문에 대한 세부사항은 *Host Configuration Guide* (SC23-7658)의 "(선택사항) `ssl.properties`, RSE SSL 암호화"에 나와 있습니다.

RSE - `pushtoclient.properties`

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

Developer for System z 클라이언트 구성 파일에 대한 호스트 기반 제어. 자세한 정보는 135 페이지의 제 7 장 『클라이언트로 푸시 고려사항』을 참조하십시오.

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

System z 클라이언트 제품 업데이트에 대한 호스트 기반 제어. 자세한 정보는 135 페이지의 제 7 장 『클라이언트로 푸시 고려사항』을 참조하십시오.

참고: 이 지시문과 기타 `pushtoclient.properties` 지시문에 대한 세부사항은 *Host Configuration Guide*(SC23-7658)의 "(선택사항) `pushtoclient.properties`, 호스트 기반 클라이언트 제어"에 나와 있습니다.

보안 정의

Developer for System z의 기본 보안 정의를 작성하는 샘플 RACF 및 z/OS UNIX 명령이 있는 샘플 FEKRACF 멤버를 사용자 정의하고 제출하십시오.

FEK.SFEKSAMP(FEKSETUP) 작업을 사용자 정의하고 제출할 때 다른 위치를 지정하지 않은 한, FEKRACF는 FEK.#CUST.JCL에 있습니다. 자세한 내용은 *IBM Rational Developer for System z Host Configuration Guide*의 "사용자 정의 설정"을 참조하십시오.

RACF 명령에 대한 자세한 정보는 *RACF Command Language Reference* (SA22-7687)를 참조하십시오.

참고:

- z/OS용 CA ACF2™을 사용하는 사이트의 경우 CA 지원 사이트(<https://support.ca.com>)의 제품 페이지를 참조하거나 관련 Developer for System z 지식 문서, TEC492389를 확인하십시오. 이 지식 문서에는 Developer for System z를 적절히 구성하는 데 필요한 보안 명령에 대한 세부사항이 들어 있습니다.
- z/OS용 CA Top Secret®을 사용하는 사이트의 경우 CA 지원 사이트(<https://support.ca.com>)의 제품 페이지를 참조하거나 관련 Developer for System z 지식 문서, TEC492091을 확인하십시오. 이 지식 문서에는 Developer for System z를 적절히 구성하는 데 필요한 보안 명령에 대한 세부사항이 들어 있습니다.

다음 절에서는 필수 단계, 선택적 구성 및 가능한 대안에 대해 설명합니다.

요구사항 및 체크리스트

보안 설정을 완료하려면 보안 관리자가 표 10에 나열된 값을 알아야 합니다. 이러한 값은 이전 Developer for System z 설치 및 사용자 정의 단계 중에 정의되었습니다.

표 10. 보안 설정 변수

설명	<ul style="list-style-type: none">• 기본값• 값을 찾을 수 있는 위치	값
Developer for System z 제품 상위 레벨 규정자	<ul style="list-style-type: none">• FEK• SMP/E 설치	
Developer for System z 사용자 정의 상위 레벨 규정자	<ul style="list-style-type: none">• FEK.#CUST• FEK.SFEKSAMP (FEKSETUP)(<i>IBM Rational Developer for System z Host Configuration Guide</i>의 "사용자 정의 설정"에 설명됨)	

표 10. 보안 설정 변수 (계속)

설명	<ul style="list-style-type: none"> 기본값 답을 찾을 수 있는 위치 	값
통합 디버거 시작 태스크 이름	<ul style="list-style-type: none"> DBGMGR FEK.#CUST.PROCLIB (DBGMGR) (IBM Rational Developer for System z Host Configuration Guide의 "PROCLIB 변경사항"에서 설명함) 	
JES 작업 모니터 시작된 태스크 이름	<ul style="list-style-type: none"> JMON FEK.#CUST.PROCLIB (JMON) (IBM Rational Developer for System z Host Configuration Guide의 "PROCLIB 변경사항"에 설명됨) 	
RSE 디먼 시작된 태스크 이름	<ul style="list-style-type: none"> RSED FEK.#CUST.PROCLIB (RSED) (IBM Rational Developer for System z Host Configuration Guide의 "PROCLIB 변경사항"에 설명됨) 	
애플리케이션 ID	<ul style="list-style-type: none"> FEKAPPL /etc/rdz/rsed.envvars (IBM Rational Developer for System z Host Configuration Guide의 "_RSE_JVAOPTS를 사용하여 추가 Java 시작 매개변수 정의"에 설명됨) 	

다음 목록은 Developer for System z의 기본 보안 설정을 완료하는 데 필요한 조치 개요입니다. 다음 절에 설명된 대로 필수 보안 레벨에 따라 여러 방법을 사용하여 이러한 요구사항을 충족시킬 수 있습니다. 선택적 Developer for System z 서비스 보안 설정에 대한 정보는 앞의 절을 참조하십시오.

- 50 페이지의 『보안 설정 및 클래스 활성화』
- 51 페이지의 『Developer for System z 사용자에게 대한 OMVS 세그먼트 정의』
- 51 페이지의 『Developer for System z 시작 태스크 정의』

- 53 페이지의 『RSE를 보안 z/OS UNIX 서버로 정의』
- 54 페이지의 『RSE에 대한 MVS 프로그램 제어 라이브러리 정의』
- 55 페이지의 『RSE에 대한 PassTicket 지원 정의』
- 56 페이지의 『RSE에 대한 애플리케이션 보호 정의』
- 56 페이지의 『JES 명령 보안 정의』
- 58 페이지의 『데이터 세트 프로파일 정의』
- 64 페이지의 『RSE에 대한 z/OS UNIX 프로그램 제어 파일 정의』
- 64 페이지의 『보안 설정 확인』

보안 설정 및 클래스 활성화

Developer for System z는 다양한 보안 메커니즘을 사용하여 안전하고 제어된 클라이언트의 호스트 시스템 환경을 확보합니다. 이를 위해 여러 클래스 및 보안 설정을 다음 샘플 RACF 명령에 표시된 대로 활성화해야 합니다.

- 현재 설정 표시
 - SETROPTS LIST
- z/OS UNIX 및 디지털 인증서 프로파일에 대해 facility 클래스 활성화
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- 시작된 태스크 정의 활성화
 - SETROPTS GENERIC(STARTED)
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- JES 작업 모니터에 대한 콘솔 보안 활성화
 - SETROPTS GENERIC(CONSOLE)
 - SETROPTS CLASSACT(CONSOLE) RACLIST(CONSOLE)
- JES 작업 모니터에 대한 운영자 명령 보호 활성화
 - SETROPTS GENERIC(OPERCMDS)
 - SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)
- RSE에 대한 애플리케이션 보호 활성화
 - SETROPTS GENERIC(APPL)
 - SETROPTS CLASSACT(APPL) RACLIST(APPL)
- RSE에 대해 PassTicket을 사용하여 보안 사인온 활성화
 - SETROPTS GENERIC(PTKTDATA)
 - SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)

- 프로그램 제어를 활성화하여 RSE가 신뢰할 수 있는 코드만 로드하도록 보장
 - RDEFINE PROGRAM ** ADDMEM('SYS1.CMDLIB'//NOPADCHK) UACC(READ)
 - SETROPTS WHEN(PROGRAM)

참고: PROGRAM 클래스에 * 프로파일이 이미 있으면 ** 프로파일을 작성하지 마십시오. 이 프로파일을 사용하면 보안 소프트웨어에서 사용하는 검색 경로가 모호하고 복잡해집니다. 이 경우, 기존 * 및 새 ** 정의를 병합해야 합니다. *Security Server RACF Security Administrator's Guide(SA22-7683)*에 설명된 대로 ** 프로파일을 사용하십시오.

주의: FTP와 같은 일부 제품의 경우 "WHEN PROGRAM"이 활성화되어 있으면 프로그램 제어가 필요합니다. 이 프로그램 제어를 테스트한 후에 프로덕션 시스템에서 활성화하십시오.

- (선택사항) X.509 HostIdMappings 및 확장 POE(Port Of Entry) 지원 활성화
 - SETROPTS GENERIC(SERVAUTH)
 - SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)

Developer for System z 사용자에게 대한 OMVS 세그먼트 정의

Developer for System z 사용자마다 0이 아닌 올바른 z/OS UNIX 사용자 ID(UID), 홈 디렉토리 및 셸 명령을 지정하는 RACF OMVS 세그먼트(또는 동등 기능)를 정의해야 합니다. 기본 그룹에는 그룹 ID가 있는 OMVS 세그먼트도 필요합니다.

선택적 통합 디버거를 사용할 때 애플리케이션을 디버깅하는 중인 사용자 ID가 활성화되며 해당 기본 그룹 또한 유효한 RACF OMVS 세그먼트 또는 그와 동등한 기능이 필요합니다.

다음 샘플 RACF 명령에서 #userid, #user-identifier, #group-name 및 #group-identifier 플레이스홀더를 실제 값으로 대체하십시오.

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

Developer for System z 시작 태스크 정의

다음 샘플 RACF 명령은 보호된 사용자 ID(STCDBGM, STCJMON 및 STCRSE)와 지정된 FEKD, DBGMGR, JMON 및 RSED 시작 태스크를 작성합니다. #group-id 및 #user-id-* 플레이스홀더를 올바른 OMVS ID로 대체하십시오.

- ```
ADDGROUP STCGROUP OMVS(GID(#group-id))
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
```

```

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

```

#### 참고:

- NOPASSWORD 키워드를 지정하여 시작된 태스크 사용자 ID가 보호되는지 확인하십시오.
- 이 uid에 부여된 z/OS UNIX 관련 권한 때문에 RSE 서버에 고유 OMVS uid가 있는지 확인하십시오.
- RSE 디먼이 적절히 작동하려면 주소 공간 크기가 커야 합니다(2GB). 사용자 ID STCRSE에 대해 OMVS 세그먼트의 ASSIZEMAX 변수에 이 값을 설정하십시오. 이 값을 설정하면 SYS1.PARMLIB(BPXPRMxx)의 MAXASSIZE 변경에 관계없이 RSE 디먼이 필요한 리전 크기를 갖습니다.
- RSE가 올바르게 작동하려면 스레드 수도 많아야 합니다. 사용자 ID STCRSE에 대해 OMVS 세그먼트의 THREADSMAX 변수에 한계를 설정할 수 있습니다. 이 한계를 설정하면 SYS1.PARMLIB(BPXPRMxx)의 MAXTHREADS 또는 MAXTHREADTASKS 변경에 관계없이 RSE가 필요한 스레드 한계를 갖습니다. 올바른 스레드 한계 값을 결정하려면 호스트 구성 참조서 (SA30-4501)의 "튜닝 고려사항"을 참조하십시오.
- JES 작업 모니터는 클라이언트 연결당 하나의 스레드를 사용하기 때문에 사용자 ID STCJMON은 OMVS 세그먼트에 THREADSMAX를 설정하기에 적합한 또 다른 후보입니다.

- 통합 디버거 시작 태스크(DBGMGR)는 선택사항인 통합 디버거 기능에서만 사용됩니다.

STCRSE 사용자 ID를 제한할 것을 고려하십시오. RESTRICTED 속성을 가진 사용자는 명확하게 액세스 권한이 부여되지 않았기 때문에 보호(MVS) 자원에 액세스할 수 없습니다.

ALTUSER STCRSE RESTRICTED

제한된 사용자가 "기타" 권한 비트를 통해 z/OS UNIX 파일 시스템 자원에 대한 액세스를 확보하지 않게 하려면 UACC(NONE)을 사용하여 UNIXPRIV 클래스에 RESTRICTED.FILESYS.ACCESS 프로파일을 정의하십시오. 사용자 ID 제한에 대한 자세한 정보는 *Security Server RACF Security Administrator's Guide*(SA22-7683)를 참조하십시오.

**경고:** 제한된 사용자 ID를 사용하는 경우, TSO **PERMIT** 또는 z/OS UNIX **setfac1** 명령을 사용하여 자원 액세스 권한을 명시적으로 추가하십시오. 이 자원에는 Developer for System z 문서가 UACC를 사용하는 자원(예: PROGRAM 클래스의 \*\* 프로파일) 또는 공통 z/OS UNIX 규약(예: 모든 사람이 Java 라이브러리에 대한 읽기 및 실행 권한을 가짐)에 의존하는 자원이 포함됩니다. 액세스를 테스트한 후에 프로덕션 시스템에서 활성화하십시오.

## RSE를 보안 z/OS UNIX 서버로 정의

클라이언트의 스레드에 대한 보안 환경을 작성 또는 삭제하려면 RSE에 BPX.SERVER 프로파일에 대한 UPDATE 액세스 권한이 필요합니다. 이 프로파일이 정의되지 않은 경우에는 RSE에 UID(0)이 필요합니다. 클라이언트가 연결할 수 있으려면 이 단계가 필요합니다.

디버그 스레드에 대한 보안 환경을 작성 또는 삭제하려면 통합 디버거에서 BPX.SERVER 프로파일에 대한 UPDATE 액세스 권한이 필요합니다. 이 프로파일이 정의되지 않은 경우 STCDBM 시작 태스크 사용자 ID에 대해 UID(0)가 필요합니다. 이 권한은 선택적 통합 디버거 기능이 사용되는 경우에만 필요합니다.

- 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

**경고:** BPX.SERVER 프로파일을 정의하면 z/OS UNIX가 UNIX 레벨 보안에서 보다 안전한 z/OS UNIX 레벨 보안으로 전체 전환됩니다. 이러한 전환으로 다른 z/OS UNIX 애플리케이션 및 조작에 영향을 줄 수 있습니다. 보안을 테스트한 후에 프로덕션 시스템에서 활성화하십시오. 다른 보안 레벨에 대한 자세한 정보는 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.



## RSE에 대한 MVS 프로그램 제어 라이브러리 정의

BPX.SERVER에 대한 권한을 가진 서버는 프로그램으로 제어되는 정리된 환경에서 실행해야 합니다. 이 요구사항은 RSE로 호출되는 모든 프로그램도 프로그램으로 제어되어야 함을 의미합니다. MVS 로드 라이브러리의 경우, 보안 소프트웨어가 프로그램 제어를 관리합니다. 클라이언트가 연결할 수 있으려면 이 단계가 필요합니다.

RSE는 시스템(SYS1.LINKLIB), Language Environment의 런타임(CEE.SCEERUN\*), ISPF의 TSO/ISPF Client Gateway(ISP.SISPLoad) 로드 라이브러리를 사용합니다.

- 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

참고: PROGRAM 클래스에 \* 프로파일이 이미 있으면 \*\* 프로파일을 사용하지 마십시오. 이 프로파일을 사용하면 보안 소프트웨어에서 사용하는 검색 경로가 모호하고 복잡해집니다. 이 경우, 기존 \* 및 새 \*\* 정의를 병합해야 합니다. *Security Server RACF Security Administrator's Guide*(SA22-7683)에 설명된 대로 \*\* 프로파일을 사용하십시오.

선택적 서비스 사용을 지원하려면 다음 추가 필수 라이브러리가 프로그램으로 제어되어야 합니다. 이 목록에는 IBM File Manager와 같이 Developer for System z가 상호 작용하는 제품에 특정한 데이터 세트가 포함되지 않습니다.

- 대체 REXX 런타임 라이브러리, SCLM 개발자 툴킷용
  - REXX.\*.SEAGALT
- 시스템 로드 라이브러리, SSL 암호화용
  - SYS1.SIEALNKE
- Developer for System z 라이브러리(통합 디버거의 경우)
  - FEK.SFEKAUTH

참고: LPA 배치를 위해 디자인된 라이브러리의 경우도 LINKLIST 또는 STEPLIB를 통해 액세스할 경우 프로그램 제어 권한이 필요합니다. 이 책에서는 다음 LPA 라이브러리 사용을 설명합니다.

- ISPF, TSO/ISPF Client Gateway용
  - ISP.SISPLPA
- REXX 런타임 라이브러리, SCLM 개발자 툴킷용
  - REXX.\*.SEAGLPA
- Developer for System z, CARMA용



## RSE에 대한 PassTicket 지원 정의

클라이언트의 비밀번호 또는 X.509 인증서 같은 다른 식별 수단은 연결 시 ID를 확인하는 데만 사용됩니다. 나중에는 PassTicket을 사용하여 스레드 보안을 유지보수합니다. 클라이언트가 연결할 수 있으려면 이 단계가 필요합니다.

PassTicket은 수명이 10분 정도인 시스템 생성 비밀번호입니다. 생성된 PassTicket은 비밀 키를 기반으로 합니다. 이 키는 64비트 숫자입니다(16개의 16진 문자). 다음 샘플 RACF 명령에서 key16 플레이스홀더를 사용자가 제공하는 16자 16진 문자열(문자 0-9 및 A-F)로 대체하십시오.

- ```
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)
```
- ```
SETRPTS RACLIST(PTKTDATA) REFRESH
```

RSE는 FEKAPPL이 아닌 애플리케이션 ID 사용을 지원합니다. 이를 활성화하려면 *IBM Rational Developer for System z Host Configuration Guide*의 "\_RSE\_JAVAOPTS"를 사용하여 추가 Java 시작 매개변수 정의"에 설명된 대로 `rsed.envvars`의 "APPLID=FEKAPPL" 옵션을 주석 해제하고 사용자 정의하십시오. PTKTDATA 클래스 정의는 RSE에서 사용하는 실제 애플리케이션 ID와 일치해야 합니다.

대부분의 z/OS UNIX 애플리케이션 비밀 키를 열기 때문에 OMVSAPPL을 애플리케이션 ID로 사용해서는 안 됩니다. 사용자 일괄처리 작업을 포함하여 대부분의 MVS 애플리케이션 비밀 키를 열기 때문에 기본 MVS 애플리케이션 ID(MVS 다음에 시스템의 SMF ID가 음)도 사용해서는 안 됩니다.

### 참고:

- PTKTDATA 클래스가 이미 정의되어 있는 경우, 앞서 나열한 프로파일을 작성하기 전에 일반 클래스로 정의되어 있는지 확인하십시오. PTKTDATA 클래스에서 일반 문자에 대한 지원은 PassTicket에 Java 인터페이스가 도입되면서 z/OS 릴리스 1.7부터 새로운 기능입니다.
- IRRPTAUTH.FEKAPPL.\* 정의의 와일드카드(\*)를 올바른 사용자 ID 마스크로 대체하여 RSE가 PassTicket을 생성할 수 있는 사용자 ID를 제한하십시오.

- RACF 설정에 따라 프로파일을 정의하는 사용자도 프로파일의 액세스 목록에 있을 수 있습니다. PTKTDATA 프로파일의 경우 이 권한을 제거하십시오.
- JES 작업 모니터가 RSE가 제공한 PassTicket을 평가할 수 있으려면 JES 작업 모니터 및 RSE는 애플리케이션 ID가 동일해야 합니다. JES 작업 모니터의 경우, 애플리케이션 ID는 APPLID 지시문이 있는 FEJJCNFG 구성 파일에 설정됩니다.
- 시스템에 암호화 제품이 설치되어 사용할 수 있는 경우, 추가된 보호에 대해 보안 사인은 애플리케이션 키를 암호화할 수 있습니다. 이를 수행하려면 KEYMASKED 대신 KEYENCRYPTED 키워드를 사용하십시오. 자세한 정보는 *Security Server RACF Security Administrator's Guide*(SA22-7683)를 참조하십시오.

경고: PassTicket이 올바르게 설정되지 않으면 클라이언트 연결 요청이 실패합니다.

## RSE에 대한 애플리케이션 보호 정의

클라이언트 로그인 중에 RSE 디먼은 사용자가 애플리케이션을 사용할 수 있는지 확인합니다.

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

참고:

- 55 페이지의 『RSE에 대한 PassTicket 지원 정의』에 자세히 설명된 대로 RSE는 FEKAPPL이 아닌 애플리케이션 ID 사용을 지원합니다. APPL 클래스 정의는 RSE에서 사용하는 실제 애플리케이션 ID와 일치해야 합니다.
- 애플리케이션 ID가 APPL 클래스에 정의되어 있지 않으면 클라이언트 연결 요청이 성공합니다.
- 애플리케이션 ID가 정의되어 있으나 사용자에게 프로파일에 대한 READ 액세스 권한이 없는 경우에만 클라이언트 연결 요청이 실패합니다.

## JES 명령 보안 정의

JES 작업 모니터는 확장 MCS(EMCS) 콘솔(해당 이름은 CONSOLE\_NAME 지시문으로 제어됨)을 통해 사용자에게 의해 요청된 모든 JES 연산자 명령을 실행합니다. 이 내용은 *IBM Rational Developer for System z Host Configuration Guide*의 "FEJJCNFG, JES 작업 모니터 구성 파일"에 설명되어 있습니다.

다음 샘플 RACF 명령은 Developer for System z 사용자에게 제한된 JES 명령 세트(보류, 해제, 취소 및 제거)에 대한 조건부 액세스 권한을 제공합니다. 사용자가 JES 작

업 모니터를 통해 명령을 발행할 경우 사용자는 실행 권한만 가집니다. #console 플레이스홀더를 실제 콘솔 이름으로 대체하십시오.

- ```
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
```

#### 참고:

- MVS.MCSOPER.#console 프로파일을 정의하지 않으면 콘솔 사용이 허용됩니다.
- WHEN(CONSOLE(JMON))이 작동하려면 CONSOLE 클래스가 활성화되어 있어야 하지만 EMCS 콘솔의 경우 CONSOLE 클래스에서 실제 프로파일 검사는 없습니다.
- WHEN(CONSOLE(JMON)) 절에서 JMON을 실제 콘솔 이름으로 대체하지 마십시오. JMON 키워드는 콘솔 이름이 아닌 진입점 애플리케이션을 나타냅니다.

**경고:** 보안 소프트웨어에 유니버설 액세스 NONE을 사용하여 JES 명령을 정의하면 다른 애플리케이션과 조작에 영향을 줄 수 있습니다. 보안을 테스트한 후에 프로덕션 시스템에서 활성화하십시오.

표 11 및 58 페이지의 표 12은 JES2 및 JES3에 대해 실행되는 운영자 명령과 이를 보호하는 데 사용할 수 있는 개별 보안 프로파일을 표시합니다.

표 11. JES2 작업 모니터 운영자 명령

| 조치 | 명령                               | OPERCMDS 프로파일                                                                       | 필수 액세스 권한 |
|----|----------------------------------|-------------------------------------------------------------------------------------|-----------|
| 보류 | \$Hx(jobid)<br>x = {J, S 또는 T}   | jesname.MODIFYHOLD.BAT<br>jesname.MODIFYHOLD.STC<br>jesname.MODIFYHOLD.TSU          | UPDATE    |
| 해제 | \$Ax(jobid)<br>x = {J, S 또는 T}   | jesname.MODIFYRELEASE.BAT<br>jesname.MODIFYRELEASE.STC<br>jesname.MODIFYRELEASE.TSU | UPDATE    |
| 취소 | \$Cx(jobid)<br>x = {J, S 또는 T}   | jesname.CANCEL.BAT<br>jesname.CANCEL.STC<br>jesname.CANCEL.TSU                      | UPDATE    |
| 제거 | \$Cx(jobid),P<br>x = {J, S 또는 T} | jesname.CANCEL.BAT<br>jesname.CANCEL.STC<br>jesname.CANCEL.TSU                      | UPDATE    |

표 12. JES3 작업 모니터 운영자 명령

| 조치 | 명령           | OPERCMDS 프로파일      | 필수 액세스 권한 |
|----|--------------|--------------------|-----------|
| 보류 | *F,J=jobid,H | jesname.MODIFY.JOB | UPDATE    |
| 해제 | *F,J=jobid,R | jesname.MODIFY.JOB | UPDATE    |
| 취소 | *F,J=jobid,C | jesname.MODIFY.JOB | UPDATE    |
| 제거 | *F,J=jobid,C | jesname.MODIFY.JOB | UPDATE    |

#### 참고:

- 값이 LIMITED 또는 NOLIMIT인 LIMIT\_COMMANDS=가 JES 작업 모니터 구성 파일에 지정되어 있지 않으면 클라이언트 사용자 ID가 소유한 스푼 파일에 대해서만 보류, 해제, 취소, 제거 JES 운영자 명령과 JCL 표시 명령을 실행할 수 있습니다. 자세한 정보는 호스트 구성 참조서 (SA30-4501)의 "작업에 대한 조치 - 대상 제한사항"를 참조하십시오.
- LIMIT\_VIEW=USERID가 JES 작업 모니터 구성 파일에 정의되어 있지 않으면 사용자가 스푼 파일을 찾아볼 수 있습니다. 자세한 정보는 호스트 구성 참조서 (SA30-4501)의 "스푼 파일에 대한 액세스"를 참조하십시오.
- 사용자에게 이러한 운영자 명령에 대한 권한이 없더라도 이러한 자원(예: JESINPUT, JESJOBS 및 JESSPOOL 클래스의 자원)을 보호하는 가능한 프로파일에 대한 충분한 권한이 있으면 JES 작업 모니터를 통해 여전히 작업을 제출하고 작업 출력을 읽을 수 있습니다.

TSO 세션에서 JMON 콘솔을 작성하여 JES 작업 모니터 서버의 ID를 가정하는 것은 보안 소프트웨어에서 금지됩니다. 콘솔을 작성할 수 있지만 진입점이 다릅니다(예: JES 작업 모니터 대 TSO). 이 책에 설명된 대로 보안이 설정되고 사용자에게 다른 방법을 통해 JES 명령에 대한 권한이 없으면 이 콘솔에서 실행된 JES 명령은 보안 검사에 실패합니다.

## 데이터 세트 프로파일 정의

대부분의 Developer for System z 데이터 세트의 경우 사용자에게 대한 READ 액세스와 시스템 프로그래머에 대한 ALTER이면 충분합니다. #sysprog 플레이스홀더를 올바른 사용자 ID 또는 RACF 그룹 이름으로 대체하십시오. 또한 제품을 설치하여 구성된 시스템 프로그래머에게 올바른 데이터 세트 이름을 요청하십시오. FEK는 설치 중에 사용되는 기본 상위 레벨 규정자이고 FEK.#CUST는 사용자 정의 프로세스 중에 작성된 데이터 세트의 기본 상위 레벨 규정자입니다.

- ```
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
```

참고:

- 이 데이터 세트는 APF의 인증을 받았으므로 FEK.SFEKAUTH가 업데이트되지 않도록 보호합니다. FEK.SFEKLOAD 및 FEK.SFEKLPA의 경우도 마찬가지지만 이러한 데이터 세트는 프로그램 제어되기 때문에 여기서는 언급하지 않습니다.
- 이 책과 FEKRACF 작업에서의 샘플 명령은 EGN(Enhanced Generic Naming)이 활성화되어 있다고 가정합니다. EGN이 활성화되면 ** 규정자를 사용하여 DATASET 클래스의 규정자를 얼마든지 나타낼 수 있습니다. 시스템에서 EGN이 활성화되어 있지 않으면 **를 *로 대체하십시오. EGN에 대한 자세한 정보는 *Security Server RACF Security Administrator's Guide*(SA22-7683)를 참조하십시오.

일부 선택적 Developer for System z 컴포넌트에는 보안 데이터 세트 프로파일이 추가로 필요합니다. #sysprog, #ram-developer, #cicsadmin 플레이스홀더를 올바른 사용자 ID 또는 RACF 그룹 이름으로 대체하십시오.

- SCLM 개발자 툴킷의 긴/짧은 이름 변환이 사용되는 경우, 사용자에게는 맵핑 VSAM(FEK.#CUST.LSTRANS.FILE)에 대한 UPDATE 액세스 권한이 필요합니다.

—

```
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 저장소 액세스 관리자(RAM) 개발자에게는 CARMA VSAM(FEK.#CUST.CRA*)에 대한 UPDATE 액세스 권한이 필요합니다.

—

```
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

- 애플리케이션 배치 관리자의 CRD(CICS 자원 정의) 서버가 사용되는 경우, CICS 관리자에게는 CRD 저장소 VSAM에 대한 UPDATE 액세스 권한이 필요합니다.

—

```
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

- 애플리케이션 배치 관리자의 Manifest 저장소가 정의된 경우, 모든 CICS Transaction Server 사용자에게는 Manifest 저장소 VSAM에 대한 UPDATE 액세스 권한이 필요합니다.

—

```
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

READ 액세스도 제어되는 경우 보다 안전한 설정을 위해 다음 샘플 RACF 명령을 사용하십시오.

- uacc(none) 데이터 세트 보호

—

```
ADDGROUP (FEK)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
OWNER(IBMUSER) SUPGROUP(SYS1)"
```

—

```
ADDSD 'FEK.*.*' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```

—

```
ADDSD 'FEK.SFEKAUTH' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```

—

```
ADDSD 'FEK.SFEKLOAD' UACC(NONE)
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
```

```

- ADDSD 'FEK.SFEKLMOD' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

-

  ADDSD 'FEK.SFEKPROC' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

-

  ADDSD 'FEK.#CUST.PARMLIB' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

-

  ADDSD 'FEK.#CUST.CNTL' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

-

  ADDSD 'FEK.#CUST.SQL' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z')

-

  ADDSD 'FEK.#CUST.LSTRANS*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')

-

  ADDSD 'FEK.#CUST.CRA*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')

-

  ADDSD 'FEK.#CUST.ADNREP*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')

-

  ADDSD 'FEK.#CUST.ADNMAN*.**' UACC(NONE)
  DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
• 시스템 프로그래머가 모든 라이브러리를 관리하도록 허용

-

  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)

```

- 클라이언트가 로드 및 실행 라이브러리에 액세스하도록 허용

```

-
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(*)

```

참고: 모든 사람이 LPA에 상주하는 모든 코드에 액세스할 수 있으므로 FEK.SFEKLPA에 대해서는 허용이 필요하지 않습니다.

- 통합 디버거가 로드 라이브러리에 액세스하도록 허용

```

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

```

- JES 작업 모니터가 로드 및 매개변수 라이브러리에 액세스하도록 허용

```

-
PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
-

```

```

PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(READ) ID(STCJMON)

```

- (선택사항) 클라이언트가 SCLMDT용 긴/짧은 이름 변환 VSAM을 업데이트하도록 허용

```

-
PERMIT 'FEK.#CUST.LSTRANS.*.**' CLASS(DATASET) ACCESS(UPDATE) ID(*)

```

- (선택사항) RAM 개발자가 CARMA용 CARMA VSAM을 업데이트하도록 허용

```

-

```


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

- (선택사항) CICS 사용자가 애플리케이션 배치 관리자용 CRD 저장소 VSAM을 읽을 수 있도록 허용

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

- (선택사항) CICS 관리자가 애플리케이션 배치 관리자용 CRD 저장소 VSAM을 업데이트하도록 허용

PERMIT 'FEK.#CUST.ADNREP*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)

- (선택사항) CICS 사용자가 애플리케이션 배치 관리자용 Manifest 저장소 VSAM을 업데이트하도록 허용

PERMIT 'FEK.#CUST.ADNMAN*.*' CLASS(DATASET) ACCESS(UPDATE) ID(*)

- (선택사항) CICS TS 서버가 양방향 및 애플리케이션 배치 관리자용 로드 라이브러리에 액세스하도록 허용

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

- (선택사항) CICS TS 서버, IMS™ 영역 및 MVS 일괄처리 작업이 IRZ 메시지의 로드 라이브러리에 액세스하도록 허용

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)

- 보안 프로파일 활성화

SETROPTS GENERIC(DATASET) REFRESH

시스템 데이터 세트에 대한 READ 액세스를 제어하는 경우, Developer for System z 서버와 사용자에게 다음 데이터 세트에 대한 READ 권한을 제공해야 합니다.

- CEE.SCEERUN
- CEE.SCEERUN2
- CBC.SCLBDLL
- ISP.SISPLoad
- ISP.SISPLPA
- SYS1.LINKLIB
- SYS1.SIEALNKE

- SYS1.SIEAMIGE
- REXX.V1R4M0.SEAGLPA

참고: Alternate Library for REXX 제품 패키지를 사용하는 경우, 기본 REXX 런타임 라이브러리 이름은 위 샘플에서 사용된 대로 REXX.*.SEAGLPA 대신 REXX.*.SEAGALT입니다.

RSE에 대한 z/OS UNIX 프로그램 제어 파일 정의

BPX.SERVER에 대한 권한을 가진 서버는 프로그램으로 제어되는 정리된 환경에서 실행해야 합니다. 이 요구사항은 RSE로 호출되는 모든 프로그램도 프로그램으로 제어되어야 함을 의미합니다. z/OS UNIX 파일의 경우, **extattr** 명령이 프로그램 제어를 관리합니다. 이 명령을 실행하려면 FACILITY 클래스의 BPX.FILEATTR.PROGCTL에 대한 READ 액세스 권한 또는 UID(0)이 필요합니다.

RSE 서버는 RACF의 Java 공유 라이브러리 (/usr/lib/libIRRRacf*.so)를 사용합니다.

- `extattr +p /usr/lib/libIRRRacf*.so`

참고:

- z/OS 1.9부터 /usr/lib/libIRRRacf*.so는 SMP/E RACF 설치 중에 프로그램 제어 모드로 설치됩니다.
- z/OS 1.10부터 /usr/lib/libIRRRacf*.so는 SAF의 일부로, 기본 z/OS와 함께 제공되므로 비RACF 고객도 사용할 수 있습니다.
- RACF 이외의 다른 보안 제품을 사용하는 경우 설정이 다를 수 있습니다. 자세한 정보는 보안 제품 문서를 참조하십시오.
- Developer for System z의 SMP/E 설치에는 내부 RSE 프로그램에 대한 프로그램 제어 비트를 설정합니다.
- **ls -Eog** z/OS UNIX 명령을 사용하여 프로그램 제어 비트의 현재 상태를 표시하십시오. 문자 **p**가 두 번째 문자열에 표시되면 파일은 프로그램에서 제어됩니다.

```
$ 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
```

보안 설정 확인

보안 관련 사용자 정의 결과를 표시하려면 다음 샘플 명령을 사용하십시오.

- 보안 설정 및 클래스
 - SETROPTS LIST
- 사용자에게 대한 OMVS 세그먼트
 - LISTUSER #userid NORACF OMVS

- LISTGRP #group-name NORACF OMVS
- 시작된 태스크
 - LISTGRP STCGROUP OMVS
 - LISTUSER STCDBM OMVS
 - LISTUSER STCJMON OMVS
 - LISTUSER STCRSE OMVS
 - RLIST STARTED DBGMGR.* ALL STDATA
 - RLIST STARTED JMON.* ALL STDATA
 - RLIST STARTED RSED.* ALL STDATA
- 보안 z/OS UNIX 서버로서의 RSE
 - RLIST FACILITY BPX.SERVER ALL
- RSE 대한 MVS 프로그램 제어 라이브러리
 - RLIST PROGRAM ** ALL
- RSE에 대한 PassTicket 지원
 - RLIST PTKTDATA FEKAPPL ALL SSIGNON
 - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.* ALL
- RSE에 대한 애플리케이션 보호
 - RLIST APPL FEKAPPL ALL
- JES 명령 보안
 - RLIST CONSOLE JMON ALL
 - RLIST OPERCMDS MVS.MCSOPER.JMON ALL
 - RLIST OPERCMDS JES%.** ALL
- 데이터 세트 프로파일
 - LISTGRP FEK
 - LISTDSD PREFIX(FEK) ALL
- RSE 대한 z/OS UNIX 프로그램 제어 파일
 - ls -E /usr/lib/libIRRRacf*.so

선택적으로 특정 사용자에게 대한 Developer for System z 동작을 지시하는 프로파일이 있을 수 있습니다. 이 프로파일은 FEK.** 필터와 일치하며 기본적으로 FACILITY 클래스에 있습니다. rsed.envvars의 _RSE_FEK_SAF_CLASS 지시문을 참조하십시오. **SEARCH** 명령을 사용하여 프로파일 이름을 나열할 수 있습니다. 프로파일에 대한 세 부사항을 표시하려면 **RLIST** 명령을 사용하십시오.

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

제 3 장 TCP/IP 고려사항

Developer for System z는 TCP/IP를 사용하여 비메인프레임 워크스테이션 사용자에게 메인프레임 액세스를 제공합니다. 또한 다양한 컴포넌트와 기타 제품 간의 통신에도 TCP/IP를 사용합니다.

대부분의 Developer for System z 함수는 z/OS UNIX 기반이므로, TCP/IP는 z/OS UNIX 검색 순서를 사용하여 해당 구성 파일을 찾습니다. 자세한 정보는 227 페이지의 제 14 장 『TCP/IP 설정』의 내용을 참조하십시오.

이 장에서 다루는 주제는 다음과 같습니다.

- 『TCP/IP 포트』
- 70 페이지의 『기본 TCP/IP 동작 대체』
- 71 페이지의 『다중 스택(CINET)』
- 72 페이지의 『분산 동적 VIPA』

TCP/IP 포트

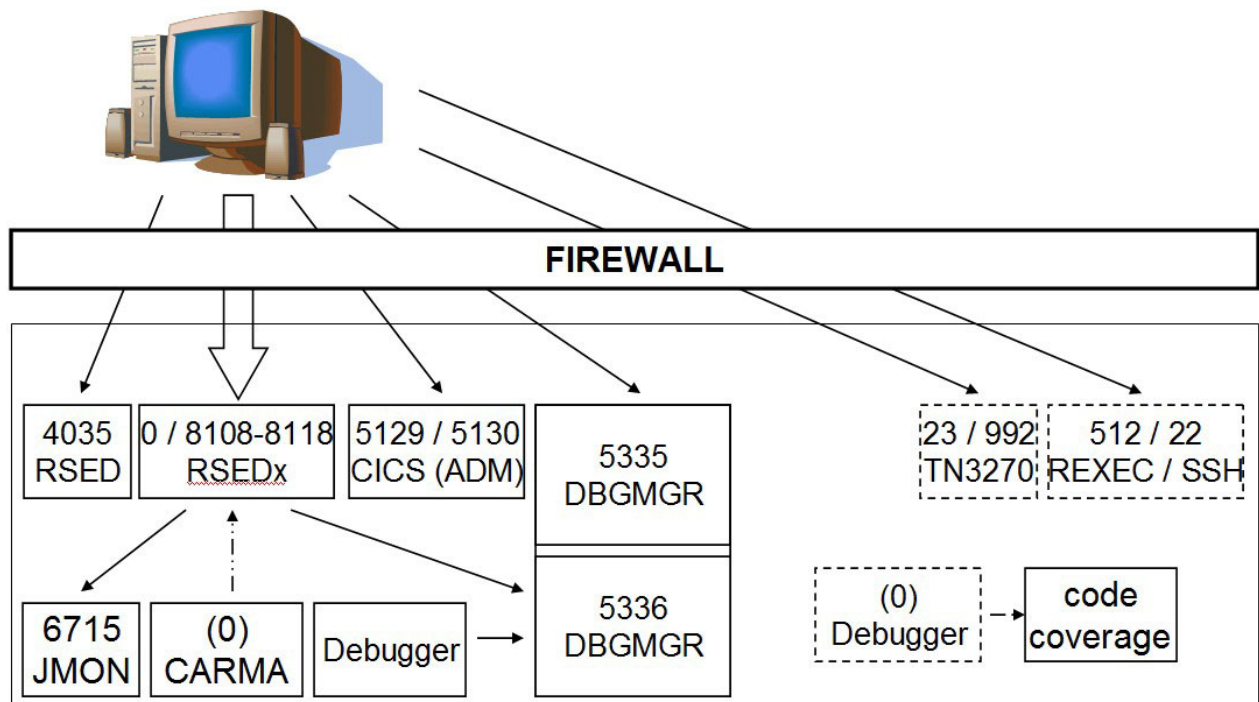


그림 10. TCP/IP 포트

67 페이지의 그림 10은 Developer for System z가 사용할 수 있는 TCP/IP 포트를 보여줍니다. 화살표는 바인드를 수행하는 부분(화살촉 부분)과 연결 부분을 보여줍니다.

외부 통신

z/OS 호스트를 보호하는 방화벽에 다음 포트를 정의합니다. 이 포트는 tcp 프로토콜을 사용한 클라이언트-호스트 통신에 사용됩니다.

- 클라이언트-호스트 통신 설정을 위한 RSE 디먼, 기본 포트 4035. 포트는 rsed.envvars 구성 파일에서 설정될 수 있습니다. 이 포트의 통신은 SSL 또는 TLS를 사용하여 암호화될 수 있습니다.
- 클라이언트-호스트 통신을 위한 RSE 서버. 기본적으로, 사용 가능한 포트를 모두 사용할 수 있지만 rsed.envvars의 _RSE_PORTRANGE 정의에 따라 지정된 범위로 제한될 수 있습니다. _RSE_PORTRANGE의 기본 포트 범위는 8108-8118(11개 포트)입니다. 이 포트의 통신은 SSL 또는 TLS를 사용하여 암호화될 수 있습니다.
- (선택사항) 통합 디버거 서비스를 위한 디버그 관리자, 기본 포트 5335. 포트는 DBGMR 시작 태스크 JCL에서 설정될 수 있습니다. 이 포트의 통신은 SSL 또는 TLS를 사용하여 암호화될 수 있습니다.
- (선택사항) z/OS UNIX 서브프로젝트의 호스트 기반 원격 조치에 대한 INETD 서비스:
 - REXEC(z/OS UNIX 버전), 기본 포트 512
 - SSH(z/OS UNIX 버전), 기본 포트 22. 이 포트의 통신은 SSL을 사용하여 암호화됩니다.
- (선택사항) 호스트 연결 에뮬레이터에 대한 TN3270 Telnet 서비스, 기본 포트 23. 통신은 SSL 또는 TLS를 사용하여 암호화될 수 있습니다(기본 포트 992). TN3270 Telnet 서비스에 지정되는 기본 포트는 사용자가 암호화 사용을 선택하는지 여부에 따라 다릅니다.
- (선택사항) 애플리케이션 배치 관리자의 CICSTS 애플리케이션 인터페이스 각각 또는 둘 다
 - RESTful 인터페이스, 기본 포트 5130. 포트는 CICS CSD에서 설정될 수 있습니다.
 - 웹 서비스 인터페이스, 기본 포트 5129. 포트는 CICS CSD에서 설정될 수 있습니다. 이 포트의 통신은 SSL을 사용하여 암호화할 수 있습니다.

참고: 일반적으로 클라이언트는 호스트 연결에 사용되는 TCP/IP 주소를 지정합니다. 그러나 디버그 세션이 올바른 호스트와 통신하는지 확인하기 위해, 디버그 관리자는 TCP/IP 주소를 반드시 사용해야 하는 클라이언트를 지시합니다.

내부 통신

여러 Developer for System z 호스트 서비스가 개별 스레드 또는 주소 공간에서 실행되며 TCP/IP 소켓을 통신 메커니즘으로 사용합니다. 이러한 서비스는 모두 클라이언트와의 통신을 위해 RSE를 사용하므로 해당 데이터 스트림은 호스트로만 한정됩니다. 서비스에 따라 사용 가능한 포트를 사용하거나 시스템 프로그래머가 사용될 포트 또는 포트 범위를 선택할 수 있습니다.

- JES 관련 서비스의 JES 작업 모니터(기본 포트는 6715). 포트는 FEJJCNFG 구성 멤버에서 설정될 수 있으며 rsed.envvars 구성 파일에서 반복됩니다.
- (선택사항) CARMA 통신은 기본적으로 임시 포트를 사용하지만 포트 범위는 CRASRV.properties 구성 파일에서 설정할 수 있습니다.
- (선택사항) 디버그 관련 서비스를 위한 디버그 관리자, 기본 포트 5336. 포트는 DBGGMGR 시작 태스크 JCL에서 설정될 수 있습니다.
- 일괄처리 작업인 호스트 기반 코드 적용은 임시 포트를 할당하여 통합 디버거가 통신할 수 있고 코드 적용 보고서에 필요한 전달할 수 있도록 합니다.

TCP/IP 포트 예약

PROFILE.TCPIP에서 PORT 또는 PORTRANGE문을 사용하여 Developer for System z에서 사용하는 포트를 예약하는 경우, RSE 스레드 풀에서 활성화된 스레드로 많은 바인드가 수행됩니다. RSE 스레드 풀의 작업 이름은 RSEDx입니다. 여기서 RSED는 RSE 시작 태스크의 이름이고 x는 1자리 난수이므로 정의에서 와일드카드가 필요합니다.

```
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 및 TCP/IP 포트

CARMA(Common Access Repository Manager)는 호스트 기반 소프트웨어 구성 관리자(SCM)(예: CA Endevor® SCM)에 액세스하는 데 사용됩니다. 대부분의 경우 RSE 디먼과 마찬가지로 서버가 포트에 바인드되어 연결 요청을 청취합니다. 그러나 CARMA는 다른 접근 방법을 사용합니다. CARMA 서버는 클라이언트가 연결 요청을 시작할 때 아직 활성화되지 않기 때문입니다.

클라이언트가 연결 요청을 보내면 RSE 스레드 풀에서 사용자 스레드로 활성화되는 CARMA 마이너가 임시 포트를 요청하거나 CRASRV.properties 구성 파일에 지정된 범위에서 여유 포트를 찾아 바인드합니다. 그런 다음 마이너가 CARMA 서버를 시작하고 포트 번호를 전달하면 서버에서 연결할 포트를 알 수 있습니다. 서버가 연결되면 클라이언트가 서버로 요청을 보내고 결과를 받을 수 있습니다.

TCP/IP 퍼스펙티브에서 RSE(CARMA 마이너 경우)가 포트에 바인드되는 서버이고 CARMA 서버는 포트에 연결되는 클라이언트입니다.

PROFILE.TCPIP에서 PORT 또는 PORTRANGE문을 사용하여 CARMA가 사용하는 포트 범위를 예약하는 경우, CARMA 마이너는 RSE 스레드 풀에서 활성화됩니다. RSE 스레드 풀의 작업 이름은 RSEDx입니다. 여기서 RSED는 RSE 시작 태스크의 이름이고 x는 1자리 난수이므로 정의에 와일드카드가 필요합니다.

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

LDAP 고려사항

RSE 서버가 하나 이상의 LDAP 서버에서 다양한 Developer for System z 서비스를 조회하도록 구성할 수 있습니다.

- LDAP 그룹에서 여러 개발자 그룹이 지원하는 클라이언트로 푸시를 조회합니다.
- 하나 이상의 인증서 폐기 목록(CRL)에서 X.509 인증을 조회합니다.

TCP/IP 보안 조치(예: 방화벽)로 인해 (호스트 기반) RSE 서버가 LDAP 서버에 접속하지 못할 수 있습니다. 다음 정보를 사용하여 LDAP 서버에 도달할 수 있는지 확인할 수 있습니다.

- LDAP 서버 TCP/IP 주소 또는 DNS 이름은 rsed.envvars의 *_LDAP_SERVER 변수에 나열됩니다.
- LDAP 서버 포트 번호는 rsed.envvars의 *_LDAP_PORT 변수에 나열됩니다.
- LDAP는 TCP 프로토콜을 사용합니다.
- LDAP 서버에 호스트 기반 RSE 서버가 접속합니다.
- RSE 서버는 RSEDx 주소 공간에서 활성화됩니다. 여기서 RSED는 RSE 시작 태스크의 이름이고 x는 1자리 난수(예: RSED8)입니다.

기본 TCP/IP 동작 대체

ACK 지연

ACK 지연은 TCP 패킷 수신 확인 응답을 200ms까지 지연시킵니다. 이 지연은 수신된 패킷에 대한 응답과 함께 ACK를 보낼 수 있는 가능성을 증대하여 네트워크 트래픽을 줄입니다. 그러나 송신자가 새 패킷을 보내기 전에 ACK를 기다리고 있고(예를 들어, Nagle 알고리즘 구현으로 인해) 방금 보낸 패킷에 대한 응답이 없으면(예를 들어, 파일 전송의 일부이기 때문에) 통신이 불필요하게 지연됩니다.

Developer for System z에서는 ACK 지연 기능을 사용하지 않을 수 있습니다. *Host Configuration Guide*(SC23-7658)에 설명된 대로 rsed.envvars의 DSTORE_TCP_NO_DELAY 지시문을 사용하여 호스트에서 이를 수행합니다.

다중 스택(CINET)

z/OS Communication Server를 사용하면 여러 TCP/IP 스택을 동시에 단일 시스템에서 활성화할 수 있습니다. 이를 CINET 설정이라고 합니다.

Developer for System z가 기본 스택에서 활성화되지 않으면 선택한 Developer for System z 기능이 실패할 수 있습니다. 이 문제를 해결할 수 있는 확실한 방법은 스택 선호도를 사용하는 것입니다. 스택 선호도는 Developer for System z가 시작된 태스크의 기본값인 사용 가능한 모든 TCP/IP 스택 대신 특정 TCP/IP 스택만 사용하도록 지시합니다.

rsed.envvars 구성 파일에서 _BPXK_SETIBMOPT_TRANSPORT 지시문을 주석 해제 및 사용자 정의하여 RSED 시작 태스크에 대한 스택 선호도를 설정합니다. 이러한 구성 파일 사용자 정의에 대한 세부사항은 *Host Configuration Guide*(SC23-7658), "2장 기본 사용자 정의"의 관련 절을 참조하십시오.

CARMA 및 스택 선호도

CARMA(Common Access Repository Manager)는 호스트 기반 소프트웨어 구성 관리자(SCM)(예: CA Endevor® SCM)에 액세스하는 데 사용됩니다. 이를 수행하기 위해 CARMA는 스택 선호도를 강화하기 위해 추가 구성이 필요한 사용자별 서버를 시작합니다.

Developer for System z 시작 태스크와 마찬가지로, CARMA 서버에 대한 스택 선호도는 _BPXK_SETIBMOPT_TRANSPORT 변수로 설정됩니다. 이 변수는 LE(Language Environment)로 전달되어야 합니다. 이 작업은 활성 crastart*.conf 또는 CRASUB* 구성 파일에서 시작 명령을 조정하여 수행할 수 있습니다.

참고:

- 시작 명령이 있는 구성 파일의 정확한 이름은 CARMA를 구성한 시스템 프로그램어의 다양한 선택에 따라 다릅니다. 이에 대한 자세한 정보는 *Host Configuration Guide*(SC23-7658)의 "3장. (선택사항) CARMA(Common Access Repository Manager)"를 참조하십시오.
- _BPXK_SETIBMOPT_TRANSPORT는 사용될 TCP/IP 스택의 이름을 지정합니다(관련 TCPIP.DATA의 TCPIPJOBNAME 문에 정의됨).
- SYSTCPD DD 문 코딩 시 요청된 스택 선호도를 설정하지 않습니다.
- 기본적으로 CARMA는 일반 TCP/IP 스택을 사용하지 않습니다. CARMA는 CARMA 마이너와 CARMA 서버 사이의 통신을 위해 루프백 주소를 사용합니다. 이렇게 하면 보안이 향상되며(로컬 프로세스만 루프백 주소에 액세스 가능) 스택 연관 관계를 CARMA 통신에 추가할 필요가 없어질 수 있습니다.

crastart*.conf

다음 파트를

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

다음과 같이 바꾸십시오(여기서 TCP/IP는 원하는 TCP/IP 스택을 나타냄).

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

참고: CRASTART는 행 연속을 지원하지 않지만 허용 행 길이에 대한 한계는 없습니다.

CRASUB*

다음 파트를

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

다음과 같이 바꾸십시오(여기서 TCP/IP는 원하는 TCP/IP 스택을 나타냄).

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

참고: 작업 제출 시 행 길이 한계는 80자입니다. 행이 더 긴 경우 공백()으로 구분할 수 있으며 첫 번째 행 끝에서 더하기(+) 부호를 사용하여 두 행을 연결할 수 있습니다.

분산 동적 DVIPA

분산 DVIPA(동적 가상 IP 주소 지정)를 사용하여 sysplex의 여러 시스템에서 동일한 Developer for System z 설정을 동시에 실행하고, 선택적으로 WLM의 도움을 받아 TCP/IP를 통해 이 시스템 간에 클라이언트 연결을 분배할 수 있습니다.

분산 DVIPA를 구성할 수 있는 몇 가지 방법이 있지만 Developer for System z는 이러한 옵션에 몇 가지 제한을 둡니다.

- RSE 디먼은 분산 DVIPA에 대해 정의된 포트를 소유하지만 실제 작업은 다른 주소 공간에서 스레드로 활성화된 RSE 서버에서 발생합니다. 따라서 SERVERWLM 분배 방법을 사용하여 시스템에서 로드 밸런싱을 수행할 수 없습니다. 그 이유는 WLM이 RSE 서버가 아닌 RSE 디먼에 대한 통계를 기반으로 조언을 제공하기 때문입니다.
- 클라이언트는 RSE 디먼에 대해 Sysplex Distributor가 사용하는 DVIPA 주소만 알고 있습니다. Sysplex Distributor는 사용 가능한 디먼 중 하나로 연결 요청을 전달 하며, RSE 디먼은 해당 시스템의 포트에 바인드할 RSE 서버 스레드를 시작합니다. 클라이언트는 이 포트에 연결할 때 실제 시스템 주소가 아닌 DVIPA 주소를 다시 사용하므로 Sysplex Distributor가 새 연결 경로를 올바른 시스템으로 재지정하는지 확인해야 합니다.

따라서 Developer for System z는 RSE 서버 스레드가 사용하는 포트가 sysplex 내에서 고유한지 확인하려면 VIPADISTRIBUTE 문에 SYSPLEXPORTS 정의가 필요합니다.

참고:

- SYSPLEXPORTS 사용은 결합 기능에 EZBEPOR 구조가 정의되어야 함을 의미합니다.
- SYSPLEXPORTS 사용은 TCP/IP가 2차 연결을 위해 임시 포트를 선택함을 의미합니다. 이는 PORT 및 PORTRANGE 지시문을 사용하여 TCP/IP 프로파일에 이러한 연결을 위한 포트를 예약할 수 없음을 의미합니다. 또한 rsed.envvars의 _RSE_PORTRANGE를 사용하여 Developer for System z가 사용하는 포트를 제한할 수 없습니다. 이로 인해 방화벽 설정이 복잡해지기 때문에 Developer for System z는 이 제한사항에 대한 임시 해결책을 제공합니다.

분산 DVIPA 사용 시 Developer for System z 내에 몇 가지 제한사항이 있습니다.

- Developer for System z 클라이언트가 TCP/IP를 통한 올바른 포트 선택을 방해하지 않게 하려면 rsed.envvars에서 deny.nonzero.port 지시문을 사용할 수 있어야 합니다.
- 참여하는 모든 Developer for System z 서버의 설정이 동일해야 합니다. 참여하는 모든 시스템 간에 /usr/lpp/rdz 및 /etc/rdz를 공유해야 합니다. /var/rdz/projects, /var/rdz/pushtoclient, /var/rdz/sclmdt가 사용되는 경우 이 디렉토리도 공유해야 합니다. /var/rdz/WORKAREA 및 /var/rdz/logs는 각 시스템마다 고유해야 합니다.
- 공유해야 하는 Developer for System z 컴포넌트, 시스템별로 고유해야 하는 컴포넌트를 알려면 181 페이지의 제 11 장 『다중 인스턴스 실행』을 참조하십시오.

JES 작업 모니터, CARMA 및 기타 Developer for System z 서버는 오직 로컬 RSE와 상호작용하므로 DVIPA 설정이 필요하지 않습니다.

통합 디버거는 로컬 RSE와 상호작용하므로, DVIPA 설정이 필요하지 않습니다. 디버그 세션이 올바른 호스트와 통신하는지 확인하기 위해, 디버그 관리자는 반드시 사용해야 하는 TCP/IP 주소를 클라이언트에게 지시하므로 DVIPA 설정이 필요하지 않습니다.

분산 DVIPA는 TCP/IP 프로파일에 VIPADynamic 블록의 VIPADefine 및 VIPABackup 키워드를 사용하여 정의됩니다. VIPADISTribute 키워드는 필수 Sysplex Distributor 정의를 추가합니다. 분산 DVIPA에서는 참여하는 모든 스택을 sysplex에서 인식해야 하는데, 이는 TCP/IP 프로파일에서 IPCONFIG 블록의 SYSPLEXRouting 및 DYNAMICXCF 키워드를 통해 수행됩니다. 이러한 지시문에 대한 자세한 내용은 *Communications Server: IP Configuration Reference*(SC31-8776)를 참조하십시오.

결합 기능에 EZBEPOR 구조를 설정하는 방법에 대한 자세한 정보는 *MVS Setting Up a Sysplex*(SA22-7625) 및 *Communication Server: SNA Network Implementation Guide*(SC31-8777)를 참조하십시오.

포트 선택 제한

SYSplexPorts 사용은 TCP/IP가 2차 연결을 위해 임시 포트를 선택함을 의미합니다. 임시 포트는 사용할 수 있고 어떠한 방식으로도 예약되지 않은 포트입니다. 임시 포트 사용은 방화벽 우수 사례와 충돌하여 통신을 위해 열려 있는 포트를 제한합니다. 이는 사용할 포트를 알 수 없기 때문입니다.

이 문제점은 시스템당 고유 `_RSE_PORTRANGE`를 정의하고 사용된 포트 범위가 모든 시스템에서 Developer for System z 사용에 예약되었는지 확인하여 Developer for System z에서 2차 연결 시 알려진 포트를 강제로 사용하도록 함으로써 방지할 수 있습니다. 이 방법을 사용하려면 TCP/IP APAR PM63379가 필요합니다.

TCP/IP가 2차 연결을 올바른 시스템으로 라우트하도록 하려면 Developer for System z가 각 시스템에서 고유 포트 범위를 사용해야 합니다. 이는 `rsed.envvars`의 `_RSE_PORTRANGE`가 고유해야 하므로 시스템에 동일한 공유 설정을 사용할 수 없음을 의미합니다. 동일한 코드를 사용하는 동안 다른 구성 파일로 여러 서버를 설정하는 방법에 대한 정보는 181 페이지의 제 11 장 『다중 인스턴스 실행』의 182 페이지의 『동일한 소프트웨어 레벨, 다른 구성 파일』을 참조하십시오. `rsed.envvars`의 마스터 사본과 스크립트를 사용하여 시스템 특정 설정에 맞게 조정하고 복사하여 다른 시스템에서도 파일을 동일하게 유지해야 합니다.

1. Developer for System z가 단일 시스템 설정인 것처럼 SYS1에서 설정하되 `/usr/lpp/rdz`와 `/etc/rdz`가 공유 파일 시스템에 있어야 합니다. 모든 MVS 기반 파트는 SYS2와도 공유해야 합니다.
2. `/etc/rdz/rsed.envvars`를 마스터 사본으로 사용하고 파일 끝에 `/etc/rdz`에 대한 참조를 추가하여 시스템 특정 사본이 나머지 구성 파일을 선택할 수 있게 합니다.

```
$ 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. 마스터 `rsed.envvars`를 복사하고 `_RSE_PORTRANGE`를 조정하는 셸 스크립트인 `/etc/rdz/update.sh`를 작성합니다.

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

```

#!/bin/sh
# Licensed materials - Property of IBM
# 5724-T07 Copyright IBM Corp. 2012
# RDz 및 DDVIPA에 사용할 PORTRANGE 설정 및 rsed.envvars 복제

file=rsed.envvars          #; echo file $file
sys=${1:-$(sysvar SYSNAME)} #; echo sys $sys
dir=$(dirname $0)          #; echo dir $dir
# sysname에 특수 문자가 있는 경우, \를 앞에 둠(예. SYS\1)
case "$sys" in
    "SYS1") range=8108-8118;;
    "SYS2") range=8119-8129;;
esac
# ##### CUSTOMIZE THIS SECTION #####
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

```

그림 11. update.sh - 방화벽이 있는 DDVIPA 설정 지원

4. /etc/rdz/SYS1 및 /etc/rdz/SYS2 디렉토리를 작성하고 /etc/rdz/update.sh 를 실행하여 디렉토리를 채웁니다.

```

$ 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. RSED 시작 태스크가 /etc/rdz/&SYSNAME을 가리키는지 확인합니다.

```
// CNFG='/etc/rdz/&SYSNAME.'
```

다음으로, sysplex의 모든 시스템에서 Developer for System z에 대해 정의된 포트 범위가 예약되어 포트 번호가 sysplex 내에서 고유할 수 있도록 해야 합니다. 모든 시스템의 모든 범위를 예약하려면 PROFILE.TCPIP의 PORT 또는 PORTRANGE 문을 사용합니다. RSE 스레드 풀의 작업 이름은 RSEDx입니다. 여기서 RSED는 RSE 시작 태스크의 이름이고 x는 1자리 난수이므로 정의에서 와일드카드가 필요합니다.

```

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

```

9 페이지의 『연결 플로우』에서 설명하는 것처럼 _RSE_PORTRANGE의 포트 범위는 작습니다. RSE 서버는 클라이언트 연결 기간 동안에만 포트가 필요한 것은 아닙니다. 이는 다른 RSE 서버가 포트에 바인드할 수 없는 (서버) 바인드와 (클라이언트) 연결 사이의 기간일 뿐입니다. 이는 대부분의 연결이 범위에 있는 첫 번째 포트를 사용하고 나머지 범위는 다중 동시 로그온의 경우 버퍼가 됨을 의미합니다.

샘플 설정

다음 샘플 설정에는 두 가지 z/OS 시스템(SYS1, SYS2)이 있습니다. 이들 시스템은 sysplex의 일부입니다. System SYS1은 일반적으로 Developer for System z 분산 DVIPA의 Sysplex Distributor를 호스트하는 시스템으로 정의됩니다.

분산 DVIPA를 정의하면 Developer for System z를 시스템에서 시작하여 시스템 간 로드 밸런싱 클라이언트 연결을 허용할 수 있습니다. JES 작업 모니터는 오직 로컬 RSE와 상호작용하므로 DVIPA 설정이 필요하지 않습니다. 클라이언트는 IP 주소 10.10.10.1에서 포트 4035에 연결합니다.

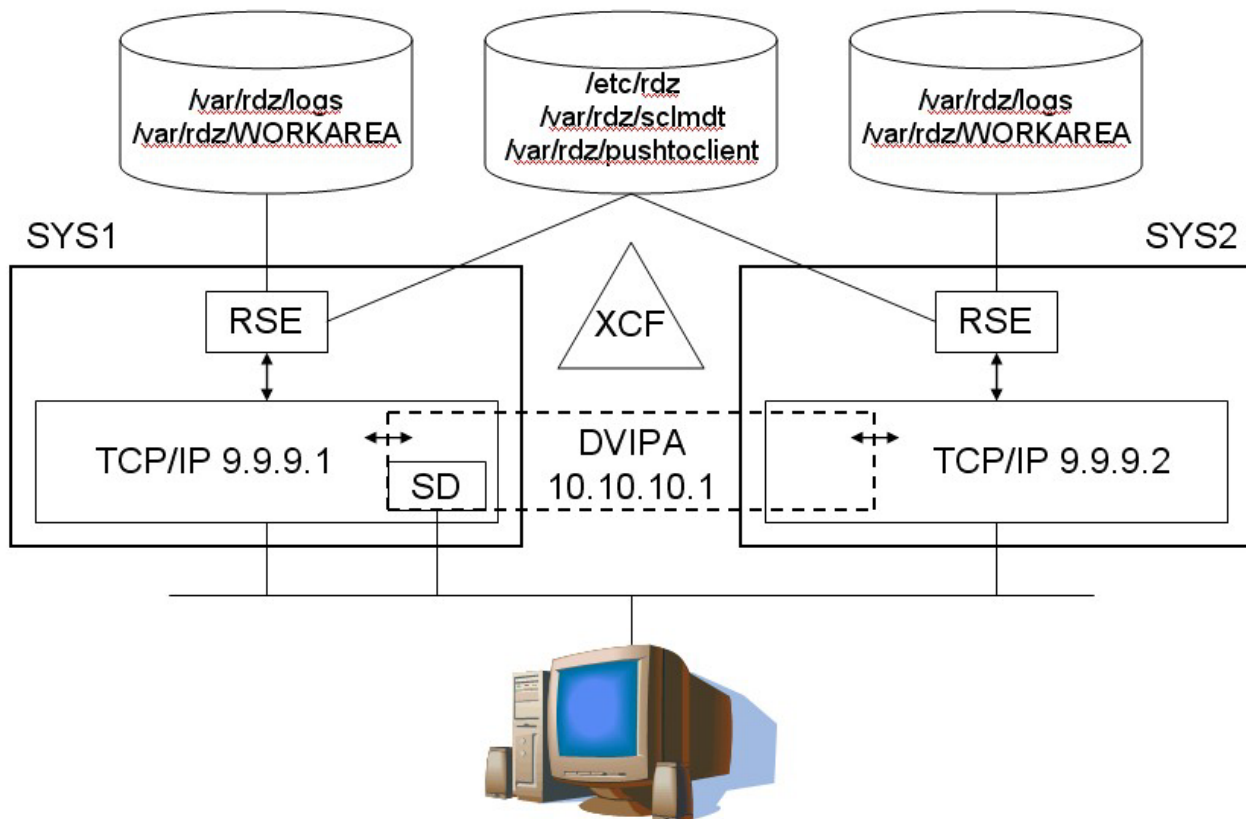


그림 12. 분산 동적 VIPA 샘플

시스템 SYS1 – TCP/IP 프로파일

```
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
  SYSPLEXPORTS ; 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
```

시스템 SYS2 – TCP/IP 프로파일

```
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
  SYSPLEXPORTS ; 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
```


제 4 장 WLM 고려사항

전통적인 z/OS 애플리케이션과 달리 Developer for System z는 워크로드 관리자(WLM)가 쉽게 식별할 수 있는 단일 애플리케이션이 아닙니다. Developer for System z는 상호 작용을 통해 호스트 서비스와 데이터에 대한 클라이언트 액세스 권한을 제공하는 여러 컴포넌트로 구성됩니다. 3 페이지의 제 1 장 『Developer for System z 이해』의 설명대로 이러한 서비스 중 일부는 다른 주소 공간에서 활성화되므로 WLM이 다르게 분류됩니다.

이 장에서 다루는 주제는 다음과 같습니다.

- 『워크로드 분류』
- 81 페이지의 『목표 설정』

워크로드 분류

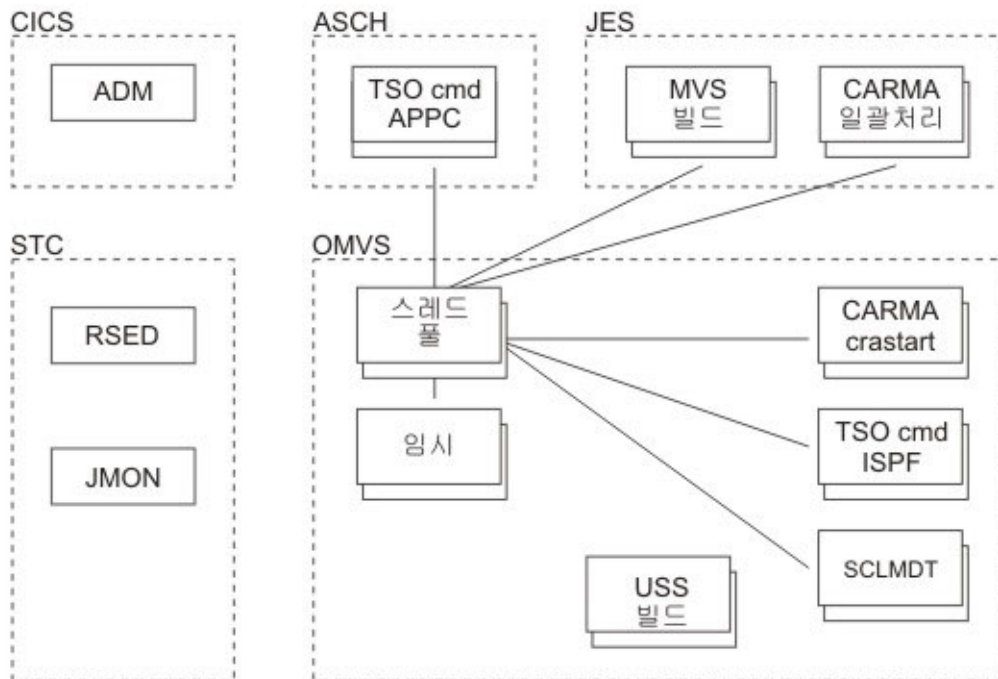


그림 13. WLM 분류

그림 13은 Developer for System z 워크로드를 WLM에 제공하는 서브시스템에 대한 기본 개요를 보여줍니다.

애플리케이션 배치 관리자(ADM)는 CICS 리전 내부에서 활성화되므로 WLM의 CICS 분류 규칙을 준수합니다.

RSE 디먼(RSED) 및 JES 작업 모니터(JMON)는 각각 개별 주소 공간을 갖는 Developer for System z 시작 태스크(또는 장기 실행 일괄처리 작업)입니다.

6 페이지의 『Java 애플리케이션으로서의 RSE』의 설명대로 RSE 디먼은 각 RSE 스레드 풀 서버(가변적인 클라이언트 수 지원)마다 하위 프로세스를 제공합니다. 각 스레드 풀은 z/OS UNIX 이니시에이터, BPXAS를 사용하는 개별 주소 공간에서 활성화됩니다. 이러한 프로세스는 제공된 프로세스이므로 시작 태스크 분류 규칙이 아닌 WLM OMVS 분류 규칙을 사용하여 분류됩니다.

스레드 풀에서 활성화되는 클라이언트는 사용자가 수행하는 조치에 따라 많은 다른 주소 공간을 작성할 수 있습니다. Developer for System z의 구성에 따라 TSO 명령 서비스(TSO cmd) 또는 CARMA와 같은 일부 워크로드를 다른 서브시스템에서 실행할 수 있습니다.

79 페이지의 그림 13에 나열된 주소 공간은 시스템에 오래 표시될 수 있지만 z/OS UNIX 디자인 방식으로 인해 여러 가지 단기 임시 공간도 존재합니다. 이러한 임시 주소 공간은 OMVS 서브시스템에서 활성화됩니다.

RSE 스레드 풀은 RSE 디먼과 동일한 사용자 ID와 유사한 작업 이름을 사용하지만 스레드 풀로 시작된 모든 주소 공간은 조치를 요청하는 클라이언트의 사용자 ID가 소유합니다. 클라이언트 사용자 ID는 또한 스레드 풀이 지정하는 모든 OMVS 기반 주소 공간의 작업 이름(또는 그 일부)으로 사용됩니다.

Developer for System z가 사용하는 기타 서비스(예: 파일 관리자(FMNCAS) 또는 z/OS UNIX REXEC(USS 빌드))로 주소 공간이 더 작성됩니다.

분류 규칙

WLM은 시스템 수신 작업을 서비스 클래스로 매핑하는 분류 규칙을 사용합니다. 이 분류는 작업 규정자를 기반으로 합니다. 첫 번째(필수) 규정자는 작업 요청을 수신하는 서브시스템 유형입니다. 표 13에는 Developer for System z 워크로드를 수신할 수 있는 서브시스템 유형이 나열되어 있습니다.

표 13. WLM 시작점 서브시스템

서브시스템 유형	작업 설명
ASCH	작업 요청에는 IBM 제공 APPC/MVS 트랜잭션 스케줄러 ASCH에서 예약하는 모든 APPC 트랜잭션 프로그램이 포함됩니다.
CICS	작업 요청에는 CICS가 처리하는 모든 트랜잭션이 포함됩니다.
JES	작업 요청에는 JES2 또는 JES3이 시작하는 모든 작업이 포함됩니다.
OMVS	작업 요청에는 z/OS UNIX 시스템 서비스 분류 하위 주소 공간에서 처리되는 작업이 포함됩니다.

표 13. WLM 시작점 서브시스템 (계속)

서브시스템 유형	작업 설명
STC	작업 요청에는 START 및 MOUNT 명령으로 시작되는 모든 작업이 포함됩니다. STC에는 또한 시스템 컴포넌트 주소 공간이 포함됩니다.

표 14에는 특정 서비스 클래스에 워크로드를 지정하는 데 사용할 수 있는 추가 규정자가 나열되어 있습니다. 나열된 작업 규정자에 대한 세부사항은 MVS Planning: Workload Management(SA22-7602)를 참조하십시오.

표 14. WLM 작업 규정자

		ASCH	CICS	JES	OMVS	STC
AI	계정 정보	x		x	x	x
LU	LU 이름(*)		x			
PF	수행(*)			x		x
PRI	우선순위			x		
SE	스케줄링 환경 이름			x		
SSC	서브시스템 콜렉션 이름			x		
SI	서브시스템 인스턴스(*)		x	x		
SPM	서브시스템 매개변수					x
PX	Sysplex 이름	x	x	x	x	x
SY	시스템 이름(*)	x			x	x
TC	트랜잭션/작업 클래스(*)	x		x		
TN	트랜잭션/작업 이름(*)	x	x	x	x	x
UI	사용자 ID(*)	x	x	x	x	x

참고: (*) 표시가 있는 규정자의 경우 유형 약어에 G를 추가하여 분류 그룹을 지정할 수 있습니다. 예를 들어, 트랜잭션 이름 그룹은 TNG입니다.

목표 설정

79 페이지의 『워크로드 분류』에서 설명하는 것처럼 Developer for System z는 시스템에 다양한 유형의 워크로드를 작성합니다. 이처럼 다른 태스크는 상호 통신을 수행합니다. 이는 태스크 간 연결에 대한 제한시간 문제를 방지하려면 실제 경과 시간이 중요함을 의미합니다. 결과적으로 Developer for System z 태스크는 고성능 서비스 클래스 또는 우선순위가 높은 중간 성능 서비스 클래스에 배치되어야 합니다.

따라서 현재 WLM 목표의 개정(일반적으로 업데이트)이 권장됩니다. 이는 특히 시간이 중요한 OMVS 워크로드에 익숙하지 않은 전통적인 MVS 작업장에 해당됩니다.

참고:

- 이 절의 목표 정보는 의도적으로 설명 레벨에서 보관됩니다. 실제 성능 목표는 사이트에 따라 큰 차이가 있기 때문입니다.

- 시스템에 대한 특정 태스크의 영향을 쉽게 이해할 수 있도록 최소, 중간, 대량 자원 사용량과 같은 용어를 사용합니다. 이러한 용어는 모두 전체 시스템이 아닌 Developer for System z 자체의 총 자원 사용량과 관련이 있습니다.

표 15에는 Developer for System z에서 사용하는 주소 공간이 나열되어 있습니다. z/OS UNIX는 "Task Name" 열의 "x"를 1자리 난수로 대체합니다.

표 15. WLM 워크로드

설명	태스크 이름	워크로드
JES 작업 모니터	JMON	STC
RSE 디먼	RSED	STC
RSE 스프레드 풀	RSEDx	OMVS
ISPF Client Gateway(TSO 명령 서비스와 SCLMDT)	<userid>x	OMVS
TSO 명령 서비스(APPC)	FEKFRSRV	ASCH
CARMA(일괄처리)	CRA<port>	JES
CARMA(crastart)	<userid>x	OMVS
CARMA (ISPF Client Gateway)	<userid> 및 <userid>x	OMVS
MVS 빌드(일괄처리 작업)	*	JES
z/OS UNIX 빌드(셸 명령)	<userid>x	OMVS
z/OS UNIX 셸	<userid>	OMVS
파일 관리자 태스크	<userid>x	OMVS
애플리케이션 배치 관리자	CICSTS	CICS

목표 선택 고려사항

다음 일반 WLM 고려사항은 Developer for System z에 대한 정확한 목표 정의를 올바르게 정의하는 데 도움을 줄 수 있습니다.

- 원하는 목표가 아닌 실제로 달성 가능한 목표를 설정해야 합니다. 필요한 것보다 목표를 높게 설정하면 WLM이 중요성이 낮은 작업의 자원을 자원이 실제로 필요하지 않은 중요성이 높은 작업으로 이동합니다.
- SYSTEM 및 SYSSTC 서비스 클래스에 지정된 작업의 양을 제한합니다. 이러한 클래스는 WLM 관리 클래스보다 디스패치 우선순위가 더 높기 때문입니다. CPU를 거의 사용하지 않는 중요한 작업에 이 클래스를 사용합니다.
- 분류 규칙을 준수하지 않는 작업은 임의 목표를 갖는 SYSOTHER 클래스에 속하게 됩니다. 임의 목표는 시스템에 여유 자원이 있는 경우 최대한 작업을 수행하도록 WLM에 지시합니다.

응답 시간 목표를 사용하는 경우:

- WLM이 응답 시간 목표를 올바르게 관리하려면 태스크 도달 속도가 일정해야 합니다(20분에 최소 10개 태스크).

- 평균 응답 시간 목표는 올바르게 제어된 워크로드에만 사용됩니다. 단일 트랜잭션이 오래 실행되면 평균 응답 시간에 큰 영향을 주고 WLM이 과잉 반응을 나타낼 수 있기 때문입니다.

속도 목표를 사용하는 경우:

- 다양한 이유로 인해 일반적으로 속도 목표를 90% 이상 달성할 수 없습니다. 예를 들어, 모든 SYSTEM 및 SYSSTC 주소 공간은 속도 유형 목표보다 디스패치 우선 순위가 더 높습니다.
- WLM은 속도 목표 의사결정의 기반이 되는 최소 (사용 및 지연) 샘플 수를 사용합니다. 따라서 서비스 클래스에서 실행되는 작업이 적을수록 필요한 샘플 수를 수집하고 디스패치 정책을 조정하는 데 시간이 더 소요됩니다.
- 하드웨어를 변경하는 경우에는 속도 목표를 다시 평가합니다. 특히 보다 적은 고속 프로세서를 이동하려면 속도 목표를 변경해야 합니다.

STC

모든 Developer for System z 시작 태스크, RSE 디먼 및 JES 작업 모니터는 실시간 클라이언트 요청을 서비스합니다.

표 16. WLM 워크로드 - STC

설명	태스크 이름	워크로드
JES 작업 모니터	JMON	STC
RSE 디먼	RSED	STC

- JES 작업 모니터

JES 작업 모니터는 작업 제출, 스푼 파일 찾아보기, JES 운영자 명령 실행과 같은 모든 JES 관련 서비스를 제공합니다. 태스크는 WLM에 개별 트랜잭션을 보고하지 않으므로 고성능, 단일 기간 속도 목표를 지정해야 합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준에서 중간 수준으로 예상됩니다.

- RSE 디먼

RSE 디먼은 클라이언트 로그인과 인증을 처리하고 다른 RSE 스레드 풀을 관리합니다. 태스크는 WLM에 개별 트랜잭션을 보고하지 않으므로 고성능, 단일 기간 속도 목표를 지정해야 합니다. 자원 사용량은 중간 수준으로 예상되며 작업일이 시작될 때 최대 수준입니다.

OMVS

OMVS 워크로드는 두 그룹(RSE 스레드 풀과 기타 그룹)으로 나눌 수 있습니다. 이는 RSE 스레드 풀을 제외한 모든 워크로드가 클라이언트 사용자 ID를 주소 공간 이름의 기반으로 사용하기 때문입니다. z/OS UNIX는 "Task Name" 열의 "x"를 1자리 난수로 대체합니다.

표 17. WLM 워크로드 - OMVS

설명	태스크 이름	워크로드
RSE 스레드 풀	RSEDx	OMVS
ISPF Client Gateway(TSO 명령 서비스와 SCLMDT)	<userid>x	OMVS
CARMA(crastart)	<userid>x	OMVS
CARMA (ISPF Client Gateway)	<userid> 및 <userid>x	OMVS
z/OS UNIX 빌드(셸 명령)	<userid>x	OMVS
z/OS UNIX 셸	<userid>	OMVS
파일 관리자 태스크	<userid>x	OMVS

- RSE 스레드 풀

RSE 스레드 풀은 Developer for System z의 심장, 두뇌와 같습니다. 거의 모든 데이터가 이 스레드 풀을 통과하며 스레드 풀 내부의 마이너(사용자별 스레드)가 대부분의 기타 Developer for System z 관련 태스크의 조치를 제어합니다. 태스크는 WLM에 개별 트랜잭션을 보고하지 않으므로 고성능, 단일 기간 속도 목표를 지정해야 합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 대량 수준으로 예상됩니다.

공통 주소 공간 이름 지정 규칙으로 인해 나머지 워크로드는 모두 동일한 서비스 클래스에서 종료됩니다. 이 서비스 클래스에는 다기간 목표를 지정해야 합니다. 첫 번째 기간은 고성능 백분위수 응답 시간 목표여야 하며 마지막 기간은 중간 성능 속도 목표를 가져야 합니다. ISPF Client Gateway와 같이 개별 트랜잭션을 WLM에 보고하는 워크로드도 있고 그렇지 않은 워크로드도 있습니다.

- ISPF Client Gateway

ISPF Client Gateway는 Developer for System z가 비대화식 TSO 및 ISPF 명령을 실행하기 위해 호출하는 ISPF 서비스입니다. 이러한 명령에는 클라이언트가 실행하는 명시적 명령과 Developer for System z가 실행하는 암시적 명령(예: PDS 멤버 목록 가져오기)이 포함됩니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준으로 예상됩니다.

- CARMA

CARMA는 호스트 기반 소프트웨어 구성 관리자(SCM)(예: CA Endeavor® SCM)와 상호작용하는 데 사용되는 선택적 Developer for System z 서버입니다. Developer

for System z는 CARMA 서버에 다른 시작 방법을 허용하며 그 중 일부는 OMVS 워크로드가 됩니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준으로 예상됩니다.

- z/OS UNIX 빌드

클라이언트가 z/OS UNIX 프로젝트에 대한 빌드를 시작하면 z/OS UNIX REXEC(또는 SSH)가 빌드를 수행하기 위해 여러 z/OS UNIX 셸 명령을 실행하는 태스크를 시작합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 프로젝트 크기에 따라 중간 수준에서 대량 수준으로 예상됩니다.

- z/OS UNIX 셸

이 워크로드는 클라이언트가 실행하는 z/OS UNIX 셸 명령을 처리합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준으로 예상됩니다.

- IBM 파일 관리자

Developer for System z 주소 공간은 아니지만 제공된 파일 관리자 하위 프로세스가 여기에 나열됩니다. 이러한 프로세스는 Developer for System z 클라이언트 요청 시 시작될 수 있으며 이러한 태스크는 Developer for System z 태스크와 동일한 이름 지정 규칙을 사용하기 때문입니다. 이러한 파일 관리자 태스크는 중요한 MVS 데이터 세트 조치(예: 형식화된 VSAM 파일 편집)를 처리합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준에서 중간 수준으로 예상됩니다.

JES

JES 관리 일괄처리 프로세스는 Developer for System z에서 다양한 방법으로 사용됩니다. 가장 일반적인 사용법은 MVS 빌드에 대한 것으로 제출된 작업을 모니터링하여 종료 시점을 결정합니다. 그러나 Developer for System z는 또한 CARMA 서버를 일괄 처리로 시작하고 TCP/IP를 사용하여 통신할 수 있습니다.

표 18. WLM 워크로드 - JES

설명	태스크 이름	워크로드
CARMA(일괄처리)	CRA<port>	JES
MVS 빌드(일괄처리 작업)	*	JES

- CARMA

CARMA는 호스트 기반 소프트웨어 구성 관리자(SCM)(예: CA Endevor® SCM)와 상호작용하는 데 사용되는 선택적 Developer for System z 서버입니다. Developer for System z는 CARMA 서버에 다른 시작 방법을 허용하며 그 중 일부는 JES 워크로드가 됩니다. 태스크는 WLM에 개별 트랜잭션을 보고하지 않으므로 고성능, 단

일 기간 속도 목표를 지정해야 합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준으로 예상됩니다.

- MVS 빌드

클라이언트가 MVS 프로젝트에 대한 빌드를 시작하면 Developer for System z가 빌드를 수행하는 일괄처리 작업을 시작합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 프로젝트 크기에 따라 중간 수준에서 대량 수준으로 예상됩니다. 해당 지역 상황에 따라 다른 중간 성능 목표 전략이 권장됩니다.

- 백분위수 응답 기간과 추적 속도 기간으로 다기간 목표를 지정할 수 있습니다. 이러한 경우 개발자는 일반적으로 동일한 빌드 프로시저와 유사한 크기의 입력 파일을 사용하여 단일 응답 시간 작업을 작성해야 합니다. WLM이 응답 시간 목표를 올바르게 관리하려면 작업 도달 속도가 일정해야 합니다(20분에 최소 10개 작업).
- 속도 목표는 매우 가변적인 실행 시간과 도달 속도를 처리할 수 있으므로 대부분의 일괄처리 작업에 가장 적합합니다.

ASCH

현재 Developer for System z 버전에서는 비대화식 TSO 및 ISPF 명령을 실행하기 위해 ISPF Client Gateway를 사용합니다. 이전 사례와 관련된 이유로 인해 Developer for System z는 또한 APPC 트랜잭션을 통한 이러한 명령 실행을 지원합니다. APPC 방법은 더 이상 사용되지 않습니다.

표 19. WLM 위크로드 - ASCH

설명	태스크 이름	위크로드
TSO 명령 서비스(APPC)	FEKFRSRV	ASCH

- TSO 명령 서비스

TSO 명령 서비스는 Developer for System z가 비대화식 TSO 및 ISPF 명령을 실행하기 위해 APPC 트랜잭션으로 시작할 수 있습니다. 이러한 명령에는 클라이언트가 실행하는 명시적 명령과 Developer for System z가 실행하는 암시적 명령(예: PDS 멤버 목록 가져오기)이 포함됩니다. 이 서비스 클래스에는 다기간 목표를 지정해야 합니다. 첫 번째 기간의 경우 고성능, 백분위수 응답 시간 목표를 지정해야 합니다. 마지막 기간의 경우 중간 성능 속도 목표를 지정해야 합니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준으로 예상됩니다.

CICS

애플리케이션 배치 관리자는 CICS 트랜잭션 서버 리전 내부에서 활성화되는 선택적 Developer for System z 서버입니다.

표 20. WLM 워크로드 - CICS

설명	태스크 이름	워크로드
애플리케이션 배치 관리자	CICSTS	CICS

- 애플리케이션 배치 관리자

CICSTS 리전 내부에서 활성화되는 선택적 애플리케이션 배치 관리자 서버를 사용하면 선택한 CICSTS 관리 태스크를 개발자에게 안전하게 전달할 수 있습니다. 자원 사용량은 사용자 조치에 따라 크게 달라지므로 변동 폭이 크지만 최소 수준으로 예상됩니다. 사용해야 하는 서비스 클래스 유형은 이 CICS 리전에서 활성화되는 다른 트랜잭션에 따라 다르므로 자세히 설명하지 않습니다.

WLM은 CICS에 사용할 수 있는 여러 가지 관리 유형을 지원합니다.

- 리전 목표에 대한 CICS 관리

목표는 CICS 주소 공간을 관리하는 서비스 클래스로 설정됩니다. 이 서비스 클래스에는 실행 속도 목표만 사용할 수 있습니다. WLM은 주소 공간에 JES 또는 STC 분류 규칙을 사용하지만 트랜잭션에는 CICS 서브시스템 분류 규칙을 사용하지 않습니다.

- 트랜잭션 응답 시간 목표에 대한 CICS 관리

단일 트랜잭션 또는 트랜잭션 그룹에 지정된 서비스 클래스에서 응답 시간 목표를 설정할 수 있습니다. WLM은 주소 공간과 트랜잭션에 각각 JES 또는 STC 분류 규칙과 CICS 서브시스템 분류 규칙을 사용합니다.

제 5 장 튜닝 고려사항

3 페이지의 제 1 장 『Developer for System z 이해』에 설명된 대로, RSE(Remote Systems Explorer)는 Developer for System z의 핵심입니다. 클라이언트로부터의 연결과 워크로드를 관리하기 위해 RSE는 스레드 풀링 주소 공간을 제어하는 디먼 주소 공간으로 구성됩니다. 디먼은 연결 및 관리를 위한 포컬 포인트의 역할을 하는 반면, 스레드 풀은 클라이언트 워크로드를 처리합니다.

따라서 RSE는 Developer for System z 설정 조정의 주요 대상이 됩니다. 그러나 각각 17개 이상의 스레드, 일정 양의 스토리지, 하나 이상의 주소 공간을 사용하는 수백 명의 사용자를 유지보수하려면 Developer for System z와 z/OS를 둘 다 올바르게 구성해야 합니다.

이 장에서 다루는 주제는 다음과 같습니다.

- 『자원 사용량』
- 101 페이지의 『스토리지 사용량』
- 107 페이지의 『z/OS UNIX 파일 시스템 공간 사용량』
- 110 페이지의 『키 자원 정의』
- 114 페이지의 『다양한 자원 정의』
- 116 페이지의 『모니터링』
- 120 페이지의 『샘플 설정』

자원 사용량

이 절의 정보를 사용하면 Developer for System z의 일반 및 최대 자원 사용량을 예측하고 그에 따라 시스템 구성을 계획할 수 있습니다.

이 절에 나와 있는 숫자와 공식을 사용하여 시스템 한계 값을 정의하는 경우, 반드시 정확한 예측값으로 작업해야 합니다. 시스템 한계를 설정할 때는 임시 태스크와 기타 태스크 또는 호스트에 여러 번 연결하는 사용자의 자원 사용을 허용할 수 있는 충분한 여백을 남겨두어야 합니다. (예를 들어, RSE 및 TN3270을 통해).

참고:

- 정보는 Developer for System z가 제공하는 RSE를 통해 액세스하는 서비스 범위로 제한됩니다. 예를 들어, TN3270의 자원 사용량이 기록되지 않거나(RSE를 통해 액세스 하지 않음) MVS 또는 z/OS UNIX 프로젝트의 원격(호스트 기반) 빌드 중에 프로그램의 자원 사용량이 호출되지 않습니다(Developer for System z에서 제공되지 않음).

- Developer for System z에 씨드파티 확장기능을 추가하면 자원 사용량 카운터를 늘릴 수 있습니다.
- 모든 서비스에는 실행 중에 자원을 사용하고 순차적으로 또는 서로 병렬로 실행될 수 있는 단기 "하우스키핑" 태스크가 있습니다. 이러한 태스크가 사용하는 자원은 기록되지 않습니다.
- 필요한 경우 ISPF Client Gateway와 같은 필수 소프트웨어의 사용자별 자원 사용량이 기록됩니다.
- 여기에 나타난 숫자는 사전 통지 없이 변경될 수 있습니다.

개요

다음 표는 Developer for System z에서 사용하는 주소 공간, 프로세스, 스레드의 수에 대한 개요 정보를 제공합니다. 표에 나와 있는 수치에 대한 세부사항은 다음 절에 나와 있습니다.

- 91 페이지의 『주소 공간 계수』
- 94 페이지의 『프로세스 개수』
- 97 페이지의 『스레드 개수』

표 21은 Developer for System z 시작 태스크에서 사용하는 주요 자원에 대한 일반 개요 정보를 제공합니다. 이러한 자원은 한 번만 할당됩니다. 모든 Developer for System z 클라이언트가 이러한 자원을 공유합니다.

표 21. 일반 자원 사용법

시작 태스크	주소 공간	프로세스	스레드
JMON	1	1	3
RSED	1	3	11
RSEDx	1 + 1 (a)	1 + 2	1 + 10

참고: (a) 하나의 APF 권한 주소 공간과 하나 이상의 RSE 스레드 풀 주소 공간이 활성화됩니다. 실제 RSE 스레드 풀 주소 공간 수를 결정하려면 91 페이지의 『주소 공간 계수』를 참조하십시오.

표 22은 필수 소프트웨어가 사용하는 주요 자원에 대한 일반 개요 정보를 제공합니다. 이러한 자원은 관련 함수를 호출하는 Developer for System z 클라이언트마다 할당됩니다.

표 22. 사용자별 필수 자원 사용량

필수 소프트웨어	주소 공간	프로세스	스레드
ISPF Client Gateway	1	2	4
APPC	1	1	2

표 23는 지정된 함수를 실행할 때 각 Developer for System z 클라이언트가 사용하는 주요 자원에 대한 일반 개요 정보를 제공합니다. ISPF와 같이 숫자가 아닌 값은 90 페이지의 표 22의 해당 값에 대한 참조입니다.

표 23. 사용자별 자원 사용량

사용자 조치	주소 공간	프로세스	스레드		
	사용자 ID	사용자 ID	사용자 ID	RSEDx	JMON
로그온	-	-	-	17	1
유휴 제한시간 타이머	-	-	-	1	-
검색	-	-	-	1	-
확장 PDS(E)	ISPF	ISPF	ISPF	-	-
데이터 세트 열기	ISPF	ISPF	ISPF	1	-
TSO 명령	ISPF	ISPF	ISPF	-	-
z/OS UNIX 셸	1	1	1	6	-
MVS 빌드	1	-	-	-	-
z/OS UNIX 빌드	3	3	3	-	-
CARMA(일괄 처리)	1	1	2	1	-
CARMA(crastart)	1	1	2	4	-
CARMA(ispf)	4	4	7	5	-
SCLMDT	ISPF	ISPF	ISPF	-	-

참고: ISPF는 APPC로 대체될 수 있습니다(SCLM 개발자 툴킷 제외).

주소 공간 계수

표 24은 Developer for System z에서 사용하는 주소 공간을 나열합니다. 여기서 “계수” 열의 “u”는 해당 크기에 기능을 사용하는 동시 활성 사용자 수를 곱해야 함을 표시합니다. z/OS UNIX에서는 “태스크 이름” 열의 “x”를 임의의 1자리 숫자로 대체합니다.

표 24. 주소 공간 계수

계수	설명	태스크 이름	공유	다음 이후에 종료
1	JES 작업 모니터	JMON	예	종료되지 않음
1	RSE 디먼	RSED	예	종료되지 않음
1	RSE APF 권한 부여됨	RSEDx	예	종료되지 않음
(a)	RSE 스레드 풀	RSEDx	예	종료되지 않음
lu	ISPF Client Gateway(TSO 명령 서비스와 SCLMDT)	<userid>x	아니오	15분 또는 사용자 로그오프
lu	TSO 명령 서비스(APPC)	FEKFRSRV	아니오	60분 또는 사용자 로그오프

표 24. 주소 공간 계수 (계속)

계수	설명	태스크 이름	공유	다음 이후에 종료
1u	CARMA(일괄처리)	CRA<port>	아니오	7분 또는 사용자 로그오프
1u	CARMA(crastart)	<userid>x	아니오	7분 또는 사용자 로그오프
4u	CARMA(ispf, 더 이상 사용되지 않음)	(1)<userid> 또는 (3)<userid>x	아니오	7분 또는 사용자 로그오프
(b)	1명의 사용자에게 의한 동시 ISPF Client Gateway 사용	<userid>x	아니오	태스크 완료
1u	MVS 빌드(일괄처리 작업)	*	아니오	태스크 완료
3u	z/OS UNIX 빌드(셸 명령)	<userid>x	아니오	태스크 완료
1u	z/OS UNIX 셸	<userid>	아니오	사용자 로그오프

참고:

- (a) 하나 이상의 RSE 스레드 풀 주소 공간이 활성화되어 있습니다. 실제 숫자는 다음에 따라 다릅니다.
 - rsed.envvars의 minimum.threadpool.process 지시문. 기본값은 1입니다.
 - 하나의 스레드 풀이 서비스할 수 있는 사용자 수. 기본 설정은 스레드 풀당 30명의 사용자를 목표로 합니다.

참고: single.logon 지시문이 활성화된 경우 minimum.threadpool.process가 1로 설정된 경우에도 2개 이상의 스레드 풀이 시작됩니다. rsed.envvars의 single.logon 기본 설정은 활성화입니다.

- (b) Developer for System z에 사용자당 여러 개의 스레드가 활성화되어 있습니다. 다른 스레드가 요청을 보낼 때 ISPF Client Gateway 주소 공간이 한 스레드의 요청에 대한 서비스 제공을 완료하지 않은 경우 ISPF는 새 Client Gateway를 시작하여 새 요청을 처리합니다. 이 주소 공간은 태스크 완료 후에 종료됩니다.
- SCLMDT에는 ISPF Client Gateway 주소 공간이 필요합니다. SCLMDT는 TSO 명령 서비스와 주소 공간을 공유합니다.
- 대부분의 MVS 데이터 세트 관련 조치는 TSO 명령 서비스를 사용합니다. 이 서비스는 ISPF Client Gateway 또는 APPC 트랜잭션에서 각각 활성화될 수 있습니다.

그림 14의 공식을 사용하면 Developer for System z에서 사용하는 최대 주소 공간 수를 예측할 수 있습니다.

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

그림 14. 최대 주소 공간 수

여기서

- “3”은 영구 활성 서버 주소 공간 수입니다.
- “A”는 RSE 스레드 풀 주소 공간 수를 나타냅니다.

- “N”은 최대 동시 사용자 수를 나타냅니다.
- “x”는 선택한 구성 옵션에 따라 다음 값 중 하나입니다.

X	SCLMDT	TSO(클라이언트 게이트웨이 사용)	TSO(APPC 사용)
1	아니오	아니오	예
1	아니오	예	아니오
1	예	예	아니오

- “y”는 선택한 구성 옵션에 따라 다음 값 중 하나입니다.

Y	
0	CARMA 없음
1	CARMA(일괄처리)
1	CARMA(crastart)
4	CARMA(ispf, 더 이상 사용되지 않음)

- “z”는 기본적으로 0이지만 사용자 조치에 따라 증가할 수 있습니다.
 - MVS 빌드가 수행되는 경우 1을 추가하십시오. 이 주소 공간은 관련 빌드 태스크(일괄처리 작업)가 완료되면 종료됩니다.
 - z/OS UNIX 빌드가 수행되는 경우 3을 추가하십시오. 실제 수치는 호출된 프로그램의 요구에 따라 더 클 수 있습니다. 이 주소 공간은 관련 빌드 태스크가 완료되면 종료됩니다.
- “2 + N*0.01”은 임시 주소 공간에 대한 버퍼를 추가합니다. 필수 버퍼 크기는 사용자의 사이트에서 다를 수 있습니다.

그림 15의 공식을 사용하면 Developer for System z 클라이언트에서 사용하는 최대 주소 공간 수를 예측할 수 있습니다(설명하지 않은 임시 주소 공간은 계산하지 않음).

$$x + y + z$$

그림 15. 클라이언트당 주소 공간 수

여기서

- “x”는 선택한 구성 옵션에 따라 다르며 최대 주소 공간 수를 계산하는 공식(92 페이지의 그림 14)에 대해 설명되어 있습니다.
- “y”는 선택한 구성 옵션에 따라 다르며 최대 주소 공간 수를 계산하는 공식(92 페이지의 그림 14)에 대해 설명되어 있습니다.
- “z”는 기본적으로 0이지만 최대 주소 공간 수를 계산하는 공식(92 페이지의 그림 14)에 대해 설명된 대로 사용자 조치에 따라 증가할 수 있습니다.

94 페이지의 표 25의 정의는 실제 주소 공간 수를 제한할 수 있습니다.

표 25. 주소 공간 한계

위치	한계	영향을 받는 자원
rsed.envvars	maximum.threadpool.process	RSE 스레드 풀 수 제한
IEASYMxx	MAXUSER	주소 공간 수 제한
ASCHPMxx	MAX	TSO 명령 서비스(APPC)에 대한 APPC 개시자 수 제한

프로세스 개수

표 26에는 Developer for System z에서 사용하는 주소 공간당 프로세스 수가 나열되어 있습니다. “주소 공간” 열의 “u”는 해당 크기에 기능을 사용하는 동시 활성 사용자 수를 곱해야 함을 표시합니다.

표 26. 프로세스 개수

프로세스	주소 공간	설명	사용자 ID
1	1	JES 작업 모니터	STCJMON
3	1	RSE 디먼	STCRSE
1	1	RSE APF 인증	STCRSE
2	(a)	RSE 스레드 풀	STCRSE
2	(b)	ISPF Client Gateway(TSO 명령 서비스와 SCLMDT)	<userid>
1	1u	TSO 명령 서비스(APPC)	<userid>
1	1u	CARMA(일괄처리)	<userid>
1	1u	CARMA(crastart)	<userid>
1	1u	CARMA(ispf, 더 이상 사용되지 않음)	<userid>
1	3u	z/OS UNIX 빌드(셸 명령)	<userid>
1	1u	z/OS UNIX 셸	<userid>
(5)	(u)	SCLM 개발자 툴킷	<userid>

참고:

- (a) 하나 이상의 RSE 스레드 풀 주소 공간이 활성화됩니다. 실제 RSE 스레드 풀 주소 공간 수를 결정하려면 91 페이지의 『주소 공간 계수』를 참조하십시오.
- RSE 디먼과 모든 RSE 스레드 풀은 동일한 사용자 ID를 사용합니다.
- (b) 일반적으로 또한 기본 구성 옵션을 사용하는 경우에는 사용자당 하나의 ISPF Client Gateway가 활성화됩니다. 실제 수는 다를 수 있습니다(91 페이지의 『주소 공간 계수』의 설명 참조).
- SCLMDT에는 ISPF Client Gateway 주소 공간이 필요합니다. SCLMDT는 TSO 명령 서비스와 주소 공간을 공유합니다.
- (u) SCLMDT 프로세스는 ISPF Client Gateway 주소 공간에서 실행되므로 주소 공간 개수 값이 없습니다.

- SCLMDT 프로세스는 임시 프로세스로 태스크 완료 시 종료되지만 단일 사용자에게 대해 동시에 여러 프로세스가 활성화될 수 있습니다. 94 페이지의 표 26에는 최대 동시 SCLMDT 프로세스 수가 나열되어 있습니다.
- 대부분의 MVS 데이터 세트 관련 조치는 TSO 명령 서비스를 사용합니다. 이 서비스는 ISPF Client Gateway 또는 APPC 트랜잭션에서 각각 활성화될 수 있습니다.
- z/OS UNIX 빌드는 각각 해당 주소 공간에서 실행되는 총 3가지 프로세스를 사용합니다.
- 나열된 모든 프로세스는 다른 표시가 없는 한 관련 주소 공간이 종료될 때까지 활성 상태를 유지합니다.

그림 16의 공식을 사용하면 Developer for System z에서 사용하는 최대 프로세스 수를 예측할 수 있습니다.

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

그림 16. 최대 프로세스 수

여기서

- “5”는 영구 활성 서버 주소 공간에서 사용하는 프로세스 수입니다.
- “A”는 RSE 스레드 풀 주소 공간 수를 나타냅니다.
- “N”은 최대 동시 사용자 수를 나타냅니다.
- “x”는 선택한 구성 옵션에 따라 다음 값 중 하나입니다.

X	SCLMDT	TSO(클라이언트 게이트웨이 사용)	TSO(APPC 사용)
1	아니오	아니오	예
2	아니오	예	아니오
7	예	예	아니오

- “y”는 선택한 구성 옵션에 따라 다음 값 중 하나입니다.

Y	
0	CARMA 없음
1	CARMA(일괄처리)
1	CARMA(crastart)
4	CARMA(ispf, 더 이상 사용되지 않음)

- “z”는 기본적으로 0이지만 사용자 조치에 따라 증가할 수 있습니다.
 - z/OS UNIX 셸이 열리면 1을 더합니다. 이 프로세스는 사용자가 로그오프할 때까지 활성 상태를 유지합니다.

- z/OS UNIX 빌드가 수행되는 경우 3을 추가하십시오. 실제 수치는 호출된 프로그램의 요구에 따라 더 클 수 있습니다. 이러한 프로세스는 관련 빌드 태스크가 완료되면 종료됩니다.

- "10 + N*0.05"는 임시 프로세스에 대한 버퍼를 추가합니다. 필수 버퍼 크기는 사용자의 사이트에서 다를 수 있습니다.

그림 17의 공식을 사용하여 RSED 시작 태스크 사용자 ID, STCRSE에서 사용하는 최대 프로세스 수를 예측할 수 있습니다(표시되지 않은 임시 프로세스는 세지 않음).

$$4 + 2 * A$$

그림 17. STCRSE의 프로세스 수

여기서

- "4"는 RSE 디먼 및 RSE APF 권한 부여 주소 공간에서 사용하는 프로세스 수와 같습니다.
- "A"는 RSE 스프레드 폴 주소 공간의 수를 나타냅니다.

그림 18의 공식을 사용하면 Developer for System z 클라이언트에서 사용하는 최대 프로세스 수를 예측할 수 있습니다(설명하지 않은 임시 프로세스는 계산하지 않음).

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

그림 18. 클라이언트당 프로세스 수

여기서

- "x"는 선택한 구성 옵션에 따라 다르며 최대 프로세스 수를 계산하는 공식을 위해 기록됩니다(95 페이지의 그림 16).
- "y"는 선택한 구성 옵션에 따라 다르며 최대 프로세스 수를 계산하는 공식을 위해 기록됩니다(95 페이지의 그림 16).
- "z"은 기본적으로 0이지만 사용자 조치에 따라 증가할 수 있으며 선택한 구성 옵션에 따라 다르며 최대 프로세스 수를 계산하는 공식을 위해 기록됩니다(95 페이지의 그림 16).
- "s"는 SCLM 개발자 툴킷을 사용하는 경우에는 1, 그렇지 않은 경우에는 0입니다.

97 페이지의 표 27의 정의는 실제 프로세스 수를 제한할 수 있습니다.

표 27. 프로세스 한계

위치	한계	영향을 받는 자원
BPXPRMxx	MAXPROCSYS	총 프로세스 수를 제한합니다.
BPXPRMxx	MAXPROCUSER	z/OS UNIX UID당 프로세스 수를 제한합니다.

참고:

- RSE 디먼과 RSE 스레드 풀은 동일한 사용자 ID를 사용합니다. 필요할 때마다 RSE 디먼이 새 스레드 풀을 시작하므로 이 사용자 ID의 프로세스 수가 증가할 수 있습니다. 따라서 해당 증가를 수용할 수 있도록 MAXPROCUSER를 설정해야 합니다. 해당 공식은 $3 + 2 \times A$ 입니다.
- MAXPROCUSER 한계는 고유 z/OS UNIX 사용자 ID(UID)에 따라 다릅니다. 사용자가 동일한 UID를 공유하는 경우에는 예상 사용자당 프로세스 개수에 동시 활성 클라이언트 수를 곱합니다.

스레드 개수

표 28에는 선택한 Developer for System z 기능이 사용하는 스레드 수가 나열되어 있습니다. "스레드" 열의 "u"는 해당 크기에 기능을 사용하는 동시 활성 사용자의 수를 곱해야 함을 나타냅니다. 이 레벨에는 한계가 설정되어 있으므로 스레드 개수는 프로세스당 기준으로 나열됩니다.

- RSEDx: 이러한 스레드는 여러 클라이언트가 공유하는 RSE 스레드 풀에서 작성됩니다. 동일한 스레드 풀에 속하는 모든 스레드를 함께 더해야 총 개수를 얻을 수 있습니다.
- 활성: 이러한 스레드는 요청된 기능을 실제로 수행하는 프로세스의 일부입니다. 각 프로세스는 독립형 단위이므로 다른 표시가 없는 한 동일한 사용자 ID에 지정되더라도 스레드 개수의 합계를 구하지 않아도 됩니다.
- 부트스트랩: 실제 프로세스를 시작하려면 부트스트랩 프로세스가 필요합니다. 각각 하나의 스레드가 있으므로 여러 연속 부트스트랩이 존재할 수 있습니다. 스레드 개수의 합계는 구하지 않아도 됩니다.

표 28. 스레드 개수

스레드			사용자 ID	설명
RSEDx	활성	부트스트랩		
-	$3 + 1u$	-	STCJMON	JES 작업 모니터
-	15	2	STCRSE	RSE 디먼
-	1	-	STCRSE	RSE APF 권한 부여됨
$10(a) + 17u$	-	$1(a)$	STCRSE	RSE 스레드 풀
-	$4u(b)$	$1u(b)$	<userid>	ISPF Client Gateway(TSO 명령 서비스와 SCLMDT)

표 28. 스레드 개수 (계속)

스레드			사용자 ID	설명
-	2u	-	<userid>	TSO 명령 서비스 (APPC)
1u	2u	-	STCRSE 및 <userid>	CARMA(일괄처리)
4u	2u	-	STCRSE 및 <userid>	CARMA(crastart)
5u	4u	3u	STCRSE 및 <userid>	CARMA(ispf, 더 이상 사용되지 않음)
-	1u (c)	2u	<userid>	z/OS UNIX 빌드(셸 명령)
6u	1u	-	STCRSE 및 <userid>	z/OS UNIX 셸
1 (d)	-	-	STCRSE	다운로드
1 (e)	-	-	STCRSE	검색
-	(5)	-	<userid>	SCLM 개발자 툴킷
1u	-	-	STCRSE	유희 제한시간 타이머

참고:

- (a) 하나 이상의 RSE 스레드 풀 주소 공간이 활성화됩니다. 실제 RSE 스레드 풀 주소 공간 수를 결정하려면 91 페이지의 『주소 공간 계수』를 참조하십시오.
- (b) 일반적으로 또한 기본 구성 옵션을 사용하는 경우에는 사용자당 하나의 ISPF Client Gateway가 활성화됩니다. 실제 수는 다를 수 있습니다(91 페이지의 『주소 공간 계수』의 설명 참조).
- SCLMDT에는 ISPF Client Gateway 주소 공간이 필요합니다. SCLMDT는 TSO 명령 서비스와 주소 공간을 공유합니다.
- SCLMDT는 선택한 조치에 따라 태스크 완료 시 종료되는 여러 단일 스레드 프로세스를 사용할 수 있습니다. 97 페이지의 표 28에는 최대 동시 SCLMDT 스레드 수가 나열되어 있습니다.
- 대부분의 MVS 데이터 세트 관련 조치는 TSO 명령 서비스를 사용합니다. 이 서비스는 ISPF Client Gateway 또는 APPC 트랜잭션에서 각각 활성화될 수 있습니다.
- (c) z/OS UNIX 빌드는 멀리 스레드 구성이 가능한 다른 빌드 유틸리티를 호출합니다. 97 페이지의 표 28에는 최소 동시 z/OS UNIX 빌드 스레드 수가 나열되어 있습니다.
- (d) 각 호스트 데이터 다운로드에는 별도의 스레드를 사용합니다. 이 스레드는 데이터가 클라이언트로 전송되면 종료됩니다.
- (e) 각 원격 검색은 별도의 스레드를 사용합니다. 이 스레드는 결과가 클라이언트로 전송되면 종료됩니다.

- 나열된 모든 스레드는 다른 표시가 없는 한 관련 프로세스가 종료될 때까지 활성 상태를 유지합니다.
- RSE APF 권한 코드에 대한 일반 스레드 개수는 1입니다. 그러나 시작 중에는 일시적으로 13개 이상의 동시 스레드가 활성화됩니다.

그림 19의 공식을 사용하면 RSE 스레드 풀이 사용하는 최대 스레드 수를 예측할 수 있습니다. 그림 20의 공식을 사용하면 JES 작업 모니터가 사용하는 최대 스레드 수를 예측할 수 있습니다.

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

그림 19. 최대 RSE 스레드 풀 스레드 수

$$3 + N$$

그림 20. 최대 JES 작업 모니터 스레드 수

여기서

- "N" 은 이 스레드 풀 또는 JES 작업 모니터의 최대 동시 사용자 수를 나타냅니다. 기본 설정은 스레드 풀당 30명의 사용자를 목표로 합니다.
- "x"는 선택한 구성 옵션에 따라 다음 값 중 하나입니다.

X	SCLMDT	TSO(클라이언트 게이트웨이 사용)	TSO(APPC 사용)	제한시간
0	아니오	아니오	예	아니오
0	아니오	예	아니오	아니오
0	예	예	아니오	아니오
1	아니오	아니오	예	예
1	아니오	예	아니오	예
1	예	예	아니오	예

- "y"는 선택한 구성 옵션에 따라 다음 값 중 하나입니다.

Y	
0	CARMA 없음
1	CARMA(일괄처리)
4	CARMA(crastart)
5	CARMA(ispf, 더 이상 사용되지 않음)

- "z"는 기본적으로 0이지만 사용자 조치에 따라 증가할 수 있습니다.

- z/OS UNIX 셸이 열리면 6을 더합니다. 이러한 스레드는 사용자가 로그오프할 때까지 활성 상태를 유지합니다.
- "20 + N*0.1"은 임시 스레드에 대한 버퍼를 추가합니다. 필수 버퍼 크기는 사용자의 사이트에서 다를 수 있습니다. 여러 동시 다운로드 및 검색은 사용자가 이 버퍼 크기를 늘려야 할 수 있는 2가지 예제입니다.

표 29의 정의는 일반적으로 RSE 스레드 풀에 중요한 프로세스 내 실제 스레드 수를 제한합니다.

표 29. 스레드 한계

위치	한계	영향을 받는 자원
BPXPRMxx	MAXTHREADS	프로세스 내 스레드 수를 제한합니다.
BPXPRMxx	MAXTHREADTASKS	프로세스 내 MVS 태스크 수를 제한합니다.
BPXPRMxx	MAXASSIZE	주소 공간 크기와, 스레드 관련 제어 블록에 사용 가능한 스토리지를 제한합니다.
rsed.envvars	Xmx	최대 Java 힙 크기를 설정합니다. 이 스토리지는 예약되므로 스레드 관련 제어 블록에 더 이상 사용할 수 없습니다.
rsed.envvars	maximum.clients	RSE 스레드 풀의 클라이언트(또한 해당 스레드) 수를 제한합니다.
rsed.envvars	maximum.threads	RSE 스레드 풀의 클라이언트 스레드 수를 제한합니다.
FEJJCNFG	MAX_THREADS	JES 작업 모니터의 스레드 수를 제한합니다.

참고:

- rsed.envvars의 maximum.threads 값은 BPXPRMxx의 MAXTHREADS 및 MAXTHREADTASKS 값보다 작아야 합니다.
- **DISPLAY PROCESS,CPU** 연산자 명령은 스레드 풀에서 활성 스레드를 표시하는데, 이는 첫 번째 4000개의 스레드만 표시하도록 제한됩니다.

임시 자원 사용량

앞의 절에서 설명한 자원 사용량은 Developer for System z의 수명 동안 영구적이거나 특정 사용자별 태스크에 반영구적입니다.

그러나 Developer for System z는 하우스키핑 태스크를 위해 또한 다음 요청을 충족하기 위해 일시적으로 추가 자원을 사용합니다.

- 감사 파일 이벤트(rsed.envvars의 audit.action 지시문) 처리는 하나의 추가 스레드, 하나의 추가 프로세스 및 가능한 경우(audit.action.id가 설정된 경우) 하나의 추가 주소 공간을 사용합니다.
- 로그인 이벤트(rsed.envvars)의 logon.action 지시문) 처리는 하나의 추가 스레드, 하나의 추가 프로세스 및 가능한 경우(logon.action.id가 설정된 경우) 하나의 추가 주소 공간을 사용합니다.
- 연산자 명령 IVP PASSTICKET은 두 개 추가 스레드를 사용합니다.

- 연산자 명령 IVP DAEMON은 하나의 추가 스레드, 하나의 추가 프로세스, 하나의 추가 주소 공간을 사용합니다.
- 연산자 명령 IVP ISPF는 하나의 추가 스레드, 하나의 추가 프로세스, 하나의 추가 주소 공간과 함께 ISPF Client Gateway가 사용하는 자원을 더 사용합니다.

스토리지 사용량

RSE는 Java 애플리케이션입니다. 따라서 Developer for System z에 대한 스토리지(메모리) 사용량 계획에서는 두 가지 스토리지 할당 한계(Java 힙 크기와 주소 공간 크기)를 고려해야 합니다.

Java 힙 크기 한계

Java는 Java 애플리케이션에 대한 코딩 노력을 줄일 수 있는 많은 서비스를 제공합니다. 이러한 서비스 중 하나가 스토리지 관리입니다.

Java의 스토리지 관리는 대형 스토리지 블록을 할당하고 애플리케이션의 스토리지 요청에 해당 블록을 사용합니다. Java가 관리하는 이 스토리지를 Java 힙이라고 합니다. 주기적인 가비지 콜렉션(조각 모음)은 힙에서 사용하지 않은 공간을 회수하여 크기를 줄입니다. CPU 사이클을 저장하기 위해, 사용된 스토리지가 실제로 필요할 때까지 가비지 콜렉션이 대기하여 절대적으로 필요한 것보다 오래 할당되어 사용되지 않는 스토리지를 남깁니다.

최대 Java 힙 크기는 Xmx 지시문으로 rsed.envvars에 정의됩니다. 이 지시문이 지정되지 않는 경우 Java는 기본 크기 512MB를 사용합니다. 256MB보다 큰 값을 지정해야 합니다. 64비트 모드로 실행 시 Java가 2GB 막대 위에 힙을 할당하려고 하여 막대 아래 공간을 해제합니다.

각 RSE 스레드 풀(클라이언트 조치 서비스)은 개별 Java 애플리케이션이므로 개인 Java 힙을 갖습니다. 모든 스레드 풀은 동일한 rsed.envvars 구성 파일을 사용하므로 Java 힙 크기 한계도 같습니다.

Java 힙의 스레드 풀 사용량은 연결된 클라이언트가 수행하는 조치에 따라 다릅니다. 최적의 힙 크기 한계를 설정하려면 힙 사용량을 주기적으로 모니터링해야 합니다. **modify display process** 연산자 명령을 사용하면 RSE 스레드 풀의 Java 힙 사용량을 모니터링할 수 있습니다.

주소 공간 크기 한계

Java 애플리케이션을 포함하는 모든 z/OS 애플리케이션은 주소 공간 내에서 활성화되며 주소 공간 크기 제한사항이 적용됩니다.

원하는 주소 공간 크기는 시작 중에 예를 들어, JCL의 REGION 매개변수로 지정됩니다. 그러나 시스템 설정이 실제 주소 공간 크기를 제한할 수 있습니다. 이러한 한계에 대해 알아보려면 207 페이지의 『주소 공간 크기』를 참조하십시오.

- SYS1.PARMLIB(BPXPRMxx)의 MAXASSIZE
- 시작 태스크에 지정된 사용자 ID의 OMVS 세그먼트에 있는 ASSIZEMAX
- 시스템 종료 IEFUSI 및 IEALIMIT
- 64비트 주소 지정 모드의 경우 SYS1.PARMLIB(SMFPRMxx)의 MEMLIMIT

RSE 스레드 풀은 RSE 디먼의 주소 공간 크기 한계를 상속합니다. 주소 공간 크기는 Java 힙, Java 자체, 공통 스토리지 영역, 시스템이 스레드 풀 활동(예: 스레드당 태스크 제어 블록(TCB))을 지원하기 위해 작성하는 모든 제어 블록을 수용할 수 있을 만큼 충분해야 합니다. 이 스토리지 사용량 중 일부는 16MB 행 미만입니다. 64비트 모드로 실행 시 Java가 2GB 막대 위에 힙을 할당하려고 하여 막대 아래 공간을 해제합니다.

영향을 주는 설정(예를 들어, Java 힙의 크기 또는 단일 스레드 풀이 지원하는 사용자의 양)을 변경하기 전에 실제 주소 공간 크기를 모니터링해야 합니다. 일반 시스템 모니터링 소프트웨어를 사용하여 Developer for system z의 실제 스토리지 사용량을 추적할 수 있습니다. 전용 모니터링 도구가 없는 경우에는 SDSF DA 보기 또는 TASID(ISPF "지원 및 다운로드" 웹 페이지에서 사용 가능한 현재(as-is) 시스템 정보 도구)와 같은 도구로 기본 정보를 수집할 수 있습니다.

크기 예측 가이드라인

앞에서 설명한 것처럼 Developer for system z의 실제 스토리지 사용량은 사용자 활동에 큰 영향을 받습니다. 조치에 따라 고정 스토리지 크기(예: 로그인)를 사용하기도 하고 가변 스토리지 크기(예: 지정된 상위 레벨 규정자로 데이터 세트 나열)를 사용하기도 합니다.

- RSE에는 Java 힙과 모든 시스템 제어 블록의 공간을 확보할 수 있도록 2GB의 주소 공간을 사용합니다.
- 64비트 모드로 실행 시 2GB 막대 위의 스토리지가 RSE에 실제로 사용 가능한지 확인하십시오.
- 주소 공간 크기 한계를 설정할 수 있는 위치에 대해 자세히 알아보려면 207 페이지의 『주소 공간 크기』의 내용을 참조하십시오.
- 샘플 rsed.envvars 설정은 스레드 풀당 30명의 사용자를 목표로 합니다.
 - maximum.clients=30
 - maximum.threads=520($10+17*30 = 520$, 따라서 520은 30개의 클라이언트를 허용함)

- 샘플 `rsed.envvars` 설정에서는 Java 힙이 최대 512MB까지 증가할 수 있습니다. 따라서 30개 클라이언트가 클라이언트당 평균 17MB를 사용할 수 있습니다($30 \times 17 = 510$).

RSE는 콘솔 메시지 FEK004I에 현재 Java 힙 및 주소 공간 크기 한계를 표시합니다.

모니터링 결과 현재 Java 힙 크기가 실제 워크로드에 충분하지 않은 것으로 나타나면 다음 시나리오 중 하나를 사용합니다.

- `rsed.envvars`의 `Xmx` 지시문으로 최대 Java 힙 크기를 늘립니다. 이를 수행하기 전에 주소 공간에 크기 증가를 수용할 수 있는 공간이 있는지 확인합니다.
- `rsed.envvars`의 `maximum.clients` 지시문으로 스레드 풀당 최대 클라이언트 수를 줄입니다. RSE는 동일한 수의 클라이언트를 계속 지원하지만 보다 많은 스레드 풀에 클라이언트가 분배됩니다.

참조로, 표 30는 실제 Developer for System z 고객이 스토리지 사용량에 영향을 주는 주요 `rsed.envvars` 설정이 사용하는 값을 보여줍니다.

표 30. 스토리지 사용량에 대한 참조 설정

mxmx(최대 Java 힙)	maximum.clients	기본 개발 유형
512M	30	PL/I
512M	10	COBOL
384M	12	COBOL
800M(64비트)	20	지정되지 않음

샘플 스토리지 사용량 분석

다음 그림에 표시된 내용은 기본 Developer for System z 설정을 다음과 같이 수정한 경우의 샘플 자원 사용량 수치를 보여줍니다.

- RSE에서 2개 이상의 스레드 풀 주소 공간을 작성하지 못하도록 `single.logon`을 사용하지 않습니다.
- 최대 Java 힙 크기는 10MB로 설정됩니다. 최대값이 작은 경우 백분위수 사용량이 커져 힙 크기 한계에 쉽게 도달하기 때문입니다.

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

그림 21. 로그인 수가 5인 경우 자원 사용량

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

그림 22. 로그인 수가 5인 경우 자원 사용량(계속)

104 페이지의 그림 21과 그림 22는 5개 클라이언트가 Java 힙이 10MB인 RSE 디먼에 로그인하는 시나리오를 보여줍니다.

- 스레드 풀(RSED8)은 시작 시 휴면 상태로, 약 27MB를 사용하며 그 중 0.7MB(10MB의 7%)는 Java에 해당됩니다.
- 첫 번째 클라이언트가 연결되면 스레드 풀이 활성화되며 추가로 27MB와 함께 연결되는 각 클라이언트마다 2MB를 사용합니다.
- 이 연결당 2MB의 일부는 Java 힙에 있으며 이는 힙 사용량 증가로 알 수 있습니다.
- 그러나 힙 사용량은 필요한 스토리지를 예측하고 필요한 것보다 많은 양을 할당하는 Java 메커니즘에 따라 다르므로 실제 패턴은 없습니다. 간헐적인 가비지 콜렉션으로 여유 스토리지를 확보하므로 동향을 감지하기가 훨씬 더 어렵습니다.
- 활성 스레드에 대한 충분한 힙 크기를 확보하기 위해 스레드 풀당 연결 수를 제한하는 내부 메커니즘은 새 스레드 풀(RSED9)에서 다섯 번째 연결을 작성합니다. 올바르게 구성된 설정을 사용할 때는 일반적으로 이러한 내부 안전망이 호출되지 않습니다. 다른 한계에 먼저 도달하기 때문입니다(대부분의 경우 `rsed.envvars의 maximum.clients`).

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

그림 23. PDS 멤버 편집 시 자원 사용량

106 페이지의 그림 23은 하나의 클라이언트가 Java 힙이 10MB인 RSE 디면에 로그인하고 PDS 멤버를 편집하는 시나리오를 보여줍니다.

- 195개 데이터 세트 이름을 얻는 카탈로그 검색은 시스템 활동으로 인해 모두 약 2MB의 스토리지를 사용했습니다. Java 힙 사용량이 증가하지 않기 때문입니다.
- 21 멤버 PDS를 여는 경우에는 스레드 풀에서 메모리를 거의 사용하지 않지만 디스플레이에는 TSO 명령 서비스가 호출된 것으로 표시됩니다. TSO에서 이 사용자 ID에 지정된 리전 크기를 사용하는 새 활성 주소 공간(IBMUSER2)이 있습니다. 이 주소 공간은 지정된 시간 동안 활성 상태를 유지하므로 TSO 명령 서비스의 향후 요청에 따라 재사용될 수 있습니다.
- 멤버를 열면 상위 레벨 규정자 확장에 따라 유사한 숫자가 표시됩니다. Java 힙 사용량은 동일하게 유지되지만 시스템 활동으로 인해 6.5MB의 스토리지가 증가합니다.

z/OS UNIX 파일 시스템 공간 사용량

DD문에 기록되지 않은 대부분의 Developer for System z 관련 데이터는 z/OS UNIX 파일에서 종료됩니다. 시스템 프로그래머는 기록되는 데이터와 저장 위치를 제어합니다. 그러나 기록되는 데이터의 양은 제어하지 않습니다.

데이터는 다음 카테고리 분류할 수 있습니다.

- 문제점 분석(로그 및 시스템 덤프 파일)(세부사항은 189 페이지의 제 12 장 『구성 문제점 해결』의 설명 참조)
- 감사(26 페이지의 『감사 로깅』의 설명 참조)
- 클라이언트로 푸시 메타데이터(137 페이지의 『클라이언트로 푸시 메타데이터』의 설명 참조)
- 임시 데이터

189 페이지의 제 12 장 『구성 문제점 해결』의 설명대로 Developer for System z는 RSE 관련 호스트 로그를 다음 z/OS UNIX 디렉토리에 기록합니다.

- /var/rdz/logs(RSE 시작 태스크 로그)
- /var/rdz/logs/\$LOGNAME(사용자 로그)

기본적으로 오류 및 경고 메시지만 로그에 기록됩니다. 따라서 모든 프로세스가 계획대로 진행되는 경우 이러한 디렉토리에는 비어 있거나 거의 비어 있는 파일만 보관됩니다(감사 로그는 세지 않음).

IBM 지원 센터의 지시에 따라 정보 메시지 로깅을 사용할 수 있으며 이 경우 로그 파일의 크기가 현저히 증가합니다.

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
```

그림 24. z/OS UNIX 파일 시스템 공간 사용량

그림 24은 디버그 레벨 2(정보 메시지)를 사용하는 경우 최소 z/OS UNIX 파일 시스템 공간 사용량을 보여줍니다.

- 시작 태스크 로그는 시작 후 34KB를 사용하며 사용자가 로그인, 로그오프 또는 연산자 명령을 실행할 때 천천히 증가합니다.

- 클라이언트 로그 디렉토리는 로그인 후 11KB를 사용하며 사용자가 작업을 시작하면 점진적으로 증가합니다(샘플에는 표시되지 않음).
- 로그오프는 사용자 로그에 40KB를 더 추가하여 51KB가 됩니다.

감사 로그를 제외하고, 다시 시작(RSE 시작 태스크) 또는 로그인(클라이언트)할 때마다 로그 파일을 겹쳐쓰고 총 크기를 계속 확인합니다. `rsed.envvars`의 `keep.last.log` 지시문은 RSE에 이전 로그의 복사본을 유지하도록 지시할 수 있으므로 이를 약간 변경합니다. 이전 복사본은 항상 제거됩니다.

감사 로그 파일을 보유하는 파일 시스템이 여유 공간에서 많이 실행되지 않고 감사가 활성화되면 콘솔로 경고 메시지를 보냅니다. 이 콘솔 메시지(FEK103E)는 공간 부족 문제가 해결될 때까지 주기적으로 반복됩니다. RSE에서 생성되는 콘솔 메시지의 목록은 *Host Configuration Guide* (SC23-7658)의 "콘솔 메시지"를 참조하십시오.

표 31의 정의는 로그 디렉토리에 기록되는 데이터와 디렉토리 위치를 제어합니다.

표 31. 로그 출력 지시문

위치	지시문	기능
<code>resecomm.properties</code>	<code>debug_level</code>	기본 로그 세부사항 레벨을 설정합니다.
<code>rsed.envvars</code>	<code>keep.last.log</code>	시작/로그온 전에 이전 로그의 사본을 보존합니다.
<code>rsed.envvars</code>	<code>enable.audit.log</code>	클라이언트 조치의 감사 추적을 보존합니다.
<code>rsed.envvars</code>	<code>enable.standard.log</code>	하나 이상의 스프레드 폴의 <code>stdout</code> 및 <code>stderr</code> 스트림을 로그 파일에 기록합니다.
<code>rsed.envvars</code>	<code>DSTORE_TRACING_ON</code>	데이터 저장소 조치 로깅을 사용합니다.
<code>rsed.envvars</code>	<code>DSTORE_MEMLOGGING_ON</code>	데이터 저장소 메모리 사용량 로깅을 사용합니다.
연산자 명령	<code>modify rsecommlog <level></code>	<code>rsecomm.log</code> 의 로그 세부사항 레벨을 동적으로 변경합니다.
연산자 명령	<code>modify rsedaemonlog <level></code>	<code>rsedaemon.log</code> 의 로그 세부사항 레벨을 동적으로 변경합니다.
연산자 명령	<code>modify rseserverlog <level></code>	<code>rseserver.log</code> 의 로그 세부사항 레벨을 동적으로 변경합니다.
연산자 명령	<code>modify rsestandardlog {onloff}</code>	<code>std*.log</code> 업데이트를 동적으로 변경합니다.
<code>rsed.envvars</code>	<code>daemon.log</code>	RSE 시작 태스크와 감사 로그의 홈 경로
<code>rsed.envvars</code>	<code>user.log</code>	사용자 로그의 홈 경로
<code>rsed.envvars</code>	<code>CGI_ISPWORK</code>	ISPF Client Gateway 로그의 홈 경로
<code>rsed.envvars</code>	<code>TMPDIR</code>	IVP 로그 디렉토리
<code>rsed.envvars</code>	<code>_CEE_DMPTARG</code>	Java 덤프 디렉토리

Developer for System z는 ISPF Client Gateway와 같은 필수 소프트웨어와 함께 /tmp 및 /var/rdz/WORKAREA에도 임시 데이터를 기록합니다. 사용자 조치의 결과로 여기에 기록되는 데이터의 양은 예측할 수 없으므로 이러한 디렉토리를 보유할 수 있는 큰 여유 공간이 파일 시스템에서 필요합니다.

Developer for System z는 항상 이러한 임시 파일을 정리하려고 시도하지만 언제라도 수동 정리(*Host Configuration Guide* (SC23-7658)의 "(선택사항) WORKAREA 및 /tmp 정리" 설명 참조)를 수행할 수 있습니다.

표 32의 정의는 임시 데이터 디렉토리의 위치를 제어합니다.

표 32. 임시 출력 지시문

위치	지시문	기능
rsed.envvars	CGI_ISPWORK	임시 데이터의 홈 경로
rsed.envvars	TMPDIR	임시 데이터 디렉토리

키 자원 정의

/etc/rdz/rsed.envvars

rsed.envvars에 정의된 환경 변수는 RSE, Java 및 z/OS UNIX에서 사용합니다. Developer for System z와 함께 제공되는 샘플 파일은 Developer for System z의 선택적 컴포넌트가 필요하지 않은 중소 규모 설치를 대상으로 합니다. *Host Configuration Guide* (SC23-7658)의 "rsed.envvars, RSE 구성 파일"에서는 샘플 파일에 정의된 각 변수에 대해 설명합니다. 샘플 파일의 다음 변수에는 특별한 주의가 필요합니다.

_RSE_JAVA_OPTS="\$_RSE_JAVA_OPTS -Xms128m -Xmx512m"

초기(Xms) 및 최대(Xmx) 힙 크기를 설정합니다. 기본값은 각각 128M과 512M입니다. 원하는 힙 크기 값을 강제 실행하도록 변경합니다. 이 지시문이 주석 처리되면 Java 기본값이 사용됩니다. 기본값은 각각 4M과 512M입니다.

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

하나의 스레드 풀이 서비스하는 최대 클라이언트 수. 기본값은 30입니다. 주석을 해제하고 스레드 풀당 클라이언트 수를 제한하도록 사용자 정의합니다. 다른 한계로 인해 RSE가 이 한계에 도달하지 못할 수 있습니다.

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

새 클라이언트를 허용할 수 있는 한 스레드 풀의 최대 활성 스레드 수. 기본값은 520입니다. 주석을 해제하고 사용 중인 스레드 수에 따라 스레드 풀당 클라이언트 수를 제한하도록 사용자 정의합니다. 각 클라이언트 연결마다 여러 스레드(17개 이상)를 사용하고 다른 한계로 인해 RSE가 이 한계에 도달하지 못할 수도 있습니다.

참고: 이 값은 SYS1.PARMLIB(BPXPRMxx)에서 MAXTHREADS 및 MAXTHREADTASKS에 대한 설정보다 작아야 합니다.

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

최대 활성 스레드 풀 수. 기본값은 1입니다. 주석을 해제하고 최소한 나열된 스레드 풀 프로세스를 시작하도록 사용자 정의합니다. 스레드 풀 프로세스는 RSE 서버 스레드의 로드 밸런싱에 사용됩니다. 필요한 경우 새 프로세스가 더 시작됩니다. 새 프로세스를 미리 시작하면 연결 지연을 방지할 수 있지만 유휴 시간에 보다 많은 자원을 사용합니다.

참고: single.logon 지시문을 활성화되면 minimum.threadpool.process가 1로 설정되더라도 최소 2개 스레드 풀이 시작됩니다. rsed.envvars에서 ingle.logon의 기본 설정은 활성화입니다.

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

최대 활성 스레드 풀 수. 기본값은 100입니다. 주석을 해제하고 스레드 풀 프로세스 수를 제한하도록 사용자 정의합니다. 스레드 풀 프로세스는 RSE 서버 스레드의 로드 밸런싱에 사용되므로 프로세스 수를 제한하면 활성 클라이언트 연결 수가 제한됩니다.

SYS1.PARMLIB(BPXPRMxx)

RSE는 Java 애플리케이션이며 이는 z/OS UNIX 환경에서 활성화됨을 의미합니다. 이 경우 BPXPRMxx는 z/OS UNIX 환경과 파일 시스템을 제어하는 매개변수를 포함하므로 중요 parmlib 멤버가 됩니다. BPXPRMxx에 대한 설명은 *MVS Initialization and Tuning Reference*(SA22-7592)에 설명되어 있습니다. 다음 지시문은 Developer for System z에 영향을 주는 것으로 알려져 있습니다.

MAXPROCSYS(nnnnn)

시스템이 허용하는 최대 프로세스 수를 지정합니다.

값 범위: nnnnn은 5에서 32767까지의 10진수 값입니다.

기본값: 900

MAXPROCUSER(nnnnn)

단일 z/OS UNIX 사용자 ID가 프로세스 작성 방법에 관계 없이 동시에 활성화할 수 있는 최대 프로세스 수를 지정합니다.

값 범위: nnnnn은 3에서 32767까지의 10진수 값입니다.

기본값: 25

참고:

- 모든 클라이언트가 RSE 프로세스 내에서 스레드로 실행되므로 모든 RSE 프로세스는 동일한 z/OS UNIX 사용자 ID(RSE 디먼에 지정된 사용자의 ID)를 사용합니다.

- 이 값은 RSED 시작 태스크에 지정된 사용자의 OMVS 보안 프로파일 세그먼트에서 PROCUSERMAX 변수로도 설정할 수 있습니다.

MAXTHREADS(nnnnnn)

단일 프로세스에서 동시에 활성화할 수 있는 최대 pthread_created 스레드 수(실행 중, 큐 대기, 종료했지만 발견되지 않은 스레드 포함)를 지정합니다. 값 0을 지정하면 애플리케이션이 pthread_create를 사용하지 않습니다.

값 범위: nnnnnn은 0에서 100000까지의 10진수 값입니다.

기본값: 200

참고:

- 각 클라이언트는 RSE 스레드 풀 프로세스 내에서 17개 이상의 스레드를 사용하므로 프로세스 내에서 여러 클라이언트가 활성화됩니다.
- 이 값은 RSED 시작 태스크에 지정된 사용자의 OMVS 보안 프로파일 세그먼트에서 THREADSMAX 변수로도 설정할 수 있습니다. THREADSMAX 값이 설정되면 MAXTHREADS와 MAXTHREADTASKS에 모두 사용됩니다.

MAXTHREADTASKS(nnnnn)

단일 프로세스에서 pthread_created 스레드에 대해 동시에 활성화할 수 있는 최대 MVS 태스크 수를 지정합니다.

값 범위: nnnnn은 0에서 32768까지의 10진수 값입니다.

기본값: 1000

참고:

- 각 활성 스레드마다 MVS (태스크 제어 블록(TCB)) 태스크가 있습니다.
- 각각의 동시 MVS 태스크에는 추가 스토리지가 필요하며 그 중 일부는 16MB 행 미만이어야 합니다.
- 각 클라이언트는 RSE 스레드 풀 프로세스 내에서 17개 이상의 스레드를 사용하므로 프로세스 내에서 여러 클라이언트가 활성화됩니다.
- 이 값은 RSED 시작 태스크에 지정된 사용자의 OMVS 보안 프로파일 세그먼트에서 THREADSMAX 변수로도 설정할 수 있습니다. THREADSMAX 값이 설정되면 MAXTHREADS와 MAXTHREADTASKS에 모두 사용됩니다.

MAXUIDS(nnnnn)

동시에 실행할 수 있는 최대 z/OS UNIX 사용자 ID(UID) 수를 지정합니다.

값 범위: nnnnn은 1에서 32767까지의 10진수 값입니다.

기본값: 200

MAXASSIZE(nnnnn)

새 프로세스의 초기값으로 설정될 RLIMIT_AS 자원 값을 지정합니다. RLIMIT_AS 는 주소 공간 리전 크기를 나타냅니다.

값 범위: nnnnn은 10485760(10MB)에서 2147483647(2GB)까지의 10진수 값입니다.

기본값: 209715200(200메가바이트)

참고:

- 이 값은 2G로 설정해야 합니다.
- 이 값은 RSED 시작 태스크에 지정된 사용자의 OMVS 보안 프로파일 세그먼트에서 ASSIZEMAX 변수로도 설정할 수 있습니다.

MAXFILEPROC(nnnnnn)

단일 프로세스에서 동시에 활성화 또는 할당할 수 있는 파일, 소켓, 디렉토리, 기타 파일 시스템 오브젝트의 최대 디스크립터 수를 지정합니다.

값 범위: nnnnnn은 3에서 524287까지의 10진수 값입니다.

기본값: 64000

참고:

- 스레드 풀의 모든 클라이언트 스레드는 단일 프로세스에 존재합니다.
- 이 값은 RSED 시작 태스크에 지정된 사용자의 OMVS 보안 프로파일 세그먼트에서 FILEPROCMAX 변수로도 설정할 수 있습니다.

MAXMAPAREA(nnnnn)

z/OS UNIX 파일의 메모리 맵핑을 위해 할당될 수 있는 최대 데이터 공간 스토리지 공간 크기(페이지)를 지정합니다. 스토리지는 메모리 맵핑이 활성 상태인 경우에만 할당됩니다.

값 범위: nnnnn은 1에서 16777216까지의 10진수 값입니다.

기본값: 40960

참고: 이 값은 RSED 시작 태스크에 지정된 사용자의 OMVS 보안 프로파일 세그먼트에서 MMAPAREAMAX 변수로도 설정할 수 있습니다.

SETOMVS 또는 **SET OMVS** 연산자 명령을 사용하면 다음 IPL까지 이전 BPXPRMxx 변수의 값을 동적으로 늘리거나 줄일 수 있습니다. 영구적으로 변경하려면 IPL에 사용될 BPXPRMxx 멤버를 편집합니다. 이러한 연산자 명령에 대한 자세한 정보는 *MVS System Commands*(SA22-7627)를 참조하십시오.

다음 정의는 NETWORK 문의 하위 매개변수입니다.

MAXSOCKETS(nnnnnnnn)

이 파일 시스템이 이 주소 패밀리에 지원하는 최대 소켓 수를 지정합니다. 이 매개 변수는 선택적입니다.

값 범위: nnnnnnnn은 0에서 16777215까지의 10진수 값입니다.

기본값: 100

INADDRANYCOUNT(nnnn)

시스템이 PORT 0, INADDR_ANY 바인드에 대한 사용을 위해 예약하는 포트의 수를 지정합니다(INADDRANYPORT 매개변수에 지정된 포트 번호로 시작). 이 값은 CINET(여러 TCP/IP 스택)에만 필요합니다.

값 범위: nnnn은 1에서 4000까지의 10진수 값입니다.

기본값: INADDRANYPORT 또는 INADDRANYCOUNT가 모두

지정되지 않는 경우 INADDRANYCOUNT의 기본값은 1000입니다.

그렇지 않은 경우에는 포트가 예약되지 않습니다(0).

다양한 자원 정의

서버 JCL의 EXEC 카드

Developer for System z 서버의 JCL에서 EXEC 카드에 다음 정의를 추가하는 것이 좋습니다.

REGION=0M

RSE 디먼과 JES 작업 모니터 시작 태스크, RSED와 JMON에 각각 REGION=0M이 권장됩니다. 이렇게 하면 주소 공간 크기가 사용 가능한 개인용 스토리지만큼 제한되며 그렇지 않으면 IEFUSI 또는 IEALIMIT 시스템이 종료됩니다. IBM에서는 RSE 디먼처럼 z/OS UNIX 주소 공간에 이러한 종료를 사용하지 않도록 권장합니다.

TIME=NOLIMIT

TIME=NOLIMIT는 모든 Developer for System z 서버에 사용하는 것이 좋습니다. 이는 모든 Developer for System z 클라이언트의 CPU 시간이 서버 주소 공간에 누적되기 때문입니다.

FEK.#CUST.PARMLIB(FEJJCNFG)

FEJJCNFG에 정의된 환경 변수는 JES 작업 모니터가 사용합니다. Developer for System z와 함께 제공되는 샘플 파일은 중소 규모 설치를 대상으로 합니다. *Host Configuration Guide* (SC23-7658)의 "FEJJCNFG, JES 작업 모니터 구성 파일"는 샘플 파일에 정의된 각 변수에 대해 설명합니다. 이 파일에서는 다음 변수에 유의해야 합니다.

MAX_THREADS

한 번에 하나의 JES 작업 모니터를 사용할 수 있는 최대 사용자 수. 기본값은 200입니다. 최대값은 2147483647입니다. 이 숫자를 늘리면 JES 작업 모니터 주소 공간의 크기를 늘려야 할 수 있습니다.

SYS1.PARMLIB(IEASYSxx)

IEASYSxx에는 시스템 매개변수가 보관되며 *MVS Initialization and Tuning Reference* (SA22-7592)에 설명되어 있습니다. 다음 지시문은 Developer for System z에 영향을 주는 것으로 알려져 있습니다.

MAXUSER=nnnnn

이 매개변수는 일반적으로 시스템이 지정된 IPL에서 동시에 실행될 수 있는 작업과 시작 태스크의 수를 제한하기 위해 사용하는 값을 지정합니다.

값 범위: nnnnn은 0에서 32767까지의 10진수 값입니다. MAXUSER, RSVSTRT 및 RSVNONR 시스템 매개변수에 대해 지정된 값의 합계는 32767을 초과할 수 없습니다.

기본값: 255

SYS1.PARMLIB(IVTPRMxx)

IVTPRMxx는 통신 스토리지 관리자(CSM)에 대한 매개변수를 설정합니다(*MVS Initialization and Tuning Reference*(SA22-7592) 참조). 다음 지시문은 Developer for System z에 영향을 주는 것으로 알려져 있습니다.

FIXED MAX(maxfix)

고정 CSM 버퍼 전용 최대 스토리지 크기를 정의합니다.

값 범위: maxfix 값 범위는 1024K - 2048M입니다.

기본값: 100M

ECSA MAX(maxecsa)

ECSA CSM 버퍼 전용 최대 스토리지 크기를 정의합니다.

값 범위: maxecsa 값 범위는 1024K - 2048M입니다.

기본값: 100M

SYS1.PARMLIB(ASCHPMxx)

ASCHPMxx parmlib 멤버에는 ASCH 트랜잭션 스케줄러의 스케줄링 정보가 포함됩니다(*MVS Initialization and Tuning Reference*(SA22-7592)의 설명 참조). 다음 지시문은 Developer for System z에 영향을 주는 것으로 알려져 있습니다.

MAX(nnnnn)

특정 트랜잭션 이니시에이터 클래스에 허용되는 최대 APPC 트랜잭션 이니시에이터 수를 지정하는 CLASSADD 정의의 선택적 매개변수. 이 한계에 도달하면 새

주소 공간이 작성되지 않고 기존 이니시에이터 주소 공간을 사용할 수 있을 때까지 수신 요청이 큐에서 대기합니다. 값은 설치 시 허용되는 최대 주소 공간 수를 초과해서는 안되므로 시스템에서 역시 주소 공간이 필요한 경쟁 제품에 유의해야 합니다.

값 범위: nnnnn은 1에서 64000까지의 10진수 값입니다.

기본값: 1

참고: APPC를 사용하여 TSO 명령 서비스를 시작하는 경우, Developer for System z의 동시 사용자마다 하나의 이니시에이터를 허용할 수 있도록 사용된 트랜잭션 클래스에 충분한 트랜잭션 이니시에이터가 있어야 합니다.

모니터링

사용자 워크로드에 따라 시스템 자원에 대한 요구가 변경될 수 있으므로 사용자 요구 사항에 따라 Rational Developer for System z와 시스템 구성을 조정할 수 있도록 시스템을 주기적으로 모니터링하여 자원 사용량을 측정해야 합니다. 다음 명령은 이 모니터링 프로세스에 활용할 수 있습니다.

RSE 모니터링

RSE 스레드 풀은 Developer for System z에서 사용자 활동의 중심 위치이므로 최적의 사용에 대한 모니터링이 필요합니다. RSE 디먼에서 일반 시스템 모니터링 도구로 수집할 수 없는 정보를 조회할 수 있습니다.

- 일반 시스템 모니터링 도구(예: RMF™)를 사용하여 사용된 실제 스토리지, CPU 시간과 같은 주소 공간별 데이터를 수집합니다. 전용 모니터링 도구가 없는 경우에는 SDSF DA 보기 또는 TASID(ISPF "지원 및 다운로드" 웹 페이지에서 사용할 수 있는 현재(as-is) 시스템 정보 도구)와 같은 도구로 기본 정보를 수집할 수 있습니다.
- 시작 중에 RSE 디먼은 사용 가능한 주소 공간 크기와 Java 힙 크기를 콘솔 메시지 FEK004I과 함께 보고합니다.

```
FEK004I RseDaemon: Max Heap Size=65MB and private AS Size=1,959MB
```

- **MODIFY RSED,APPL=DISPLAY PROCESS** 연산자 명령은 RSE 스레드 풀 프로세스를 표시합니다. "메모리 사용량" 필드에는 실제로 사용되는 정의된 Java 힙의 양을 보여줍니다. 이 명령에 대한 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "운영자 명령"을 참조하십시오.

```
f rsed,appl=d p
BPXM023I (STCRSE)
ProcessId(16777456) Memory Usage(33%) Clients(4) Order(1)
```

자세한 정보는 **DISPLAY PROCESS** modify 명령의 **DETAIL** 옵션을 사용할 때 제공됩니다.

```
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
```

DISPLAY PROCESS modify 명령의 CPU 옵션은 스레드 풀 내 각 스레드의 누적 CPU 사용량(밀리초)을 보여줍니다.

```
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  0EDE87000000000 005E69C8 001
STCRSE  0EDE98000000000 005E6518 1814
STCRSE  0EDEBA000000000 005E66B0 2305
STCRSE  0EDECB000000000 005E62F8 001
STCRSE  0EDED0000000000 005E60D8 001
STCRSE  0EDF86000000000 005C2BF8 628 6/ThreadPoolMonitor$Memory
UsageMonitor
STCRSE  0EDF97000000000 005C2D90 003 7/ThreadPoolMonitor
IBUSER  0EE2C7000000000 005C08B0 050 38/JESMiner
IBUSER  0EE2B6000000000 005C0690 004 40/FAMiner
IBUSER  0EE30B000000000 005C0250 002 41/LuceneMiner
IBUSER  0EE31C000000000 005C0030 002 42/CDTParserMiner
IBUSER  0EE32D000000000 005BDE00 002 43/MVSLuceneMiner
IBUSER  0EE33E000000000 005BDBE0 002 44/CDTMVSParserMiner
```

- RSE 스레드 풀 프로세스가 종료되면 해당 RSE 스레드 풀 프로세스에만 **DISPLAY PROCESS,DETAIL** modify 명령이 실행된 것처럼 세부 자원 사용량 통계가 표시됩니다. 최고 수위는 RSE 스레드 풀 프로세스 수명에서의 최대 동시 자원 사용량을 보여주므로 시스템 조정자가 RSE에 지정된 자원이 과다 할당 또는 과소 할당되었는지 여부를 결정할 수 있습니다.

z/OS UNIX 모니터링

Developer for System z와 관련이 있는 대부분의 z/OS UNIX 한계는 연산자 명령을 사용하여 표시할 수 있습니다. 일부 명령은 특정 한계에 대한 최고 수위와 현재 사용량까지 표시합니다. 이러한 명령에 대한 자세한 정보는 *MVS System Commands* (SA22-7627)를 참조하십시오.

- SYS1.PARMLIB(BPXPRMxx)의 LIMMSG(ALL) 지시문은 parmlib 한계에 도달할 때 z/OS UNIX에서 콘솔 메시지(BPXI040I)를 표시하도록 지시합니다. LIMMSG의 기본값은 NONE이며 이 경우 기능을 사용하지 않습니다. 연산자 명령 **SETOMVS LIMMSG=ALL**을 사용하면 이 기능이 동적으로 활성화됩니다(다음 IPL까지). 이

지시문에 대한 자세한 정보는 *MVS Initialization and Tuning Reference* (SA22-7592)를 참조하십시오(아래 설명 참조).

- **DISPLAY OMVS,OPTIONS** 연산자 명령은 동적으로 설정할 수 있는 z/OS UNIX 지시문의 현재 값을 표시합니다.

```
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
MAXCPUTIME      =      1000    MAXUIDS        =      200
MAXPTYS        =      256
MAXMMAPAREA     =      256    MAXASSIZE      = 209715200
MAXTHREADS     =      200    MAXTHREADTASKS =      1000
MAXCORESIZE    = 4194304    MAXSHAREPAGES =      4096
IPCMSGQBYTES    = 2147483647 IPCMSGQMNUM     =     10000
IPCSGNIDS       =      500    IPCSEMNIIDS     =      500
IPCSEMNOPI      =      25     IPCSEMNSEMS     =     1000
IPCSHMMPAGES    =     25600    IPCSHMNIIDS     =      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         BRLM SERVER     = N/A
LIMMSG          = NONE      AUTOCVT         = OFF
RESOLVER PROC   = DEFAULT
AUTHPGMLIST    = NONE
SWA            = BELOW
```

- **DISPLAY OMVS,LIMITS** 연산자 명령은 현재 z/OS UNIX 시스템 서비스 parmlib 한계, 해당 최고 수위, 현재 시스템 사용량을 표시합니다.

```
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
IPCSGNIDS           0           0           500
IPCSEMNIIDS         0           0           500
IPCSHMNIIDS         0           0           500
IPCSHMSPAGES        0           0        262144 *
IPCMSGQBYTES        ---          0        262144
IPCMSGQMNUM         ---          0        10000
```



```
IPCSHMMPAGES      ---          0          256
SHRLIBRGNSIZE      0            0        67108864
SHRLIBMAXPAGES      0            0          4096
```

이 명령은 PID=processid 키워드도 함께 지정되는 경우 개별 프로세스의 최고 수위와 현재 사용량을 표시합니다.

```
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
```

- **DISPLAY OMVS,PFS** 연산자 명령은 현재 TCP/IP 스택을 포함하는 z/OS UNIX 구성의 일부인 각 실제 파일 시스템에 대한 정보를 표시합니다.

```
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)
```

- **DISPLAY OMVS,PID=processid** 연산자 명령은 특정 프로세스에 대한 스레드 정보를 표시합니다.

```

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
0E1920000000003CE 005A0B70 OMVS      IBMUSER    14.088 POL  JFNV
0E18D0000000003CF 005A1938      IBMUSER     .581 SND  JYNV

```

네트워크 모니터링

호스트에 연결되는 많은 클라이언트를 지원하는 경우, Developer for System z뿐 아니라 네트워크 인프라 또한 워크로드를 처리할 수 있어야 합니다. 네트워크 관리는 광범위하고 잘 정리된 주제로, Developer for System z 문서 범위에는 해당되지 않습니다. 따라서 다음과 같은 주요사항만 제공됩니다.

- **DISPLAY NET,CSM** 연산자 명령을 사용하면 통신 스토리지 관리자(CSM)가 관리하는 스토리지 사용을 모니터링할 수 있습니다. 이 명령을 사용하면 ECSA 및 데이터 공간 스토리지 풀에 사용 중인 CSM 스토리지의 크기를 결정할 수 있습니다(*Communications Server SNA Operations*(SC31-8779)의 설명 참조).

z/OS UNIX 파일 시스템 모니터링

Developer for System z는 z/OS UNIX 파일 시스템을 사용하여 로그, 임시 파일과 같은 다양한 유형의 데이터를 저장합니다. z/OS UNIX **df** 명령을 사용하면 계속 사용할 수 있는 파일 디스크립터의 수와 기본 HFS 또는 zFS 데이터 세트의 다음 범위가 작성되기 전에 남아 있는 여유 공간을 확인할 수 있습니다.

```

$ 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

```

샘플 설정

다음 샘플 설정은 다음 요구사항을 지원하기 위해 필요한 구성을 보여줍니다.

- 500개 동시 클라이언트 연결
- 300개 동시 MVS 빌드(일괄처리 작업)
- 200개 동시 CARMA 연결(CRASTART 시작 방법 사용)
- 3시간 비활성 제한시간
- z/OS UNIX 사용 비허용

- SCLM 개발자 툴킷과 파일 관리자 통합 사용 안 함
- 평균 20MB의 Java 힙 사용량 예측
- 사용자가 고유 z/OS UNIX UID 소유

스레드 풀 개수

기본적으로 Developer for System z는 단일 스레드 풀에 30명의 사용자를 추가하려고 시도합니다. 그러나 요구사항에는 비활동 제한시간이 활성 상태가 되는 것으로 표시됩니다. 97 페이지의 표 28는 이 경우 연결된 클라이언트당 하나의 스레드가 추가됨을 보여줍니다. 이 스레드는 타이머 스레드이므로 지속적으로 활성 상태를 유지합니다. 이 경우 RSE가 30명의 사용자를 단일 스레드 풀에 배치하지 않습니다. 이는 $10+30*(17+1)=550$ 이 성립하고 maximum.threads가 기본적으로 520으로 설정되기 때문입니다.

maximum.threads를 늘릴 수는 있지만 사용자당 평균 20MB의 Java 힙이 필요한 요구사항으로 인해 maximum.clients를 25로 낮추도록 선택합니다($10+25*18 = 460$). 이렇게 하면 기본 최대 Java 힙 크기인 512MB($20*25 = 500$)를 유지할 수 있습니다.

스레드 풀당 25명의 클라이언트가 배치되고 500개 연결을 지원해야 하는 경우에는 20개 스레드 풀 주소 공간이 필요합니다.

최소 한계 결정

이 장에서 이전에 표시된 공식과 이 절의 시작 부분에서 언급한 기준을 사용하여 수용해야 하는 자원 사용량을 결정할 수 있습니다.

- 주소 공간 개수 - 최대값

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

$$3 + 20 + 500*1 + 200*1 + 300*1 + (2 + 500*0.01) = 1030$$

- 주소 공간 개수 - 사용자당

$$x + y + z$$

$$1 + 1 + 1 = 3$$

- 프로세스 개수 - 최대값

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

$$5 + 2*20 + 500*2 + 200*1 + 300*0 + (10 + 500*0.05) = 1570$$

- 프로세스 개수 - STCRSE

$$4 + 2*A$$

$$4 + 2*20 = 44$$

- 프로세스 개수 - 사용자당

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

$$(2 + 1 + 0) + 5*0 = 3$$

- 스레드 개수 - RSE 스레드 풀

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

$$10 + 25*(17 + 1 + 4 + 0) + (20 + 25*0.1) = 583$$

- 스레드 개수 - JES 작업 모니터

$$3 + N$$

$$3 + 500 = 503$$

- 사용자 ID

$$500 + 2 = 502$$

여분의 두 개 사용자 ID는 Developer for System z 시작 태스크 사용자 ID인 STCJMON 및 STCRSE에 대한 ID입니다.

한계 정의

자원 사용 번호가 알려졌으므로 해당 값을 사용하여 제한 지시문을 사용자 정의할 수 있습니다.

- /etc/rdz/rsed.envvars

- Xmx512m

변경되지 않음

- Dmaximum.clients=25

- Dmaximum.threads=520

변경되지 않음

- Dminimum.threadpool.process=10

이 변경은 선택사항이며, RSE는 필요에 따라 새 스레드 풀을 시작함

- DHIDE_ZOS_UNIX=true

- DDSTORE_IDLE_SHUTDOWN_TIMEOUT=10800000

- FEK.#CUST.PARMLIB(FEJJCNFG)

- MAX_THREADS=503

- SYS1.PARMLIB(BPXPRMxx)

- MAXPROCSYS(2500)

최소 1572, 기타 태스크에 대한 여분의 버퍼가 추가됨

Developer for System z

- MAXPROCUSER(80)

최소 44, RSE 스레드 풀의 경우 여분의 버퍼가 추가됨

프로젝트된 25개 클라이언트보다 적은 지원

- MAXTHREADS(1500)

사용자 ID STCRSE의 OMVS 세그먼트에서 THREADSMAX가 RSE 한계를 설정하는 데 사용되는 경우 최소값 503(JES 작업 모니터의 경우)이어야 함(최소 582)

- MAXTHREADTASKS(1500)

사용자 ID STCRSE의 OMVS 세그먼트에서 THREADSMAX가 RSE 한계를 설정하는 데 사용되는 경우 최소값 503(JES 작업 모니터의 경우)이어야 함(최소 582)

- MAXUIDS(700)

최소값 503, Developer for System z

이외의 기타 태스크에 대한 여분의 버퍼 추가

- MAXASSIZE(209715200)

변경되지 않음(200MB 시스템 기본값), 사용자 ID STCRSE의 OMVS 세그먼트에서 ASSIZEMAX 사용

- SYS1.PARMLIB(IEASYSxx)

- MAXUSER=2000

최소 1030, 기타 태스크에 대한 여분의 버퍼가 추가됨

Developer for System z

- 사용자 ID STCRSE의 OMVS 세그먼트

- ASSIZEMAX(2147483647)

2GB

모니터 자원 사용량

122 페이지의 『한계 정의』의 설명대로 시스템 한계를 활성화한 후 Developer for System z의 자원 사용량 모니터링을 시작하여 일부 변수의 조정이 필요한지 여부를 확인할 수 있습니다. 124 페이지의 그림 25은 499명의 사용자가 로그인한 후 자원 사용량을 보여 줍니다. 이 그림의 예는 로그인만 보여주며 사용자 조치는 예에 표시되지 않습니다.

```

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
Jobname      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

```

제 6 장 성능 고려사항

z/OS는 사용자 정의가 매우 용이한 운영 체제이므로 시스템 변경(때때로 사소한 변경 포함) 시 전체 성능에 막대한 영향을 줄 수 있습니다. 이 장에서는 Developer for System z의 성능을 향상시킬 수 있는 몇몇 변경사항에 대해 설명합니다.

시스템 튜닝에 대한 자세한 정보는 *MVS Initialization and Tuning Guide*(SA22-7591)와 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.

zFS 파일 시스템 사용

zFS(zSeries File System)와 HFS(Hierarchical File System)는 모두 z/OS UNIX 환경에서 사용할 수 있는 UNIX 파일 시스템입니다. 그러나 zFS는 다음과 같은 기능과 이점을 제공합니다.

- 크기가 8K에 가깝고 자주 액세스, 업데이트하는 파일에 액세스하는 경우 많은 고객 환경에서 성능이 향상됩니다. 작은 파일의 액세스 성능은 HFS의 액세스 성능과 동일합니다.
- 동일한 데이터 세트의 읽기 전용 파일 시스템 복제. 복제된 파일 시스템은 사용자가 파일 시스템의 읽기 전용 특정 시점 사본을 제공하기 위해 사용할 수 있습니다. 이 기능은 bisysplex 환경에서만 사용할 수 있는 선택적 기능입니다.
- zFS는 전략적 z/OS UNIX 파일 시스템입니다. HFS 기능은 안정화되었으며 파일 시스템에 대한 개선사항은 zFS에만 해당됩니다.

zFS에 대해 알아보려면 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.

STEPLIB 사용 방지

exec 전반에서 상위에서 하위로 전파되는 STEPLIB가 있는 각 z/OS UNIX 프로세스는 약 200바이트의 ECSA(Extended Common Storage Area)를 이용합니다. STEPLIB 환경 변수가 정의되어 있지 않거나 STEPLIB=CURRENT로 정의된 경우, z/OS UNIX는 현재 활성화된 모든 TASKLIB, STEPLIB, JOBLIB 할당을 fork(), spawn() 또는 exec() 함수 중에 전파합니다.

구성 파일인 rsed.envvars에 설명된 대로 Developer for System z는 rsed.envvars에 기본값 STEPLIB=NONE이 코드화되어 있습니다. 앞서 언급한 이유로 이 지시문을 변경해서는 안되며 대신에 LINKLIST 또는 LPA(Link Pack Area)에 대상 데이터 세트를 배치해야 합니다.

시스템 라이브러리에 대한 액세스 향상

특정 시스템 라이브러리와 로드 모듈은 z/OS UNIX와 애플리케이션 개발 활동에 주로 사용됩니다. 이러한 라이브러리 로드 모듈에 대한 액세스 향상 작업(예: LPA(Link Pack Area)에 추가)은 시스템 성능을 향상시킬 수 있습니다. SYS1.PARMLIB 멤버 변경에 대한 자세한 정보는 *MVS Initialization and Tuning Reference*(SA22-7592)를 참조하십시오(아래 설명 참조).

LE(Language Environment) 런타임 라이브러리

C 프로그램(z/OS UNIX 셸 포함)이 실행되는 경우 일반적으로 LE(Language Environment) 런타임 라이브러리에서 루틴을 사용합니다. 평균적으로 LE 사용 프로그램을 실행하는 모든 주소 공간마다 약 4MB의 런타임 라이브러리가 메모리에 로드되고 모든 포크에 복사됩니다.

CEE.SCEELPA

CEE.SCEELPA 데이터 세트에는 z/OS UNIX에서 주로 사용하는 LE 런타임 루틴의 서브세트가 포함됩니다. 성능을 극대화하려면 이 데이터 세트를 SYS1.PARMLIB(LPALSTxx)에 추가해야 합니다. 이렇게 하면 디스크에서 한 번만 모듈을 읽고 공유 위치에 저장됩니다.

참고: 로드 모듈을 동적 LPA(Link Pack Area)에 추가하려는 경우에는 SYS1.PARMLIB(PROGxx)에 다음 명령문을 추가합니다.

```
LPA ADD MASK(*) DSN(CEE.SCEELPA)
```

SYS1.PARMLIB(LNKLSTxx) 또는 SYS1.PARMLIB(PROGxx)에 데이터 세트를 추가하여 LINKLIST에 LE 런타임 라이브러리 CEE.SCEERUN, CEE.SCEERUN2를 배치하는 것도 좋습니다. 이렇게 하면 z/OS UNIX STEPLIB 오버헤드가 제거되며 LLA와 VLF 또는 유사한 제품의 관리로 인해 입출력(I/O)이 감소합니다.

참고: 같은 이유로 LINKLIST에 C/C++ DLL 클래스 라이브러리 CBC.SCLBDLL도 추가합니다.

이러한 라이브러리를 LINKLIST에 배치하지 않으려면 rsed.envvars에 해당 STEPLIB 문을 설정해야 합니다(rsed.envvars 구성 파일의 설명 참조). 이 방법은 항상 추가 가상 스토리지를 사용하지만 LLA 또는 유사한 제품에 LE 런타임 라이브러리를 정의하여 성능을 향상시킬 수 있습니다. 이렇게 하면 모듈을 로드하는 데 필요한 입출력(I/O)이 감소합니다.

애플리케이션 개발

애플리케이션 개발이 기본 활동인 시스템의 경우 SYS1.PARMLIB(PROGxx)에 다음 행을 추가하여 링크 편집기를 동적 LPA에 배치해도 성능이 향상될 수 있습니다.

```
LPA ADD MODNAME(CEEINIT,CEEBLIBM,CEEV003,EDCV) DSN(CEE.SCEERUN)
LPA ADD MODNAME(IEFIB600,IEFXB603) DSN(SYS1.LINKLIB)
```

C/C++ 개발의 경우 CBC.SCCNCMP 컴파일러 데이터 세트를 SYS1.PARMLIB (LPALSTxx)에 추가할 수 있습니다.

이전 명령문은 가능한 LPA 후보 샘플이며 실제 사이트 요구는 다를 수 있습니다. 기타 LE 로드 모듈을 동적 LPA에 배치하는 작업에 대한 정보는 *Language Environment Customization*(SA22-7564)을 참조하십시오. C/C++ 컴파일러 로드 모듈을 동적 LPA에 배치하는 작업에 대한 자세한 정보는 *UNIX System Services Planning* (GA22-7800)을 참조하십시오.

보안 검사 성능 향상

z/OS UNIX에 수행되는 보안 검사 성능을 향상시키려면 보안 소프트웨어의 FACILITY 클래스에 BPX.SAFFASTPATH 프로파일을 정의합니다. 이렇게 하면 다양한 조작에 대한 z/OS UNIX 보안 검사를 수행할 때 오버헤드가 감소합니다. 여기에는 파일 액세스 확인, IPC 액세스 확인 및 프로세스 소유권 확인이 포함됩니다. 이 프로파일에 대한 자세한 정보는 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.

참고: 사용자는 BPX.SAFFASTPATH 프로파일에 대한 권한이 필요하지 않습니다.

워크로드 관리

각 사이트의 특정 요구에 따라 해당 요구를 충족시키기 위해 사용 가능한 자원을 최대한 활용하도록 z/OS 운영 체제를 사용자 정의할 수 있습니다. 워크로드 관리를 사용하면 성능 목표를 정의하고 각 목표에 비즈니스 중요성을 지정할 수 있습니다. 사용자가 비즈니스 관점에서 업무 목표를 정의하면 시스템이 해당 목표를 충족시키기 위해 작업에 부여해야 하는 자원(예: CPU와 스토리지)의 양을 결정합니다.

해당 프로세스에 올바른 목표를 설정하여 Developer for System z 성능의 균형을 맞출 수 있습니다. 일반 지침의 예는 다음과 같습니다.

- 사용 시 TSO 성능 그룹에 APPC 트랜잭션을 지정합니다.
- Developer for System z 서버 주소 공간(JES 작업 모니터(JMON), RSE 디먼(RSED), RSE 스레드 풀(RSEDx))에 시작 태스크 성능 그룹(SYSSTC)을 지정합니다.

이 주제에 대한 자세한 정보는 *MVS Planning Workload Management* (SA22-7602)를 참조하십시오.

고정 Java 힙 크기

고정 크기 힙을 사용하는 경우 힙 확장 또는 축소가 발생하지 않으므로 일부 상황에서 성능이 크게 향상될 수 있습니다. 그러나 고정 크기 힙을 사용하는 것은 일반적으로 좋은 방법이 아닙니다. 힙이 가득 찰 때까지 가비지 콜렉션 시작이 지연되고 주요 태스크에 대한 부담이 있기 때문입니다. 또한 힙 압축이 필요한 단편화 위험이 증가합니다. 따라서 고정 크기 힙은 올바른 테스트 후에 또는 IBM 지원 센터의 지시가 있는 경우에만 사용해야 합니다. 힙 크기 및 가비지 콜렉션에 대한 자세한 정보는 *Java Diagnostics Guide*(SC34-6650)를 참조하십시오.

JVM(z/OS Java Virtual Machine)의 초기 및 최대 힙 크기는 -Xms (초기값) 및 -Xmx (최대값) Java 명령행 옵션으로 설정할 수 있습니다.

Developer for System z에서 Java 명령행 옵션은 `rsed.envvars`의 `_RSE_JAVAOPTS` 지시문에 , 정의됩니다(*Host Configuration Guide* (SC23-7658)의 "`_RSE_JAVAOPTS`를 사용하여 추가 Java 시작 매개변수 정의" 설명 참조).

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx128m"
```

Java -Xquickstart 옵션

참고: Java -Xquickstart는 RSE 서버에 REXEC/SSH 대체 시작 메소드를 사용하는 경우에만 유용합니다. 이 메소드는 *Host Configuration Guide* (SC23-7658)의 "(선택 사항) REXEC(또는 SSH) 사용"에 설명되어 있습니다.

일부 Java 애플리케이션의 시작 시간을 개선하는 데 -Xquickstart 옵션을 사용할 수 있습니다. -Xquickstart를 사용하면 JIT(Just In Time) 컴파일러가 최적화 서브세트(즉, 빠른 컴파일)를 사용하여 실행됩니다. 이 빠른 컴파일에서는 시작 시간이 향상됩니다.

-Xquickstart 옵션은 단기 실행 애플리케이션(특히, 실행 시간이 소수의 메소드에 집중되지 않은 애플리케이션)에 적합합니다. 핫 메소드가 포함된 장기 실행 애플리케이션에서 -Xquickstart를 사용하는 경우 성능이 저하될 수 있습니다.

RSE 서버에 -Xquickstart 옵션을 사용하려면 `rsed.envvars` 끝에 다음 지시문을 추가하십시오.

```
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xquickstart"
```

JVM 간에 클래스 공유

IBM JVM(Java Virtual Machine) 버전 5 이상에서는 공유 메모리의 캐시에 저장하여 JVM 간에 부트스트랩 및 애플리케이션 클래스를 공유할 수 있습니다. 클래스를 공유하면 두 개 이상의 JVM이 캐시를 공유할 때 전체 가상 메모리 소비가 줄어듭니다. 클래스를 공유하면 캐시가 작성된 후 JVM 시작 시간도 단축됩니다.

공유 클래스 캐시는 활성 JVM에 종속되지 않으며, 캐시를 작성한 JVM의 수명이 다한 후에도 지속됩니다. 공유 클래스 캐시는 JVM이 수명이 다한 후에도 지속되므로 캐시는 파일 시스템의 클래스 또는 JAR의 모든 수정사항을 반영하여 동적으로 업데이트됩니다.

새 캐시를 작성하고 채우는 오버헤드는 소량입니다. 단일 JVM의 경우 JVM 시작 비용은 로드되는 클래스 수에 따라 다르지만 클래스 공유를 사용하지 않는 시스템과 비교하여 일반적으로 0 - 5% 줄어듭니다. 채워진 캐시로 인해 JVM 시작 시간 개선은 로드되는 클래스 수와 운영 체제에 따라 다르지만 클래스 공유를 사용하지 않는 시스템과 비교하여 일반적으로 10% - 40% 빨라집니다. 동시에 실행 중인 다중 JVM을 보면 전체적으로 시작 시간이 훨씬 빨라짐을 알 수 있습니다.

클래스 공유에 대해 자세히 알려면 *Java SDK and Runtime Environment User Guide*를 참조하십시오.

클래스 공유 사용

RSE 서버에 클래스 공유를 사용하려면 `rsed.envvars` 끝에 다음 지시문을 추가하십시오. 첫 번째 명령문은 그룹 액세스 권한을 가진 RSE라는 캐시를 정의하며 클래스 공유에 실패하더라도 RSE 서버 시작을 허용합니다. 두 번째 명령문은 선택사항이며 캐시 크기를 6MB(시스템 기본값은 16MB임)로 설정합니다. 세 번째 명령문은 Java 시작 옵션에 클래스 공유 매개변수를 추가합니다.

```
_RSE_CLASS_OPTS=-Xshareclasses:name=RSE,groupAccess,nonFatal
# RSE_CLASS_OPTS="$_RSE_CLASS_OPTS -Xscmx6m
_RSE_JAVAOPTS="$_RSE_JAVAOPTS $_RSE_CLASS_OPTS"
```

참고: 『캐시 보안』에 설명된 대로 공유 클래스를 사용하는 모든 사용자는 기본 그룹 ID(GID)가 동일해야 합니다. 이는 사용자가 보안 소프트웨어에 정의된 동일한 기본 그룹을 갖거나 다른 기본 그룹이 OMVS 세그먼트에 동일한 GID를 갖고 있어야 함을 의미합니다.

캐시 크기 한계

이론상 최대 공유 캐시 크기는 2GB입니다. 사용자가 지정할 수 있는 캐시 크기는 시스템에 사용 가능한 스왑 공간과 실제 메모리 양에 따라 제한됩니다. 공유 클래스 캐시와 Java 힙 간에 프로세스의 가상 주소 공간이 공유되기 때문에 최대 Java 힙 크기를 늘리면 작성할 수 있는 공유 클래스 캐시 크기가 줄어듭니다.

캐시 보안

공유 클래스 캐시에 대한 액세스는 운영 체제 권한 및 Java 보안 권한에 따라 제한됩니다.

기본적으로 클래스 캐시는 사용자 레벨 보안을 사용하여 작성되므로 캐시를 작성한 사용자만 액세스할 수 있습니다. z/OS UNIX에서는 캐시를 작성한 사용자의 기본 그룹

에 속한 모든 사용자에게 액세스를 제공하는 groupAccess 옵션이 있습니다. 그러나 사용된 액세스 레벨에 관계없이 캐시를 작성한 사용자 또는 루트 사용자(UID 0)만 캐시를 영구 삭제할 수 있습니다.

Java SecurityManager를 사용한 추가 보안 옵션에 대해 자세히 알려면 *Java SDK and Runtime Environment User Guide*를 참조하십시오.

SYS1.PARMLIB(BPXPRMxx)

일부 SYS1.PARMLIB(BPXPRMxx) 설정은 공유 클래스 성능에 영향을 줍니다. 잘못된 설정을 사용하면 공유 클래스 작업이 중지될 수 있습니다. 이러한 설정은 성능에도 영향을 줄 수 있습니다. 이러한 매개변수 사용 및 성능 영향에 대한 추가 정보는 *MVS Initialization and Tuning Reference*(SA22-7592) 및 *UNIX System Services Planning*(GA22-7800)을 참조하십시오. 공유 클래스 조작에 가장 중요한 영향을 미치는 BPXPRMxx 매개변수는 다음과 같습니다.

- MAXSHAREPAGES, IPCSHMPAGES, IPCSHMMPAGES, IPCSHMNSEGS

이러한 설정은 JVM에 사용 가능한 공유 메모리 페이지의 양에 영향을 줍니다. 31 비트 z/OS UNIX 시스템 서비스의 공유 페이지 크기는 4KB로 고정되어 있습니다. 공유 클래스는 기본적으로 16MB 캐시를 작성하려고 합니다. 따라서 IPCSHMMPAGES를 4096보다 크게 설정하십시오.

-Xscmx를 사용하여 캐시 크기를 설정하는 경우, JVM은 가장 가까운 MB로 값을 반올림합니다. 사용자의 시스템에 IPCSHMMPAGES를 설정할 때 이를 고려해야 합니다.

- IPCSEMNIDS 및 IPCSEMNSEMS

이러한 설정은 UNIX에 사용 가능한 세마포어 양에 영향을 줍니다. 공유 클래스는 IPC 세마포어를 사용하여 JVM 간에 통신합니다.

디스크 공간

공유 클래스 캐시에는 시스템에 있는 캐시에 대한 ID 정보를 저장할 디스크 공간이 필요합니다. 이 정보는 /tmp/javasharedresources에 저장됩니다. ID 정보 디렉토리가 삭제되면, JVM이 시스템에서 공유 클래스를 식별할 수 없으며 캐시를 재작성해야 합니다.

캐시 관리 유틸리티

Java -Xshareclasses 행 명령에는 여러 개의 옵션이 있을 수 있는데, 이 중 일부는 캐시 관리 유틸리티입니다. 이 중 세 개가 다음 샘플에 표시되어 있습니다(\$는 z/OS UNIX 프롬프트임). 지원되는 명령행 옵션에 대한 전체 개요는 *Java SDK and Runtime Environment User Guide*를 참조하십시오.

```
$ 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.
```

참고:

- 캐시 유틸리티는 JVM을 시작하지 않고 지정된 캐시에 대해 필요한 조작을 수행하므로 "Could not create the Java virtual machine." 메시지는 정상입니다.
- 캐시를 사용하는 모든 JVM이 종료되고 명령을 실행하는 사용자에게 충분한 권한이 있는 경우에만 캐시를 영구 삭제할 수 있습니다.

제 7 장 클라이언트로 푸시 고려사항

클라이언트로 푸시 또는 호스트 기반 클라이언트 제어는 다음 항목에 대한 중앙 관리를 지원합니다.

- 클라이언트 구성 파일
- 클라이언트 제품 버전
- 프로젝트 정의

이 장에서 다루는 주제는 다음과 같습니다.

- 『소개』
- 136 페이지의 『기본 시스템』
- 137 페이지의 『클라이언트로 푸시 메타데이터』
- 139 페이지의 『클라이언트 구성 제어』
- 139 페이지의 『클라이언트 버전 제어』
- 140 페이지의 『복수 개발자 그룹』
- 144 페이지의 『LDA 기반 그룹 선택』
- 150 페이지의 『SAF 기반 그룹 선택』
- 153 페이지의 『호스트 기반 프로젝트』

소개

Developer for System z 클라이언트 버전 8.0.1 이상은 연결 시 호스트에서 클라이언트 구성 파일과 제품 업데이트 정보를 가져올 수 있으므로 모든 클라이언트가 공통 설정을 갖고 최신 상태를 유지합니다.

버전 8.0.3부터는 클라이언트 관리자가 다양한 개발자 그룹의 요구에 맞는 여러 클라이언트 구성 세트와 여러 클라이언트 업데이트 시나리오를 작성할 수 있습니다. 따라서 사용자가 LDAP 그룹 멤버십 또는 보안 프로파일 허가와 같은 기준에 따라 사용자 정의 설정을 수신할 수 있습니다.

클라이언트의 z/OS 프로젝트 퍼스펙티브를 통해 개별적으로 z/OS 프로젝트를 정의하거나 z/OS 프로젝트를 호스트에서 중앙 집중식으로 정의하여 개별 사용자별로 클라이언트에 전파할 수 있습니다. 이러한 "호스트 기반 프로젝트"의 모양과 기능은 해당 구조, 멤버, 특성을 클라이언트가 수정할 수 없고 호스트에 연결한 상태에서만 액세스할 수 있다는 점을 제외하고는 클라이언트에 정의된 프로젝트에 일치합니다.

pushtoclient.properties는 해당 기능을 사용하는지 여부와 관련 데이터 저장 위치를 클라이언트에 알려줍니다. 자세한 정보는 *Host Configuration Guide*(SC23-7658)의 "(선택사항) pushtoclient.properties, 호스트 기반 클라이언트 제어"를 참조하십시오.

일반적으로 z/OS 시스템, 개발자 워크스테이션, 개발 프로젝트는 다른 사용자 그룹이 관리합니다. 클라이언트로 푸시 디자인은 이 원칙을 준수하며 각 그룹에 특정 책임을 지정합니다.

- z/OS 시스템 프로그래머는 클라이언트로 푸시 메타데이터의 위치, 기본 보안 측면, 클라이언트로 푸시 활성화 여부를 제어합니다.
- 클라이언트 관리자는 Developer for System z 클라이언트를 사용하여 하나 이상의 클라이언트 구성을 작성하거나 IBM Installation Manager를 사용하여 Developer for System z 클라이언트를 업데이트하는 데 사용되는 응답 파일을 작성하여 클라이언트로 푸시 메타데이터의 콘텐츠를 유지관리합니다.
- 개발 프로젝트 관리자는 프로젝트를 정의하고 개별 개발자를 프로젝트에 지정합니다.

클라이언트 관리자와 개발 프로젝트 관리자가 연관된 태스크를 수행할 수 있는 방법에 대한 세부사항은 Developer for System z Information Center (<http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp>)를 참조하십시오.

여러 개발자 그룹에 대한 구성 또는 버전 제어 지원을 사용하는 경우에는 한 팀이 더 클라이언트로 푸시 관리에 참여합니다. 해당 팀은 개발자가 속하는 그룹을 식별하기 위해 선택된 옵션에 따라 다릅니다.

- LDAP 관리자는 각 개발자를 없음, 하나 이상의 FEK.PTC.* LDAP 그룹에 배치하는 그룹 정의를 유지관리합니다.
- 보안 관리자는 FEK.PTC.* 보안 프로파일에 대한 액세스 목록을 유지합니다. 개발자에게 없음, 하나 이상의 프로파일에 대한 권한을 부여할 수 있습니다.

기본 시스템

클라이언트로 푸시는 관리 노력을 줄이기 위해 단일 시스템(기본 시스템)에서 공통(글로벌) 데이터를 유지하면서 시스템의 시스템별 데이터를 저장하도록 디자인되어 있습니다. 기본 시스템은 pushtoclient.properties의 primary.system 지시문으로 식별됩니다. 기본값은 false입니다.

기본 시스템으로 정의되는 시스템은 하나뿐이어야 합니다. Developer for System z 클라이언트 관리자는 대상 시스템이 기본 시스템이 아닌 한 글로벌 구성 데이터를 내보낼 수 없습니다. Developer for System z 클라이언트는 동기화되지 않은 구성으로 여러 기본 시스템에 연결할 때 불규칙적인 동작을 나타낼 수 있습니다.

여러 시스템이 Developer for System z 구성(/etc/rdz)과 클라이언트로 푸시 메타데이터(/var/rdz/pushtoclient)를 공유하는 경우에는 유일 규칙이 적용되지 않습니다.

구성을 공유하므로 모든 관련 시스템은 기본 시스템으로 식별됩니다. 그러나 모든 시스템이 메타데이터도 공유하는 경우에는 이러한 중복이 문제가 되지 않습니다.

클라이언트로 푸시 메타데이터

메타데이터 위치

`pushtoclient.properties`의 `pushtoclient.folder` 지시문은 클라이언트로 푸시 메타데이터가 저장되는 기본 디렉토리를 식별합니다. 기본값은 `/var/rdz/pushtoclient`입니다.

기본 디렉토리는 루트 클라이언트로 푸시 구성 파일인 `keymapping.xml`을 보관합니다. 다른 모든 메타데이터는 서브디렉토리에 있습니다.

대부분의 서브디렉토리는 클라이언트 관리자가 클라이언트로 푸시 작업공간 구성을 내보낼 때 동적으로 작성됩니다. 이러한 서브디렉토리는 메타데이터를 주제(예: 맵핑과 환경 설정)별로 그룹화합니다. 보다 많은 Developer for System z 클라이언트 컴포넌트를 클라이언트로 푸시로 관리할 수 있게 되면 보다 많은 서브디렉토리가 동적으로 작성됩니다. 이러한 서브디렉토리에 저장되는 항목에 대해 알아보려면 Developer for System z 클라이언트의 내보내기 마법사(파일 > 내보내기... > **Rational Developer for System z** > 구성 파일)를 참조하십시오.

일부 서브디렉토리는 초기 호스트 사용자 정의 중에 작성됩니다. 이러한 서브디렉토리에는 클라이언트 관리자 또는 개발 프로젝트 관리자가 수동으로 유지관리하는 데이터가 보관됩니다.

- `/var/rdz/pushtoclient/projects/`는 호스트 기반 프로젝트 정의 파일을 보관합니다. 실제 위치는 `/var/rdz/pushtoclient/keymapping.xml`에 지정됩니다. 이 파일은 Developer for System z 클라이언트 관리자가 작성, 유지관리합니다. 해당 위치의 파일은 프로젝트 관리자 또는 리드 개발자가 유지관리합니다.
- `/var/rdz/pushtoclient/install/`은 호스트에 대한 연결 시 클라이언트 제품 버전을 업데이트하는 데 사용되는 구성 파일을 보관합니다. 실제 위치는 `/var/rdz/pushtoclient/keymapping.xml`에 지정됩니다. 이 파일은 Developer for System z 클라이언트 관리자가 작성, 유지관리합니다. 해당 위치의 파일은 클라이언트 관리자가 유지관리합니다.
- `/var/rdz/pushtoclient/install/responsefiles/`는 호스트에 대한 연결 시 클라이언트 제품 버전을 업데이트하는 데 사용되는 구성 파일을 보관합니다. 실제 위치는 `/var/rdz/pushtoclient/keymapping.xml`에 지정됩니다. 이 파일은 Developer for System z 클라이언트 관리자가 작성, 유지관리합니다. 해당 위치의 파일은 클라이언트 관리자가 유지관리합니다.

이러한 서브디렉토리 작성에 대한 자세한 정보는 *Host Configuration Guide(SC23-7658)*, "기본 사용자 정의" 장의 "사용자 정의 설정"을 참조하십시오.

메타데이터 보안

기본적으로(pushtoclient.properties의 file.permission 지시문 참조) 기본 디렉토리에서 작성되는 모든 파일과 디렉토리는 소유자와 소유자의 기본 그룹에 디렉토리 구조와 해당 파일에 대한 읽기 및 쓰기 액세스 권한을 허용하는 권한 비트마스크 775(rwxrwxr-x)를 수신합니다. 다른 파일과 디렉토리는 디렉토리 구조와 해당 파일에 대한 읽기 액세스 권한만 보유합니다.

클라이언트로 푸시 설정을 시작하기 전에 이러한 디렉토리에 올바른 소유자 UID(사용자 ID)와 GID(그룹 ID)가 설정되어야 합니다.

다음 샘플 RACF 명령은 새 그룹(RDZADMIN)을 작성하고 고유 GID(2)를 지정하며 사용자 ID RDZADM1의 기본 그룹으로 설정합니다. 이 그룹 역시 고유 UID(6)을 수신합니다.

```
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))
```

다음 샘플 **chown** z/OS UNIX 명령은 /var/rdz/pushtoclient와 모든 해당 항목의 소유자와 그룹을 각각 RDZADM1과 RDZADMIN으로 변경합니다. 권한 문제점을 방지하려면 슈퍼유저(UID 0)가 이 명령을 실행해야 합니다.

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

다음 샘플 **chmod** z/OS UNIX 명령은 /var/rdz/pushtoclient와 모든 해당 항목의 권한 비트마스크를 775로 변경합니다. 이 명령을 실행하면 디렉토리에 수동으로 추가한 후 Developer for System z에서 사용하는 로직이 실행됩니다. 권한 문제점을 방지하려면 슈퍼유저(UID 0)가 이 명령을 실행해야 합니다.

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

샘플 RACF 명령에 대한 자세한 정보는 *Security Server RACF Command Language Reference(SA22-7687)*를 참조하십시오. 샘플 z/OS UNIX 명령에 대한 자세한 정보는 *UNIX System Services Command Reference(SA22-7802)*를 참조하십시오. 추가 정보는 16 페이지의 『z/OS UNIX 디렉토리 구조』를 참조하십시오.

메타데이터 공간 사용

클라이언트로 푸시 메타데이터는 해당 벌크가 UTF-8 인코드 XML 파일이므로 z/OS UNIX에서 매우 적은 양의 디스크 공간을 사용합니다. 클라이언트 업데이트 시나리오에 사용된 제품 코드는 네트워크에서 임의 위치에 저장될 수 있습니다. 즉, 관련 클라

이언트로 푸시 메타데이터(응답 파일이라고 함)가 클라이언트를 올바른 위치로 가리키므로 제품 코드를 z/OS UNIX에 저장하지 않아도 됩니다.

클라이언트 구성 제어

Developer for System z 클라이언트(버전 8.0.1 이상)가 호스트에 연결되면 `pushtoclient.properties`의 정의를 읽습니다. 지시문 `config.enabled`를 사용하는 경우, 클라이언트는 현재 구성과 클라이언트로 푸시 메타데이터의 정의를 비교합니다. 차이가 발견되면 클라이언트가 필요한 데이터를 가져오고 클라이언트로 푸시의 지시대로 설정을 활성화하는 마법사를 시작합니다.

`pushtoclient.properties`의 `reject.config.updates` 지시문은 클라이언트로 푸시가 전달될 시점에 사용자가 구성 업데이트를 거부할 수 있는지 여부를 제어합니다.

Developer for System z 클라이언트(버전 8.0.1 이상)에는 클라이언트 관리자가 사용하는 마법사가 있습니다. 이 마법사는 현재 구성을 내보낼 수 있으며 해당 구성은 모든 Developer for System z 클라이언트가 클라이언트로 푸시를 통해 가져옵니다. 이 기능은 모든 클라이언트에서 사용할 수 있으므로 클라이언트로 푸시 메타데이터를 보유하는 z/OS UNIX 디렉토리(/var/rdz/pushtoclient)에 대한 쓰기 권한이 클라이언트 관리자에게만 있는지 확인해야 합니다.

클라이언트와 호스트가 모두 그룹 지원을 사용하려면 버전 8.0.3 이상이 필요합니다(140 페이지의 『복수 개발자 그룹』의 설명 참조).

클라이언트 버전 제어

Developer for System z 클라이언트(버전 8.0.1 이상)가 호스트에 연결되면 `pushtoclient.properties`의 정의를 읽습니다. 지시문 `product.enabled`를 사용하는 경우, 클라이언트는 현재 제품 버전과 클라이언트로 푸시 메타데이터의 정의를 비교합니다. 차이가 발견되면 클라이언트가 필요한 데이터를 가져오고 클라이언트로 푸시의 지시대로 설정을 활성화하는 마법사를 시작합니다.

`pushtoclient.properties`의 `reject.product.updates` 지시문은 클라이언트로 푸시가 전달될 시점에 사용자가 제품 업데이트를 거부할 수 있는지 여부를 제어합니다.

클라이언트와 호스트가 모두 그룹 지원을 사용하려면 버전 8.0.3 이상이 필요합니다(140 페이지의 『복수 개발자 그룹』의 설명 참조).

복수 개발자 그룹

버전 8.0.3부터는 클라이언트 관리자가 다양한 개발자 그룹의 요구에 맞는 여러 클라이언트 구성 세트와 여러 클라이언트 업데이트 시나리오를 작성할 수 있습니다. 따라서 사용자가 LDAP 그룹 멤버십 또는 보안 프로파일 허가과 같은 기준에 따라 사용자 정의 설정을 수신할 수 있습니다.

활성화

고유한 클라이언트 구성 및 클라이언트 업데이트 요구사항을 갖는 여러 개발자 그룹에 대한 지원을 사용하려면 관련 지시문(`pushtoclient.properties`의 `config.enabled` 및 `product.enabled`)에 원하는 값을 지정해야 합니다(표 33 참조).

표 33. *.enabled에 대한 클라이언트로 푸시 그룹 지원 매트릭스

*.enabled 값	기능 사용	복수 그룹 지원
false	아니오	아니오
true	예	아니오
LDAP	예	예. LDAP 그룹 FEK.PTC.*.ENABLED.sysname.devgroup의 멤버십 기반
SAF	예	예, 보안 프로파일 FEK.PTC.*.ENABLED.sysname.devgroup에 대한 허가 기반

기능을 사용하는 경우(TRUE 값 포함) 개발자는 항상 기본 그룹에 포함됩니다. 개발자는 없음, 하나 또는 여러 추가 그룹에 속할 수 있습니다.

업데이트 거부를 조건부로 지정할 수도 있습니다(표 34 참조).

표 34. reject.*.updates에 대한 클라이언트로 푸시 그룹 지원 매트릭스

reject.*.updates 값	기능 사용
false	아니오
true	예
LDAP	LDAP 그룹 멤버십 FEK.PTC.REJECT.*.UPDATES.sysname에 따라 다릅니다.
SAF	보안 프로파일 FEK.PTC.REJECT.*.UPDATES.sysname에 대한 권한에 따라 다릅니다.

`pushtoclient.properties`의 지시문은 서로 독립적으로 실행됩니다. 원하는 지시문에 지원되는 값을 지정할 수 있습니다. 설정을 동일하게 유지할 필요는 없습니다.

각 기능에 필요한 설정에 대한 세부사항은 144 페이지의 『LDA 기반 그룹 선택』 및 150 페이지의 『SAF 기반 그룹 선택』을 참조하십시오. 여러 그룹 지원 사용에 대한 자세한 정보는 *Host Configuration Guide*(SC23-7658)의 "(선택사항) `pushtoclient.properties`, 호스트 기반 클라이언트 제어"를 참조하십시오.

그룹 연결

pushtoclient.properties에서 *.enabled 기능을 사용하는 경우(TRUE 값 포함) 개발자는 항상 관련 기능의 기본 그룹에 포함됩니다. 개발자는 없음, 하나 또는 여러 추가 그룹에 속할 수 있습니다.

여러 그룹에 정의된 변경사항 적용의 복잡도를 제한하기 위해 Developer for System z는 사용자 선택에 따라 사용될 정의를 제한합니다.

표 35. 클라이언트로 푸시 그룹 연결

추가 그룹	사용된 정의
없음	기본값
하나	기본값 또는 (기본값 + 그룹)
복수	기본값 또는 (기본값 + 하나의 그룹)

Developer for System z는 변경 세트를 빌드, 적용할 때 다음 로직을 사용합니다.

1. 해당되는 경우 기본 정의에 지정된 업데이트를 가져옵니다.
2. 해당되는 경우 기본값을 변경하여(이미 존재하는 경우) 선택한 그룹 정의에 지정된 업데이트를 가져옵니다.
3. 클라이언트에서 업데이트를 적용합니다.

참고: 업데이트는 삭제, 추가, 오버레이 조치로 구성됩니다.

작업공간 바인딩

한 명의 개발자가 동시에 여러 그룹에 속할 수는 있지만 개발자의 활성 작업공간은 동시에 여러 그룹에 속할 수 없습니다. 구성 또는 제품 업데이트를 받으려면 활성 작업공간이 특정 그룹(기본 그룹 가능)에 바인딩되어야 합니다. 완료된 바인드는 실행 취소할 수 없습니다. 새 그룹 바인딩이 필요한 경우 새 작업공간을 작성해야 합니다.

그룹 바인딩이 없는 작업공간이 호스트에 연결되고 config.enabled(또는 product.enabled)가 클라이언트로 푸시 기능이 활성 상태임을 나타내면 Developer for System z가 모든 그룹을 조회하여 사용자가 속하는 그룹을 결정하고 사용자에게 관련 기능의 그룹을 선택하라는 프롬프트를 표시합니다. 연속 연결 시에는 선택한 그룹만 조회하여 그룹 멤버십이 계속 올바른지 여부를 확인합니다.

reject.*.updates 지시문은 여러 그룹에 대한 작업을 수행하지 않으므로 설정이 보다 간단하고 작업공간 바인딩이 필요하지 않습니다. 업데이트가 있는 경우 Developer for System z는 사용자가 업데이트를 거부할 수 있는지 여부를 결정하고 그에 따라 조치를 수행합니다.

그룹 메타데이터 위치

137 페이지의 『메타데이터 위치』의 설명대로 그룹 지원 없이 설정을 사용하는 경우 모든 클라이언트로 푸시 메타데이터는 `/var/rdz/pushtoclient/`뿐 아니라 디렉토리 구조에도 저장됩니다. 그룹 지원이 활성화되는 경우에도 동일한 데이터 레이아웃이 유지되지만 기본 디렉토리 `/var/rdz/pushtoclient/`에 대한 해석은 다음과 같이 약간 다릅니다.

- `/var/rdz/pushtoclient/`의 기존 데이터는 기본 그룹의 데이터로 해석됩니다. 기본 그룹으로 내보내면 `/var/rdz/pushtoclient/`의 메타데이터가 작성 또는 업데이트됩니다. 이 해석은 버전 8.0.1, 버전 8.0.2 클라이언트(클라이언트로 푸시를 사용하지만 여러 그룹은 지원하지 않음)와의 호환성을 보장합니다.
- 그룹으로 내보내면 `/var/rdz/pushtoclient/` 대신 기본 디렉토리인 것처럼 `/var/rdz/pushtoclient/grouping/<devgroup>/`의 메타데이터를 작성 또는 업데이트합니다. `<devgroup>` 값은 특정 개발자 그룹에 지정된 그룹 이름을 비교합니다.

초기 제품 사용자 정의는 `/var/rdz/pushtoclient/`에 `grouping/` 디렉토리를 작성합니다. 클라이언트 관리자는 `/var/rdz/pushtoclient/grouping/`에 `<devgroup>/` 디렉토리를 추가해야 합니다.

초기 제품 사용자 정의 동안에는 `projects/`, `install/`, `install/responsefiles/` 디렉토리가 `/var/rdz/pushtoclient/`에서 작성됩니다. 그룹별 제품 업그레이드 시나리오 또는 그룹별 호스트 기반 프로젝트가 필요한 경우 클라이언트 관리자가 `/var/rdz/pushtoclient/grouping/<devgroup>/`에서 이러한 디렉토리 작성 조치를 반복해야 합니다.

다음 샘플 z/OS UNIX 명령 시퀀스는 올바른 권한 비트마스크로 서브디렉토리를 작성합니다. 클라이언트 관리자는 소유권 문제점을 방지하기 위해 이 명령을 실행해야 합니다.

```
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
```

샘플 z/OS UNIX 명령에 대한 자세한 정보는 *UNIX System Services Command Reference*(SA22-7802)를 참조하십시오.

설정 단계

여러 개발자 그룹에 대한 지원을 설정하려면 z/OS 시스템 프로그래머, 클라이언트 관리자, 선택 기준을 관리하는 관리자(LDAP 또는 보안 관리자) 간에 조정 작업이 필요합니다. 워크플로우에 대한 다음 설명에서는 보안 관리자가 선택 기준을 관리합니다.

1. 클라이언트 관리자는 개발자의 기존 그룹화 설정에 대한 입력을 보안 관리자에게 요청합니다. 기존 설정을 재사용하면 속도가 빨라지고 클라이언트로 푸시 설정이 단순화됩니다.
2. 클라이언트 관리자는 원하는 다중 그룹 지원 구조화 방법과 해당 클라이언트로 푸시 그룹의 구성원이 될 사용자를 결정합니다.

참고:

- 항상 기본 구성 세트와 기본 제품 업데이트 시나리오가 존재합니다.
 - 클라이언트로 푸시 변경 세트에는 삭제, 추가, 오버레이 조치가 포함될 수 있습니다.
 - 클라이언트로 푸시 변경 세트는 비어 있을 수 있습니다.
 - 개발자는 없음, 하나 또는 여러 클라이언트로 푸시 그룹에 속할 수 있습니다.
 - 클라이언트 관리자는 각 클라이언트로 푸시 그룹의 구성원이어야 합니다.
3. 클라이언트 관리자와 보안 관리자는 사용할 클라이언트로 푸시 그룹 이름에 동의합니다.
 4. 클라이언트 관리자가 각 클라이언트로 푸시 그룹의
`/var/rdz/pushtoclient/grouping/<devgroup>`

디렉토리를 작성합니다.

참고: 이 디렉토리에 대한 권한 비트는 775(drwxrwxr-x)여야 합니다.

5. 보안 관리자는 클라이언트로 푸시 선택 기준 프로파일을 정의하는 데 필요한 초기 설정을 수행하고 액세스 목록에 클라이언트로 푸시 그룹을 추가합니다.

참고:

- 클라이언트 관리자가 관련 클라이언트로 푸시 메타데이터를 작성하려면 최소한 액세스 목록의 클라이언트 관리자로 선택 기준 구조를 정의해야 합니다.
 - 초기 설정의 경우에는 클라이언트로 푸시 그룹의 액세스 목록에 클라이언트 관리자만 있어야 합니다. 이 경우 Developer for system z 클라이언트가 생성 중인 설정을 수신하지 못합니다.
6. z/OS 시스템 프로그래머는 `pushtoclient.properties`를 조정하여 다중 그룹 지원을 활성화합니다.

참고: 클라이언트 관리자가 관련 클라이언트로 푸시 메타데이터를 작성하려면 `*.enabled` 지시문을 사용해야 합니다.

7. 클라이언트 관리자는 각 그룹의 작업공간을 작성하고 각 그룹 이름을 사용하는 호스트로 내보냅니다. 클라이언트 관리자는 또한 그룹별 제품 업데이트 시나리오를 작성하는 데 필요한 응답 파일을 작성합니다.
8. 보안 관리자는 클라이언트로 푸시 그룹에 개발자를 추가하고 개발자에 대해 클라이언트로 푸시를 활성화합니다.

LDA 기반 그룹 선택

LDAP(Lightweight Directory Access Protocol)는 TCP/IP 기반 프로토콜의 이름이지만 일반적으로 분배 디렉토리 서비스 세트를 설명하는 데 사용됩니다. 디렉토리는 데이터베이스와 마찬가지로 구조화된 레코드 컬렉션입니다. Developer for System z는 LDAP 서버를 단순 계층 구조 데이터베이스로 사용할 수 있습니다. 이 데이터베이스에서는 그룹이 하나 이상의 멤버를 보유합니다.

LDAP 서버의 정의를 선택 메커니즘으로 사용하는 경우(LDAP 값이 `pushtoclient.properties`의 지시문에 대해 지정됨), Developer for System z는 표 36에 나열된 그룹 이름의 멤버십을 확인하여 사용자가 속하는 개발자 그룹과 사용자가 업데이트를 거부할 수 있는지 여부를 결정합니다.

표 36. 클라이언트로 푸시 LDAP 정보

그룹 이름(cn=)	결과
FEK.PTC.CONFIG.ENABLED.sysname.devgroup	클라이언트가 지정된 그룹에 대한 구성 업데이트를 허용합니다.
FEK.PTC.PRODUCT.ENABLED.sysname.devgroup	클라이언트가 지정된 그룹에 대한 제품 업데이트를 허용합니다.
FEK.PTC.REJECT.CONFIG.UPDATES.sysname	사용자가 구성 업데이트를 거부할 수 있습니다.
FEK.PTC.REJECT.PRODUCT.UPDATES.sysname	사용자가 제품 업데이트를 거부할 수 있습니다.

`devgroup` 값은 특정 개발자 그룹에 지정된 그룹 이름을 비교합니다. 그룹 이름은 Developer for System z 클라이언트에서 표시됩니다.

`sysname` 값은 대상 시스템의 시스템 이름을 비교합니다.

LDAP 스키마

LDAP 스키마는 다음 규칙을 충족시켜야 합니다.

1. 각 클라이언트로 푸시 그룹은 스키마에서 그룹으로 정의되어야 합니다.
2. 각 사용자는 스키마에서 사용자로 정의되어야 합니다.
3. 그룹 항목에는 해당 그룹에 속하는 사용자 항목에 대한 참조가 있습니다.

그림 26는 그룹과 사용자에 대한 샘플 LDAP 정의(LDIF 형식으로 표시)입니다.

참고: LDIF(LDAP Data Interchange Format)는 LDAP 오브젝트와 LDAP 업데이트를 나타내기 위한 표준 텍스트 형식입니다. LDIF 레코드를 포함하는 파일은 디렉토리 서버 간에 데이터를 전송하기 위해 또는 LDAP 유틸리티의 입력으로 사용됩니다.

```
# Group Definition
dn: cn=FEC.PTC.CONFIG.ENABLED.CDFMVS08.GROUPA,o=PTC,c=DeveloperForZ
objectClass: groupOfUniqueNames
objectClass: top
cn: FEC.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
```

그림 26. 샘플 LDAP 스키마 정의

LDAP 서버 선택

다양한 상용 및 무료 LDAP 서버를 선택할 수 있습니다. 한 예로 IBM Tivoli® Directory Server(<http://www-01.ibm.com/software/tivoli/products/directory-server/>)가 있습니다. LDAP 서버를 관리하기 위해 다양한 명령행 및 GUI 기반 도구를 선택할 수도 있습니다.

144 페이지의 『LDAP 스키마』에서 언급한 것처럼 각 사용자는 LDAP 서버에 정의되어야 합니다. 관리 노력을 줄이려면 이미 모든 사용자 정의에 대한 액세스 권한이 있는 LDAP 서버에 클라이언트로 푸시 스키마를 배치하는 것이 가장 효과적입니다. 예를 들어, SDBM 데이터베이스(보안 데이터베이스의 래퍼)를 사용하여 z/OS에서 활성화된 IBM Tivoli Directory Server를 사용할 수 있습니다.

사이트 정책에 따라 LDAP 서버의 클라이언트로 푸시 스키마를 클라이언트 관리자가 관리할 수 있습니다. 이러한 경우 협업 요구와 함께 가능한 지연 및 통신 오류가 감소합니다.

클라이언트 관리자의 LDAP 관리를 지지하는 주장은 클라이언트로 푸시 스키마가 기밀 또는 보안 관련 항목을 보존하지 않는다는 것입니다. LDAP 서버가 다른 스키마를 통해 사용자 정의를 사용할 수 있는 경우, Developer for System z LDAP 오브젝트는 작업공간 레이아웃과 자동 Developer for System z 클라이언트 제품 업그레이드에 대한 개발자의 선택사항을 결정합니다.

LDAP 서버 위치

LDAP 프로토콜을 지원하는 데이터베이스 서버를 사용하여 Developer for System z 클라이언트로 푸시 스키마를 호스트할 수 있습니다. 따라서 Developer for System z를 사용하면 LDAP 서버에 연결하는 데 필요한 정보를 지정할 수 있습니다. 또한 LDAP 서버 내에서 데이터베이스를 고유하게 만드는 접미부를 지정할 수 있습니다.

rsed.envvars 지시문	기본값
_RSE_LDAP_SERVER	로컬 호스트 시스템
_RSE_LDAP_PORT	389
_RSE_LDAP_PTC_GROUP_SUFFIX	"O=PTC,C=DeveloperForZ"

TCP/IP 보안 조치(예: 방화벽)로 인해 (호스트 기반) RSE 서버가 LDAP 서버에 접속하지 못할 수 있습니다. TCP/IP 관리자에게 다음 정보를 제공하여 LDAP 서버에 도달할 수 있는지 확인할 수 있습니다.

- LDAP 서버 TCP/IP 주소 또는 DNS 이름
- LDAP 서버 포트 번호
- LDAP는 TCP 프로토콜을 사용합니다.
- LDAP 서버에 호스트 기반 RSE 서버가 접속합니다.
- RSE 서버는 RSEDx 주소 공간에서 활성화됩니다. 여기서 RSED는 RSE 시작 태스크 이름이고 x는 1자리 난수입니다.

샘플 설정

CDFMVS08 시스템에 Developer for System z가 활성화된 회사가 있는 것으로 가정합니다. IBM Tivoli Directory Server는 CDFMVS08에서도 활성화되며 LDAP 서버로 사용됩니다. LDAP 서버는 147 페이지의 『LDAP에 클라이언트로 푸시 백엔드 추가』의 설명대로 구성됩니다.

Developer for System z를 사용하는 사용자는 다음과 같습니다.

- 은행 애플리케이션 작업을 수행하는 개발자(사용자 ID BNK010 -> BNK014)
- 보험 애플리케이션 작업을 수행하는 개발자(사용자 ID INS010 -> INS014)
- Developer for System z 클라이언트 관리자(사용자 ID RDZADM1)

각 개발자 그룹에는 특정 클라이언트 구성 파일이 필요하며 모든 개발자는 동일한 클라이언트 버전의 제어를 받을 수 있습니다. 클라이언트 관리자와 달리 개발자는 클라이언트로 푸시가 제공하는 변경사항을 거부할 수 없습니다.

클라이언트 관리자와 LDAP 관리자는 구성 업데이트를 위해 그룹 이름 BANKING과 INSURANCE를 사용하는 데 동의했습니다.

LDAP에 클라이언트로 푸시 백엔드 추가

이 예에서는 클라이언트로 푸시 스키마를 호스트할 LDBM 데이터베이스(z/OS UNIX 파일)를 추가하여 현재 SDBM 데이터베이스(보안 데이터베이스 랩퍼)만 사용하는 z/OS의 IBM Tivoli Directory Server를 업데이트합니다.

1. LDAP 구성 파일에 LDBM 백엔드 섹션을 추가합니다.

```
# 파일 이름 ds.conf
# GLDSRV 시작 태스크를 재시작하여 변경사항 선택

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

# SDBM 백엔드 섹션(RACF)
database SDBM GLDBSD31/GLDBSD64
suffix "cn=RACF,o=IBM,c=US"

# LDBM 백엔드 섹션(z/OS UNIX 파일)
database LDBM GLDBLD31/GLDBLD64 LDBM-RDZ
suffix "o=PTC,c=DeveloperForZ"
databaseDirectory /var/ldap/ldbm/rdz
```

2. LDAP 시작 태스크, GRDSRV를 중지했다가 시작하여 구성 변경사항을 선택합니다.
3. /var/ldap/ldbm/rdz 디렉토리를 작성합니다.

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

4. LDAP 스키마를 업데이트하여 LDBM 백엔드를 추가합니다.

```
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. LDBM 백엔드에 루트 항목을 추가합니다.

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

여기서 /u/ibmuser/ptc_root.ldif에는 다음이 포함됩니다.

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

초기 LDAP 그룹 설정

스키마에 다른 LDAP 그룹 오브젝트를 추가하고 클라이언트 관리자를 각 오브젝트의 일부로 만듭니다. RDZADM1 사용자 ID에 대한 사용자 정의는 RACF 스키마에서 가져옵니다.

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

여기서 /u/ibmuser/ptc_setup.ldif에는 다음이 포함됩니다.

```
# 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
```

LDAP 그룹에 개발자 추가

LDAP 그룹 오브젝트에 개발자를 추가합니다. 사용자 ID에 대한 사용자 정의는 RACF 스키마에서 가져옵니다.

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

여기서 /u/ibmuser/ptc_add.ldif에는 다음이 포함됩니다.

```
# 은행 작업공간 구성
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
```

보험 작업공간 구성

```
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 및 INSURANCE에는 다른 구성 요구가 있음
config.enabled=LDAP
# 모두가 제품 업데이트를 받음
product.enabled=TRUE
# RDZADMIN만 구성 업데이트를 거부할 수 있음
reject.config.updates=LDAP
# RDZADMIN만 제품 업데이트를 거부할 수 있음
reject.product.updates=LDAP
```

rsed.envvars

기본값을 사용하므로 업데이트가 필요하지 않습니다.

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

/var/rdz/pushtoclient/*install

BANKING, INSURANCE 그룹에 대한 작업공간 구성을 내보내는 동안 내보내기 마법사가 `/var/rdz/pushtoclient/grouping/<devgroup>/` 디렉토리와 배경 디렉토리 구조를 작성합니다.

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

개별 제품 업그레이드 시나리오가 없으므로 클라이언트 관리자가 `/var/rdz/pushtoclient/grouping/<devgroup>/install/, install/responsefiles/` 서브디렉토리를 작성 또는 업데이트하지 않아도 됩니다.

클라이언트 관리자는 기본 그룹 디렉토리, `/var/rdz/pushtoclient/install/responsefiles/`에서 제품 업데이트에 필요한 응답 파일을 작성해야 합니다.

SAF 기반 그룹 선택

SAF(Security Access Facility)는 z/OS 보안 제품에 액세스하기 위한 인터페이스입니다. Developer for System z는 이 인터페이스를 사용하여 보안 제품을 조회하고 클라이언트로 푸시 관련 정보를 검색할 수 있습니다.

보안 데이터베이스의 정의를 선택 메커니즘으로 사용하는 경우(SAF 값이 `pushtoclient.properties`의 지시문에 대해 지정됨), Developer for System z는 표 37에 나열된 프로파일에 대한 액세스 허가를 확인하여 사용자가 속하는 개발자 그룹과 사용자가 업데이트를 거부할 수 있는지 여부를 결정합니다.

표 37. 클라이언트로 푸시 SAF 정보

FACILITY 프로파일	고정 길이	필수 액세스 권한	결과
FEK.PTC.CONFIG.ENABLED. sysname.devgroup	23	READ	클라이언트가 지정된 그룹에 대한 구성 업데이트를 허용합니다.
FEK.PTC.PRODUCT.ENABLED. sysname.devgroup	24	READ	클라이언트가 지정된 그룹에 대한 제품 업데이트를 허용합니다.
FEK.PTC.REJECT.CONFIG. UPDATES.sysname	30	READ	사용자가 구성 업데이트를 거부할 수 있습니다.
FEK.PTC.REJECT.PRODUCT. UPDATES.sysname	31	READ	사용자가 제품 업데이트를 거부할 수 있습니다.

참고: Developer for System z에서는 보안 소프트웨어가 사용자에게 프로파일에 대한 액세스 권한이 있는지 여부를 결정할 수 없음을 나타내는 경우 사용자에게 액세스 권한이 없는 것으로 가정합니다. 예를 들어, 프로파일이 정의되지 않는 경우입니다.

`devgroup` 값은 특정 개발자 그룹에 지정된 그룹 이름을 비교합니다. 그룹 이름은 Developer for System z 클라이언트에서 표시됩니다.

`sysname` 값은 대상 시스템의 시스템 이름을 비교합니다.

"Fixed length" 열에는 관련 보안 프로파일의 고정 파트 길이가 기록됩니다.

기본적으로 Developer for System z는 FEK.* 프로파일이 FACILITY 보안 클래스에 있는 것으로 예상합니다. FACILITY 클래스의 프로파일은 39자로 제한됩니다. 고정 프로파일 파트 (FEK.PTC.<key>.)의 길이와 사이트별 프로파일 파트(sysname 또는 sysname.devgroup)의 길이 합계가 이 숫자를 초과하는 경우에는 프로파일을 다른 클

래스에 배치하고 Developer for System z이 대신 이 클래스를 사용하도록 지시할 수 있습니다. 이를 수행하려면 rsed.envvars에서 _RSE_FEK_SAF_CLASS의 주석을 해제하고 원하는 클래스 이름을 제공합니다.

샘플 설정

CDFMVS08 시스템에 Developer for System z가 활성화된 회사가 있는 것으로 가정합니다. RACF 보안 데이터베이스를 여러 시스템이 공유하고 보안 데이터베이스에 다음 그룹이 정의됩니다.

- DEVBANK : 은행 애플리케이션 작업을 수행하는 개발자
- DEVINSUR : 보험 애플리케이션 작업을 수행하는 개발자
- RDZADMIN : Developer for System z 클라이언트 관리자

각 개발자 그룹에는 특정 클라이언트 구성 파일이 필요하며 모든 개발자는 동일한 클라이언트 버전의 제어를 받을 수 있습니다. 클라이언트 관리자와 달리 개발자는 클라이언트로 푸시가 제공하는 변경사항을 거부할 수 없습니다. 거부 규칙은 향후 확장에 대비하여 모든 시스템에 올바릅니다.

클라이언트와 보안 관리자는 구성 업데이트에 클라이언트로 푸시 그룹 이름 BANKING과 INSURANCE를 사용하는 데 동의합니다.

보안 정의

```
# RDZADMIN 및 DEVBANK를 허용하여 클라이언트로 푸시 그룹 BANKING 선택
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)
```

```
# RDZADMIN 및 DEVINSUR를 허용하여 클라이언트로 푸시 그룹 INSURANCE 선택
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)
```

```
# RDZADMIN은 모든 시스템에서 구성 업데이트를 거부할 수 있음
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)
```

```
# RDZADMIN은 모든 시스템에서 제품 업데이트를 거부할 수 있음
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)
```

```
# 활성화 변경
SETROPTS RACLIST(FACILITY) REFRESH
```

pushtoclient.properties

```
# BANKING 및 INSURANCE에는 다른 구성 요구가 있음
config.enabled=SAF
# 모두가 제품 업데이트를 받음
product.enabled=TRUE
# RDZADMIN만 구성 업데이트를 거부할 수 있음
reject.config.updates=SAF
# RDZADMIN만 제품 업데이트를 거부할 수 있음
reject.product.updates=SAF
```

rsed.envvars

기본값을 사용하므로 업데이트가 필요하지 않습니다.

```
_RSE_FEK_SAF_CLASS=FACILITY
```

/var/rdz/pushtoclient/*install

BANKING, INSURANCE 그룹에 대한 작업공간 구성을 내보내는 동안 내보내기 마법사가 /var/rdz/pushtoclient/grouping/<devgroup>/ 디렉토리와 배경 디렉토리 구조를 작성합니다.

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

개별 제품 업그레이드 시나리오가 없으므로 클라이언트 관리자가 /var/rdz/pushtoclient/grouping/<devgroup>/의 install/, install/responsefiles/ 서브디렉토리를 작성 또는 업데이트하지 않아도 됩니다.

클라이언트 관리자는 기본 그룹 디렉토리, /var/rdz/pushtoclient/install/responsefiles/에서 제품 업데이트에 필요한 응답 파일을 작성해야 합니다.

변경사항 거부 유예 기간

샘플 설정이 활성화되어 있는 동안 중요 수정사항이 포함된 Developer for System z 수정팩을 사용할 수 있게 되지만 은행 업무 프로젝트의 일정 계획 상 여러 개발자가 워크스테이션을 즉시 변경하기를 꺼려할 수 있는 것으로 가정합니다.

문제를 해결하기 위해 보안 관리자가 모든 DEVBANK 개발자에게 업데이트 연기(거부)를 선택할 수 있는 유예 기간을 부여할 수 있습니다.

유예 기간 설정은 다음과 같이 매우 간단한 프로세스입니다.

```
# 유예 기간 시작
PERMIT FEK.PTC.REJECT.PRODUCT.UPDATES.* CLASS(FACILITY) -
    ID(DEVBANK) ACCESS(READ)
```

```
# 활성화 변경
SETROPTS RACLIST(FACILITY) REFRESH
```

유예 기간 끝에는 다음과 같이 추가 권한을 다시 제거할 수 있습니다.

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

# 활성화 변경
SETROPTS RACLIST(FACILITY) REFRESH
```

호스트 기반 프로젝트

클라이언트의 z/OS 프로젝트 퍼스펙티브를 통해 개별적으로 z/OS 프로젝트를 정의하거나 z/OS 프로젝트를 호스트에서 중앙 집중식으로 정의하여 개별 사용자별로 클라이언트에 전파할 수 있습니다. 이러한 "호스트 기반 프로젝트"의 모양과 기능은 해당 구조, 멤버, 특성을 클라이언트가 수정할 수 없고 호스트에 연결한 상태에서만 액세스할 수 있다는 점을 제외하고는 클라이언트에 정의된 프로젝트에 일치합니다.

호스트 기반 프로젝트의 기본 디렉토리는 클라이언트 관리자가 /var/rdz/pushtoclient/keymapping.xml에 정의하며 기본값은 /var/rdz/pushtoclient/projects입니다.

호스트 기반 프로젝트를 구성하려면 프로젝트 관리자 또는 리드 개발자가 다음과 같은 유형의 구성 파일을 정의해야 합니다. 모든 파일은 UTF-8 인코드 XML 파일입니다.

- 프로젝트 인스턴스 파일은 단일 사용자 ID에만 해당되며 재사용 가능 프로젝트 정의 파일을 가리킵니다. 각각의 프로젝트를 다운로드하려면 호스트 기반 프로젝트 작업을 수행하는 각 사용자에게 하나의 프로젝트 인스턴스 파일(*.hbpin)을 포함하는 서브디렉토리, /var/rdz/pushtoclient/projects/<userid>/가 필요합니다.
- 프로젝트 정의 파일은 프로젝트의 구조와 콘텐츠를 정의합니다. 이 파일은 여러 사용자가 재사용할 수 있습니다. 프로젝트 정의 파일(*.hbppd)은 프로젝트에 포함되는 서브프로젝트가 나열되며 이 파일은 루트 프로젝트 정의 디렉토리 또는 그 서브디렉토리 중 하나에 있습니다.
- 서브프로젝트 정의 파일은 서브프로젝트의 구조와 콘텐츠를 정의합니다. 이 파일은 여러 사용자가 재사용할 수 있습니다. 서브프로젝트 정의 파일(*.hbpsd)은 단일 로드 모듈을 빌드하는 데 필요한 자원 세트를 정의합니다. 이러한 파일은 루트 프로젝트 정의 디렉토리 또는 그 서브디렉토리 중 하나에 있습니다.
- 서브프로젝트 특성 파일은 변수 대체 지원을 포함하는 특성 파일이며 여러 서브프로젝트에서 재사용할 수 있습니다. 서브프로젝트 특성 파일(*.hbppr)은 변수 대체를 지원하므로 여러 사용자가 특성 파일을 공유할 수 있습니다. 이러한 파일은 루트 프로젝트 정의 디렉토리 또는 그 서브디렉토리 중 하나에 있습니다.

호스트 기반 프로젝트는 또한 140 페이지의 『복수 개발자 그룹』에서 설명하는 여러 그룹 설정에 참여할 수 있습니다. 이러한 적격성은 호스트 기반 프로젝트를 /var/rdz/pushtoclient/grouping/<devgroup>/projects/에도 정의할 수 있음을 의미합니다.

작업공간이 특정 그룹에 바인드되고 이 그룹과 기본 그룹의 사용자에게 대한 프로젝트 정의가 있는 경우, 사용자는 기본 그룹과 특정 그룹에서 모두 프로젝트 정의를 수신합니다.

제 8 장 CICSTS 고려사항

일반적으로 CICS에 자원을 정의하는 역할은 CICS 관리자의 영역이었습니다. 다음과 같은 여러 가지 이유로 애플리케이션 개발자가 CICS 자원을 정의할 수 있게 하는 것을 꺼려왔습니다.

- 대부분의 CICS 자원 정의에는 많은 매개변수가 있는데, 매개변수가 복잡하기 때문에 다른 자원 정의와의 상호 관계, 작업장 표준을 올바르게 정의하려면 CICS 관리자의 지식이 필요합니다. 잘못 정의하면 전체 CICS 리전에 영향을 주는 예상치 못한 결과를 초래할 수 있습니다.
- 대부분의 고객 작업장은 여러 애플리케이션 그룹과 개발자가 공유하여 사용할 수 있어야 하는 CICS 개발 및 테스트 환경을 제공합니다. 많은 고객 작업장에는 이러한 환경에 대한 서비스 레벨 계약이 있습니다. 이러한 계약을 충족시키려면 환경을 엄격히 제어해야 합니다.

Developer for System z는 CICS 관리자가 CICS 자원 정의 기본값을 제어하고 애플리케이션 배치 관리자의 일부인 CICS 자원 정의(CRD) 서버를 통해 CICS 자원 정의 매개변수의 표시 특성을 제어할 수 있게 하여 이러한 문제점을 해결합니다.

예를 들어, CICS 관리자는 애플리케이션 개발자가 업데이트할 수 없는 특정 CICS 자원 정의 매개변수를 제공할 수 있습니다. 제공된 기본값을 사용하거나 사용하지 않고 다른 CICS 자원 정의 매개변수를 업데이트하거나 CICS 자원 정의 매개변수를 숨겨 불필요한 복잡도를 방지할 수 있습니다.

애플리케이션 개발자는 CICS 자원 정의에 만족하면 실행 중인 CICS 테스트 환경에 자원 정의를 즉시 설치하거나 CICS 관리자의 추가 편집 및 승인을 위해 Manifest에 정의를 내보낼 수 있습니다. CICS 관리자는 관리 유틸리티(일괄처리 유틸리티) 또는 Manifest 처리 도구를 사용하여 자원 정의 변경사항을 구현할 수 있습니다.

참고: Manifest 처리 도구는 IBM CICS Explorer 플러그인입니다.

호스트 시스템에서 애플리케이션 배치 관리자를 설정하는 데 필요한 태스크에 대한 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "(선택사항) 애플리케이션 배치 관리자"를 참조하십시오.

애플리케이션 배치 관리자를 사용자 정의하면 Developer for System z에 다음 서비스가 추가됩니다.

- (클라이언트에서) IBM CICS Explorer®는 CICS 자원을 보고 관리하기 위한 Eclipse 기반 인프라를 제공하며 CICS 도구 간의 탁월한 통합을 가능하게 합니다.
- (클라이언트에서) CICS 자원 정의(CRD) 편집기

- (호스트에서) CICS 애플리케이션으로 실행되는 CICS 자원 정의(CRD) 서버

애플리케이션 배치 관리자 CICS 자원 정의(CRD) 서버는 CRD 서버 자체, CRD 저장소, 연관된 CICS 자원 정의, 웹 서비스 인터페이스를 사용하는 경우 웹 서비스 바인드 파일, 샘플 파이프라인 메시지 핸들러로 구성됩니다. CRD 서버는 Developer for System z 문서에서 CICS 1차 연결 리전으로 참조된 웹 소유 리전(WOR)에서 실행되어야 합니다.

현재 Developer for System z 릴리스에서 사용 가능한 애플리케이션 배치 관리자 서비스에 대해 자세히 알려면 Developer for System z Information Center(<http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp>)를 참조하십시오.

RESTful 대 웹 서비스

CICS 트랜잭션 서버는 버전 4.1 이상의 경우 RESTful(Representational State Transfer) 원칙을 사용하여 디자인된 HTTP 인터페이스에 대한 지원을 제공합니다. 이 RESTful 인터페이스는 클라이언트 애플리케이션이 사용하는 전략적 CICSTS 인터페이스입니다. 이전 웹 서비스 인터페이스는 안정화되었으며 기능 보강은 RESTful 인터페이스에만 해당됩니다.

애플리케이션 배치 관리자는 이 지시 명령문을 따르며 Developer for System 버전 7.6 이상의 새 서비스 모두에 RESTful CRD 서버가 필요합니다.

RESTful, 웹 서비스 인터페이스는 원하는 경우 단일 CICS 리전에서 동시에 활성화될 수 있습니다. 이러한 경우 리전에서 두 CRD 서버가 활성화됩니다. 두 서버 모두 동일한 CRD 저장소를 공유합니다. CICS는 두 번째 인터페이스가 리전에 정의될 때 중복 정의에 대한 경고를 발행합니다.

기본 대 비1차 연결 리전

CICS 테스트 환경은 몇 개의 MRO(Multi-Region Option) 연결 리전으로 구성될 수 있습니다. 시간이 지남에 따라 이러한 리전을 카테고리화하는 데 비공식적인 명칭이 사용되어 왔습니다. 일반적인 명칭은 터미널 소유 리전(TOR), 웹 소유 리전(WOR), 애플리케이션 소유 리전(AOR), 데이터 소유 리전(DOR)입니다.

웹 소유 리전은 CICS 웹 서비스 지원을 구현하는 데 사용되며, 애플리케이션 배치 관리자 CICS 자원 정의(CRD) 서버는 이 리전에서 실행되어야 합니다. 이 리전은 애플리케이션 배치 관리자에 CICS 1차 연결 리전으로 알려져 있습니다. CRD 클라이언트는 CICS 1차 연결 리전과의 웹 서비스 연결을 구현합니다.

CICS 비1차 연결 리전은 CRD 서버가 서비스를 제공할 수 있는 기타 모든 리전입니다. 이 서비스에는 IBM CICS Explorer를 사용한 자원 보기, CICS 자원 정의 편집기를 사용한 자원 정의가 포함됩니다.

CICSplex® SM BAS(Business Application Services)를 사용하여 CICS 1차 연결 리전의 CICS 자원 정의를 관리하는 경우, CRD 서버는 BAS가 관리하는 기타 모든 CICS 리전에 서비스를 제공할 수 있습니다.

BAS가 관리하지 않는 CICS 리전은 CRD 서버가 서비스할 수 있도록 추가로 변경해야 합니다.

CICS 자원 설치 로깅

CRD 서버가 CICS 자원에 대해 수행한 조치는 CICS CSDL TD 큐(일반적으로 CICS 리전의 DD MSGUSR을 가리킴)에 로그됩니다.

CICSplex SM BAS(Business Application Services)를 사용하여 CICS 자원 정의를 관리하는 경우, 로깅을 작성하려면 CICSplex SM EYUPARM 지시문 BASLOGMSG를 (YES)로 설정해야 합니다.

애플리케이션 배치 관리자 보안

CRD 저장소 보안

CRD 서버 저장소 VSAM 데이터 세트는 모든 기본 자원 정의를 보관하므로 업데이트로부터 보호해야 하지만 개발자는 여기에 저장된 값을 읽을 수 있어야 합니다. CRD 저장소 보호를 위한 샘플 RACF 명령은 58 페이지의 『데이터 세트 프로파일 정의』를 참조하십시오.

파이프라인 보안

웹 서비스 인터페이스를 통해 CICS가 SOAP 메시지를 수신하면 파이프라인을 통해 메시지가 처리됩니다. 파이프라인은 순서대로 실행되는 메시지 핸들러 세트입니다. CICS는 파이프라인 구성 파일을 읽어 파이프라인에서 호출되어야 하는 메시지 핸들러를 결정합니다. 메시지 핸들러는 웹 서비스 요청 및 응답에 대한 특수 처리를 수행할 수 있는 프로그램입니다.

애플리케이션 배치 관리자는 메시지 핸들러 및 SOAP 헤더 처리 프로그램 호출을 지정하는 샘플 파이프라인 구성 파일을 제공합니다.

파이프라인 메시지 핸들러(ADNTMSGH)는 SOAP 헤더의 사용자 ID와 비밀번호를 처리하여 보안에 사용됩니다. ADNTMSGH는 샘플 파이프라인 구성 파일에서 참조되므로 CICS RPL 연결에 배치해야 합니다.

트랜잭션 보안

CPIH는 파이프라인에서 호출한 애플리케이션이 실행될 기본 트랜잭션 ID입니다. 일반적으로 CPIH는 최소 권한 레벨에 대해 설정됩니다.

Developer for System z는 CICS 자원을 정의하고 조회할 때 CRD 서버가 사용하는 다중 트랜잭션을 제공합니다. 요청된 조작에 따라 CRD 서버가 이러한 트랜잭션 ID를 설정합니다. 트랜잭션 ID 사용자 정의에 대한 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "(선택사항) 애플리케이션 배치 관리자"를 참조하십시오.

트랜잭션	설명
ADMS	Manifest 처리 도구로부터 CICS 자원 변경 요청의 경우. 일반적으로 CICS 관리자를 대상으로 합니다. 이 트랜잭션에는 상호 레벨 권한이 필요합니다.
ADMI	CICS 자원을 정의, 설치 또는 설치 제거하는 요청의 경우. 이 트랜잭션에는 사이트 정책에 따라 중간 레벨의 권한이 필요합니다.
ADMR	CICS 환경 또는 자원 정보를 검색하는 기타 모든 요청의 경우. 이 트랜잭션에는 사이트 정책에 따라 최소 레벨의 권한이 필요합니다.

CRD 서버 트랜잭션이 수행하는 자원 정의 요청의 일부 또는 전부를 보안해야 합니다. CICS 관리자를 제외한 모든 사용자가 글로벌 자원 기본값을 설정하는 데 사용되는 이러한 명령을 실행하지 못하도록 최소한 업데이트 명령(기본 웹 서비스 매개변수, 기본 디스크립터 매개변수, 데이터 세트 이름에 파일 이름 바인딩 업데이트)을 보안해야 합니다.

트랜잭션에 접속되면, CICS 자원 보안 검사(사용 가능한 경우)가 사용자 ID에 트랜잭션 ID를 실행할 수 있는 권한이 있는지 확인합니다.

자원 검사는 실행 중인 트랜잭션의 RESSEC 옵션, RESSEC 시스템 초기화 매개변수, XPCT 시스템 초기화 매개변수(CRD 서버의 경우)를 사용하여 제어됩니다.

자원 검사는 XPCT 시스템 초기화 매개변수 값이 NO가 아닌 경우와 TRANSACTION 정의의 RESSEC 옵션이 YES이거나 RESSEC 시스템 초기화 매개변수가 ALWAYS인 경우에 만 발생합니다.

다음 RACF 명령은 CRD 서버 트랜잭션을 보호할 수 있는 방법에 대한 샘플을 제공합니다. CICS 보안 정의에 대한 자세한 정보는 *RACF Security Guide for CICSTS*를 참조하십시오.

-
- 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)
```

•

```
SETRPTS RACLIST(TCICSTRN) REFRESH
```

SSL 암호화된 통신

애플리케이션 배치 관리자 클라이언트가 웹 서비스 인터페이스를 사용하여 CRD 서버를 호출하는 경우 데이터 스트림 SSL 암호화가 지원됩니다. *RACF Security Guide for CICSTS*에 설명된 대로 이 통신에 SSL 사용은 CICSTS TCIPSERVICE 정의의 SSL(YES) 키워드를 사용하여 제어됩니다.

자원 보안

CICSTS는 자원 보호 기능과 자원 조작 명령을 제공합니다. 보안이 활성화되어 있지만 완전히 구성되어 있지 않으면(예를 들어, 새 자원 유형 조작 권한 부여) 특정 애플리케이션 배치 관리자 조치가 실패합니다.

애플리케이션 배치 관리자에서 기능이 실패하면 CICS 로그를 검사하여 다음과 같은 메시지가 있는지 확인하고 *RACF Security Guide for CICSTS*에 설명된 대로 정정 조치를 수행하십시오.

```
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').
```

관리 유틸리티

Developer for System z는 CICS 관리자가 CICS 자원 정의에 대한 기본값을 제공할 수 있도록 관리 유틸리티를 제공합니다. 이 기본값은 읽기 전용이거나 애플리케이션 개발자가 편집할 수 있습니다.

관리 유틸리티가 제공하는 기능은 다음과 같습니다.

- CICSplex 관리 테스트 환경에 대한 CICSplex 이름
- CICSplex SM 스테이징 그룹 이름
- Manifest 내보내기 규칙 설정
- CICS 자원 속성 기본값 및 표시 권한
- VSAM 데이터 세트 정의에 사용되는 CICS 논리 대 실제 바인딩

관리 유틸리티는 FEK.#CUST.JCL 데이터 세트의 샘플 작업 ADNJSPAU에 의해 호출됩니다. 이 유틸리티를 사용하려면 CRD 저장소에 대한 UPDATE 액세스 권한이 필요합니다.

z/OS 시스템 프로그래머가 FEK.SFEKSAMP(FEKSETUP) 작업을 사용자 정의하고 제출할 때 다른 위치를 지정하지 않았다면 ADNJSPAU는 FEK.#CUST.JCL에 있습니다. 자세한 내용은 *Host Configuration Guide* (SC23-7658)의 "사용자 정의 설정"을 참조하십시오.

참고: ADNJSPAU 작업을 실행하기 전에 CICS에서 CRD 저장소를 닫아야 합니다. 작업 완료 후 저장소를 다시 열 수 있습니다. 예를 들어, CICS에 사인온한 후 다음 명령을 입력하여 파일을 각각 닫고 여십시오.

- CEMT S FILE(ADNREPF0) CLOSED
- CEMT S FILE(ADNREPF0) OPEN

입력 제어문은 CICS 테스트 환경에 대해 CRD 저장소를 업데이트하는 데 사용되는데, 다음 일반 구문 규칙이 적용됩니다.

- 위치 1의 별표는 주석 행을 표시합니다.
- DEFINE 명령은 위치 1에서 시작하고 다음에 하나의 공백이 오고 그 다음에 TRANSACTION과 같은 올바른 키워드가 와야 합니다.
- 키워드 값은 키워드 바로 다음에 와야 합니다. 사이에 공백이 허용되지 않습니다. 유일한 예외는 값이 없는 표시 권한 키워드(UPDATE, PROTECT, HIDDEN)의 경우입니다.
- 키워드 값은 소괄호로 묶어야 합니다.
- 키워드와 키워드 값은 단일 행에 포함되어야 합니다.

다음 샘플 정의는 CICSTS용 CICS 자원 정의 안내서에 정의된 대로 DFHCSDUP 명령 구조를 따릅니다. 유일한 차이점은 속성 값을 세 개의 권한 세트로 그룹화하는 데 사용되는 다음 표시 권한 키워드가 삽입된다는 점입니다.

UPDATE	Developer for System z를 사용하는 애플리케이션 개발자는 이 키워드 다음에 오는 속성을 업데이트할 수 있습니다. 누락된 속성에 대한 기본값이기도 합니다.
PROTECT	Developer for System z를 사용하는 애플리케이션 개발자는 이 키워드 다음에 오는 속성을 표시하지만 업데이트를 방지합니다.
HIDDEN	Developer for System z를 사용하는 애플리케이션 개발자는 이 키워드 다음에 오는 속성을 표시하지 않지만 업데이트를 방지합니다.

다음 ADNJSPAU 코드 샘플을 참조하십시오.

```

//ADNJSPAU JOB <JOB PARAMETERS>
//*
//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 매개변수
*
DEFINE CPSMNAME( )
*DEFINE STAGINGGROUPNAME(ADMSTAGE)
*
* Manifest 내보내기 규칙
*
DEFINE MANIFESTEXPORTRULE(installOnly)
*
* CICS 자원 정의 기본값
* 업데이트 할 누락된 속성 기본값
*
* DB2TRAN 기본 속성
*
DEFINE DB2TRAN()
    UPDATE DESCRIPTION()
        ENTRY()
        TRANSID()
*
* DOCTEMPLATE 기본 속성
*
DEFINE DOCTEMPLATE()
    UPDATE DESCRIPTION()
        TEMPLATENAME()
        FILE() TSQUEUE() TDQUEUE() PROGRAM() EXITPGM()
        DDNAME(DFHHTML) MEMBERNAME()
        HFSFILE()
        APPENDCRLF(YES) TYPE(EBCDIC)
*
* 파일 기본 속성
*
DEFINE FILE()
    UPDATE DESCRIPTION()
        RECORDSIZE() KEYLENGTH()
        RECORDFORMAT(V) ADD(NO)
        BROWSE(NO) DELETE(NO) READ(YES) UPDATE(NO)
        REMOTESYSTEM() REMOTENAME()
    PROTECT DSNNAME() RLSACCESS(NO) LSRPOOLID(1) STRINGS(1)
        STATUS(ENABLED) OPENTIME(FIRSTREF)
        DISPOSITION(SHARE) DATABUFFERS(2) INDEXBUFFERS(1)
        TABLE(NO) MAXNUMRECS(NOLIMIT)
        READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS)
        UPDATEMODEL(LOCKING) LOAD(NO)
        JNLREAD(NONE) JOURNAL(NO)
        JNLSYNCREAD(NO) JNLUPDATE(NO)
        JNLADD(NONE) JNLSYNCWRITE(YES)
        RECOVERY(NONE) FWDRECOVLOG(NO)
        BACKUPTYPE(STATIC)
        PASSWORD() NSRGROUP()
        CFDTPOOL() TABLENAME()

```

```

*
* Mapset 기본 속성
*
DEFINE MAPSET()
    UPDATE DESCRIPTION()
    PROTECT RESIDENT(NO) STATUS(ENABLED)
    USAGE(NORMAL) USELPACOPY(NO)
** Processtype 기본 속성
*
DEFINE PROCESSTYPE()
    UPDATE DESCRIPTION()
    FILE(BTS)
    PROTECT STATUS(ENABLED)
    AUDITLOG() AUDITLEVEL(OFF)
*
* 프로그램 기본 속성
*
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)
*
* TDQueue 기본 속성
*
DEFINE TDQUEUE()
    UPDATE DESCRIPTION()
    TYPE(INTRA)
* 외부 파티션 매개변수
    DDNAME() DSNNAME()
    REMOTENAME() REMOTESYSTEM() REMOTELLENGTH(1)
    RECORDSIZE() BLOCKSIZE(0) RECORDFORMAT(UNDEFINED)
    BLOCKFORMAT() PRINTCONTROL() DISPOSITION(SHR)
* 내부 파티션 매개변수
    FACILITYID() TRANSID() TRIGERRLEVEL(1)
    USERID()
* 간접 매개변수
    INDIRECTNAME()
    PROTECT WAIT(YES) WAITACTION(REJECT)
* 외부 파티션 매개변수
    DATABUFFERS(1)
    SYSOUTCLASS() ERROROPTION(IGNORE)
    OPENTIME(INITIAL) REWIND(LEAVE) TYPEFILE(INPUT)
* 내부 파티션 매개변수
    ATIFACILITY(TERMINAL) RECOVSTATUS(NO)

```

그림 28. ADNJS PAU - CICSTS 관리 유틸리티(2/3)

```

*
* 트랜잭션 기본 속성
*
DEFINE TRANSACTION()
    UPDATE  DESCRIPTION()
            PROGRAM()
            TWASIZE(0)
            REMOTESYSTEM() REMOTENAME() LOCALQ(NO)
    PROTECT PARTITIONSET() PROFILE(DFHCICST)
            DYNAMIC(NO) ROUTABLE(NO)
            ISOLATE(YES) STATUS(ENABLED)
            RUNAWAY(SYSTEM) STORAGECLEAR(NO)
            SHUTDOWN(DISABLED)
            TASKDATAKEY(USER) TASKDATALOC(ANY)
            BREXIT() PRIORITY(1) TRANCLASS(DFHTCL00)
            DTIMOUT(NO) RESTART(NO) SPURGE(NO) TPURGE(NO)
            DUMP(YES) TRACE(YES) CONFDATA(NO)
            OTSTIMEOUT(NO) WAIT(YES) WAITTIME(00,00,00)
            ACTION(BACKOUT) INDOUBT(BACKOUT)
            RESSEC(NO) CMDSEC(NO)
            TRPROF()
            ALIAS() TASKREQ()
            XTRANID() TPNAME() XTPNAME()

*
* URDIMAP 속성
*
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()

*
* VSAM 데이터 세트 이름에 선택적 파일 이름 바인딩
*
*DEFINE DSBINDING() DSNAME()
/*

```

그림 29. ADNJSAPU - CICSTS 관리 유틸리티(3/3)

관리 유틸리티 마이그레이션 참고사항

Developer for System z 버전 7.6.1에서는 관리 유틸리티에 URIMAP 지원을 추가했습니다. URIMAP 지원을 사용할 수 있으려면 CRD 저장소 VSAM 데이터 세트의 최대 레코드 크기를 3000으로 할당해야 합니다. Developer for System z 버전 7.6.1까지 샘플 CRD 저장소 할당 작업에서는 최대 레코드 크기를 2000으로 사용합니다.

이전 CRD 저장소를 사용 중인 경우 URIMAP 지원을 사용하려면 다음 단계를 따르십시오.

1. 기존 CRD 저장소(FEK.#CUST.ADNREPF0)의 백업을 작성하십시오.
2. 기존 CRD 저장소를 삭제하십시오.
3. FEK.SFEKSAMP(ADNVCRD) 작업을 사용자 정의하고 제출하여 새 CRD 저장소를 할당하고 초기화하십시오. 사용자 정의 지시사항은 멤버 내 문서를 참조하십시오.
4. FEK.SFEKSAMP(ADNJSPAU) 작업을 사용자 정의하고 제출하여 관리 유틸리티를 사용하여 새 CRD 저장소를 채우십시오.

참고:

- 관리 유틸리티는 실행될 때마다 CRD 저장소의 전체 콘텐츠를 대체하기 때문에 기존 CRD 저장소 마이그레이션은 필요하지 않습니다.
- CRD 저장소와의 버전 호환성 문제는 없습니다. 지원되는 모든 Developer for System z 클라이언트 및 호스트 코드는 양쪽 최대 레코드 크기와 작동합니다. 그러나 최대 레코드 크기가 3000이 아니면 URIMAP 지원을 사용할 수 없습니다.

관리 유틸리티 메시지

관리 유틸리티는 SYSPRINT DD에 다음 메시지를 발행합니다. CRAZ1803E, CRAZ1891E, CRAZ1892E, CRAZ1893E 메시지에는 파일 상태, VSAM 리턴, VSAM 기능 및 VSAM 피드백 코드가 포함되어 있습니다. VSAM 리턴, 기능 및 피드백 코드는 *DFSMS Macro Instructions for Data Sets*(SC26-7408)에 설명되어 있습니다. 파일 상태 코드는 *Enterprise COBOL for z/OS Language Reference*(SC27-1408)에 설명되어 있습니다.

CRAZ1800I

<마지막 제어문 행 번호>행에서 완료되었습니다.

설명: 시스템 프로그래머 관리 유틸리티가 완료되었습니다.

사용자 응답: 없음.

CRAZ1801W

<마지막 제어문 행 번호>행에서 완료되고 경고가 표시되었습니다.

설명: 시스템 프로그래머 관리 유틸리티가 완료되었고 제어문 처리 시 하나 이상의 경고가 표시되었습니다.

사용자 응답: 기타 경고 메시지를 확인하십시오.

CRAZ1802E

<행 번호>행에서 오류가 발생했습니다.

설명: 시스템 프로그래머 관리 유틸리티에 심각한 오류가 발생했습니다.

사용자 응답: 기타 경고 메시지를 확인하십시오.

CRAZ1803E

저장소 열기 오류, 상태=<파일 상태 코드> RC=<VSAM 리턴 코드>
FC=<VSAM 기능 코드> FB=<VSAM 피드백 코드>

설명: CRD 저장소를 여는 중에 시스템 프로그래머 관리 유틸리티에 심각한 오류가 발생했습니다.

사용자 응답: VSAM 상태, 리턴, 기능 및 피드백 코드를 확인하십시오.

CRAZ1804E

<행 번호>행에 인식할 수 없는 입력 레코드가 있습니다.

설명: 시스템 프로그래머 관리 유틸리티가 인식할 수 없는 입력 제어문을 발견했습니다.

사용자 응답: **DEFINE** 명령 다음에 하나의 공백이 오고, 그 다음에 CPSMNAME, STAGINGGROUPNAME, MANIFESTEXPORTRULE, DSBINDING, DB2TRAN, DOCTEMPLATE, FILE, MAPSET, PROCESSTYPE, PROGRAM, TDQUEUE 또는 TRANSACTION 키워드가 오는지 확인하십시오.

CRAZ1805E

<행 번호>행에서 <키워드> 키워드를 처리 중입니다.

설명: 시스템 프로그래머 관리 유틸리티가 **DEFINE** 키워드 입력 제어문을 처리 중입니다.

사용자 응답: 없음.

CRAZ1806E

<행 번호>행에 올바르지 않은 **Manifest** 내보내기 규칙이 있습니다.

설명: 시스템 프로그래머 관리 유틸리티가 올바르지 않은 **Manifest** 내보내기 규칙을 발견했습니다.

사용자 응답: MANIFESTEXPORTRULE 키워드 값이 "installOnly", "exportOnly" 또는 "both"인지 확인하십시오.

CRAZ1807E

<행 번호>행에서 **DSNAME** 키워드가 누락되었습니다.

설명: 시스템 프로그래머 관리 유틸리티가 **DSNAME** 키워드가 누락된 **DEFINE** **DSBINDING** 제어문을 처리 중이었습니다.

사용자 응답: DEFINE DSBINDING 제어문에 DSNNAME 키워드가 포함되어 있는지 확인하십시오.

CRAZ1808E

<행 번호>행에서 <키워드> 키워드의 키워드 값이 올바르지 않습니다.

설명: 시스템 프로그래머 관리 유틸리티가 DEFINE 제어문을 처리 중이었고 이름 지정된 키워드에 올바르지 않은 값이 발생했습니다.

사용자 응답: 이름 지정된 키워드의 길이와 값이 올바른지 확인하십시오.

CRAZ1890W

<행 번호>행에 키워드 구문 오류가 있습니다.

설명: 시스템 프로그래머 관리 유틸리티가 DEFINE 제어문을 처리 중이었고 키워드 또는 키워드 값에 구문 오류가 발생했습니다.

사용자 응답: 키워드 값이 소괄호로 묶여 있고 키워드 바로 다음에 오는지 확인하십시오. 키워드와 키워드 값은 둘 다 동일한 행에 포함되어야 합니다.

CRAZ1891W

저장소 중복 키 쓰기 오류, 상태=<파일 상태 코드> RC=<VSAM 리턴 코드>
FC=<VSAM 기능 코드> FB=<VSAM 피드백 코드>

설명: CRD 저장소에 기록하는 중에 시스템 프로그래머 관리 유틸리티가 중복 키 오류를 발견했습니다.

사용자 응답: VSAM 상태, 리턴, 기능 및 피드백 코드를 확인하십시오.

CRAZ1892W

저장소 쓰기 오류, 상태=<파일 상태 코드> RC=<VSAM 리턴 코드>
FC=<VSAM 기능 코드> FB=<VSAM 피드백 코드>

설명: CRD 저장소에 기록하는 중에 시스템 프로그래머 관리 유틸리티에 심각한 오류가 발생했습니다.

사용자 응답: VSAM 상태, 리턴, 기능 및 피드백 코드를 확인하십시오.

CRAZ1893W

저장소 읽기 오류, 상태=<파일 상태 코드> RC=<VSAM 리턴 코드>
FC=<VSAM 기능 코드> FB=<VSAM 피드백 코드>

설명: CRD 저장소에서 읽는 중에 시스템 프로그래머 관리 유틸리티에 심각한 오류가 발생했습니다.

사용자 응답: VSAM 상태, 리턴, 기능 및 피드백 코드를 확인하십시오.

CICS 트랜잭션 디버깅

CICS 트랜잭션을 디버그하려면, 통합 디버거에 다음 CICS 업데이트가 필요합니다.

- CICS JCL 업데이트:

- 라이브러리가 LINKLIST에 없는 경우 리전의 DFHRPL DD 명령문에서 FEK,SFEKAUTH 로그 라이브러리를 정의하십시오.
- 라이브러리가 LINKLIST에 없는 경우 리전의 STEPLIB DD 명령문에서 SYS1,SIEAMIGE 로그 라이브러리를 정의하십시오.

- CICS CSD 업데이트:

AQECSD 샘플 CSD 업데이트 작업에서 설명한 대로 CICS 리전에 디버거를 정의하십시오. FEK.SFEKSAMP(FEKSETUP) 작업을 사용자 정의 및 제출할 때 z/OS 시스템 프로그래머가 다른 위치를 지정한 경우가 아니면 AQECSD는 FEK.#CUST.JCL에 있습니다. 세부사항은 *Host Configuration Guide*(SC23-7658)의 "사용자 정의 설정"을 참조하십시오.

읽기 전용 메모리에 로드되는 CICS 트랜잭션을 디버그하려면, 통합 디버거에 다음 시스템 업데이트가 필요합니다.

- 시스템에 정의된 통합 디버거 수퍼바이저 호출(SVC). 세부사항은 *Host Configuration Guide*(SC23-7658)의 "PARMLIB 변경사항"을 참조하십시오.
- 문제점 상태(권한 없는) 환경에서 사용되는 경우 SVC에서는 사용자에게 보안 프로파일이 허용되어야 합니다. 자세한 내용은 42 페이지의 『디버그 보안』을 참조하십시오.

하나의 LE(Language Environment) 기반 디버거만이 지정된 CICS 리전에서 활성화될 수 있음을 참고하십시오. LE 기반 디버거의 표시가 지워지면 이는 애플리케이션에 대해 반드시 사용 가능해야 하는 CEEEVDBG 로드 모듈 또는 별명을 제공함을 의미합니다.

제 9 장 사용자 엑시트 고려사항

이 장에서는 종료 루틴을 기록하여 Developer for System z의 기능을 개선하는 데 필요한 정보를 제공합니다.

Developer for System z는 Developer for System z 이벤트 선택 종료점을 제공합니다. 종료점은 함수가 종료 루틴(있는 경우)을 호출하는 함수 처리의 특정 지점입니다. 종료 루틴을 기록하여 추가 처리를 수행할 수 있습니다.

일반적인 대부분의 종료점과 달리 Developer for System z 종료점은 함수 동작 변경을 허용하지 않습니다. 종료 루틴(있는 경우)은 함수가 완료된 후 비동기로 호출됩니다. Developer for System z 처리는 종료 루틴이 종료되기를 기다리지 않으며 완료 상태를 확인하지도 않습니다.

사용자 엑시트 특성

사용자 엑시트 활성화

사용자 엑시트는 `rsed.envvars`의 `_RSE_JAVAOPTS <exit_point>.action` 변수를 사용하여 활성화됩니다. 여기서 `<exit_point>`는 172 페이지의 『사용 가능한 종료점』에 설명된 대로 특정 종료점을 식별하는 키워드입니다.

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

기본적으로 모든 종료점을 사용할 수는 없습니다. 종료점을 사용하려면 주석을 해제하고 사용자 엑시트 루틴의 전체 경로 이름을 지정해야 합니다.

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

기본적으로 RSE 디먼에 지정된 사용자 ID가 제공된 종료 루틴을 실행하는 데 사용됩니다. 사용자 엑시트를 실행하려면 주석을 해제하고 지정된 ID를 사용할 사용자 ID를 지정하십시오. RSE는 지정된 사용자 ID로 전환할 때 비밀번호로 사용될 PassTicket을 생성하기 때문에 비밀번호를 지정하지 않아도 됩니다.

사용자 엑시트 루틴 기록

사용자 엑시트 루틴은 하나 이상의 인수를 가진 z/OS UNIX 셸 명령으로 호출됩니다. 이는 사용자가 개발하는 종료 루틴을 z/OS UNIX 명령행에서 실행할 수 있어야 함을 의미합니다. 일반적인 코딩 기술은 z/OS UNIX 셸 스크립트와 z/OS UNIX REXX exec를 포함하지만 C/C++과 같은 컴파일된 코드도 가능합니다.

z/OS UNIX 셸 스크립트에 대해 자세히 알려면 *UNIX System Services User's Guide*(SA22-7801)를 참조하십시오. REXX 언어에 대한 z/OS UNIX 특정 확장기능에 대해 자세히 알려면 *Using REXX and z/OS UNIX System Services* (SA22-7806)를 참조하십시오.

종료 루틴은 특수 권한을 가진 사용자 ID(예: PassTicket을 생성할 수 있는 RSE 시작된 태스크 사용자 ID)가 실행할 가능성이 높습니다. 따라서 혼란을 방지하려면 종료 루틴에 대한 업데이트 권한을 제한하는 것이 중요합니다. 다음 샘플 z/OS UNIX 명령은 쓰기 권한을 소유자로만 제한하는 반면, 모든 사람이 스크립트를 읽고 실행할 수 있습니다.

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

rsed.envvars의 정의를 사용자 엑시트 루틴에 환경 변수로 사용할 수 있습니다.

RSE는 단일 인수 문자열을 사용하여 사용자 엑시트 루틴을 호출합니다. 인수 문자열은 단일 값이거나 여러 개의 공백으로 구분된 키워드와 값을 보유한 단일 문자열입니다. 자세한 내용은 172 페이지의 『사용 가능한 종료점』을 참조하십시오.

콘솔 메시지

Developer for System z는 콘솔 메시지 ID FEK910I를 사용하여 사용자 엑시트와 관련된 데이터를 표시합니다.

종료 루틴 호출이 다음 콘솔 메시지와 함께 표시됩니다.

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

stdout에 기록된 모든 데이터(셸 스크립트에서는 **echo** 명령, REXX exec에서는 **say** 명령)가 콘솔로 전송됩니다.

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

종료 루틴 종료는 다음 콘솔 메시지와 함께 표시됩니다.

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

가변 사용자 ID를 사용하여 실행

Developer for System z에서는 시작된 태스크 사용자 ID 또는 지정된 사용자 ID를 사용하여 종료 루틴을 실행할 수 있습니다. 그러나 다른 사용자 ID를 사용하여 종료 루틴에서 일부 조치를 실행할 수 있습니다(예: 로그인 종료 루틴에서 클라이언트 사용자 ID 사용). 다음 샘플에 표시된 대로 표준 z/OS UNIX 서비스를 사용하여 이를 수행할 수 있습니다.

z/OS UNIX 셸 스크립트

UNIX System Services Command Reference(SA22-7802)에 설명된 대로 z/OS UNIX는 슈퍼유저 또는 다른 사용자의 권한을 사용할 수 있도록 **su** 명령을 제공합니다. **su** 명령 사용 시 기억해야 할 몇 가지 사항이 있습니다.

- **su** 명령을 실행하는 사용자 ID에 보안 제품의 SURROGAT 클래스에 있는 BPX.SRV.<userid> 프로파일에 대한 READ 권한이 있어야 비밀번호를 지정하지 않고 <userid>가 식별한 사용자 ID로 전환할 수 있습니다.
- **su** 명령은 새 셸을 시작하므로 셸 스크립트의 나머지 명령은 **su** 명령으로 시작된 셸이 종료되어야 실행됩니다. **su** 명령으로 시작된 새 셸에서 스테이지 명령을 실행하려면, 다음 예제에 표시된 대로 **echo** 명령을 사용하여 원하는 명령과 파이프 명령 문자를 작성하여 새 셸에 공급할 수 있습니다. 특수 문자 이스케이프에는 표준 셸 스크립팅 규칙이 적용됩니다.

```
#!/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
```

시작된 태스크 사용자 ID가 이 샘플 로그인 종료를 실행한 결과 다음 콘솔 메시지가 표시됩니다.

```
+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

Using REXX and z/OS UNIX System Services(SA22-7806)에 설명된 대로 z/OS UNIX는 현재 프로세스의 유효 UID를 설정할 수 있도록 **seteuidd** SYSCALL 명령을 제공합니다. **seteuidd** 명령 사용 시 기억해야 할 몇 가지 사항이 있습니다.

- **seteuid** 명령은 MVS 사용자 ID가 아닌 z/OS UNIX UID를 사용합니다. 대상 사용자 ID의 UID를 먼저 결정해야 합니다(**getpwnam** SYSCALL 명령을 사용하여 수행할 수 있음).
- **seteuid** 명령을 실행하는 사용자 ID에 보안 제품의 SURROGAT 클래스에 있는 BPX.SRV.<userid> 프로파일에 대한 READ 권한이 있어야 비밀번호를 지정하지 않고 <userid>가 식별한 사용자 ID로 전환할 수 있습니다. 다중 사용자 ID가 동일한 UID를 공유하는 경우, 검사할 사용자 ID를 결정하는 방법은 없습니다.

```
/* 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()
```

시작된 태스크 사용자 ID가 이 샘플 로그인 종료를 실행한 결과 다음 콘솔 메시지가 표시됩니다.

```
+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
```

사용 가능한 종료점

Developer for System z가 제공하는 종료점은 다음과 같습니다.

- 『audit.action』
- 173 페이지의 『logon.action』

audit.action

- 타이밍:

감사 사용자 엑시트는 활성 감사 로그 파일이 닫힐 때 호출됩니다(RSE가 새 감사 로그 파일로 전환했기 때문에 감사는 계속됨).

- 호출 인수 (1):

– <audit_log>: 닫힌 감사 로그 파일의 전체 경로 이름

- 샘플:

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

이 샘플 z/OS UNIX REXX exec는 닫힌 감사 로그를 처리할 일괄처리 작업을 빌드합니다.

logon.action

- 타이밍:

로그온 사용자 엑시트는 사용자가 로그인 프로세스를 완료하면 호출됩니다.

- 호출 인수 (6):

- -i <userid>: 클라이언트 사용자 ID(대소문자는 클라이언트가 제공한 그대로임)
- -u <user_log_path>: 이 클라이언트의 사용자 로그가 보관된 디렉토리
- -s <server_log_path>: 서버 로그가 보관된 디렉토리
- -c <config_path>: 구성 파일이 보관된 디렉토리
- -b <binaries_path>: Developer for System z가 설치된 디렉토리
- -p <port>: RSE 디먼 포트

- 샘플:

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

이 샘플 z/OS UNIX 셸 스크립트는 콘솔에 로그인 메시지를 기록합니다.

제 10 장 TSO 환경 사용자 정의

이 장에서는 Developer for System z의 TSO 환경에 DD 문과 데이터 세트를 추가하여 TSO 로그인 프로시저를 모방하는 데 필요한 정보를 제공합니다.

TSO 명령 서비스

TSO 명령 서비스는 TSO 및 (일괄처리) ISPF 명령을 실행하고 그 결과를 요청 클라이언트에 리턴하는 Developer for System z 컴포넌트입니다. 제품이 내재적으로 또는 사용자가 명시적으로 이 명령을 요청할 수 있습니다.

Developer for System z와 함께 제공되는 샘플 멤버는 최소 TSO/ISPF 환경을 작성합니다. 작업장의 개발자가 사용자 정의 또는 써드파티 라이브러리에 액세스해야 하는 경우, z/OS 시스템 프로그래머는 TSO 명령 서비스 환경에 필요한 DD 문과 라이브러리를 추가해야 합니다. Developer for System z에서는 구현이 다르지만 이면의 로직은 TSO 로그인 프로시저와 동일합니다.

참고: TSO 명령 서비스는 비대화식 명령행 도구이므로 데이터를 입력하도록 프롬프트를 표시하거나 ISPF 패널을 표시하는 명령이나 프로시저는 작동하지 않습니다. 이를 실행하려면 Developer for System z 클라이언트의 일부인 호스트 연결 에뮬레이터와 같은 3270 에뮬레이터가 필요합니다.

액세스 방법

버전 7.1부터 Developer for System z는 TSO 명령 서비스에 액세스하는 방법에 대한 선택사항을 제공합니다.

- ISPF의 TSO/ISPF Client Gateway 서비스 - 최소 ISPF 서비스 레벨이 필요합니다. 이 방법은 제공된 샘플에서 사용되는 기본 방법입니다.
- APPC 트랜잭션(버전 7.1 이전 릴리스의 경우). 이 방법은 더 이상 더 이상 사용되지 않습니다.

참고:

- ISPF의 TSO/ISPF Client Gateway 서비스는 버전 7.1에서 사용되는 SCLM 개발자 툴킷 기능을 대체합니다.
- Developer for System z의 APPC 사용은 더 이상 사용되지 않음으로 표시됩니다. APPC 관련 정보는 이 책에서 제거되었습니다. 자세한 정보는 Developer for System z 라이브러리(<http://www-01.ibm.com/support/docview.wss?uid=swg27038517>)에 있는 *Using APPC to provide TSO command services*(SC14-7291) 백서를 참조하십시오.

rsed.envvars를 확인하여 버전 7.1 이상의 호스트에 사용되는 액세스 방법을 결정하십시오. 구성 프로세스 중에 기본값을 사용한 경우, rsed.envvars는 /etc/rdz/에 있습니다.

- `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` 문이 없거나 주석 처리된 경우, ISPF의 TSO/ISPF Client Gateway 서비스가 사용됩니다.
- `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` 문이 있고 주석 처리되지 않은 경우, APPC가 사용됩니다.

TSO/ISPF Client Gateway 액세스 방법 사용

ISPF.conf

ISPF.conf 구성 파일(기본적으로 /etc/rdz/에 있음)은 Developer for System z에서 사용하는 TSO/ISPF 환경을 정의합니다. 하나의 활성 ISPF.conf 구성 파일(모든 Developer for System z 사용자가 사용)만 있습니다.

구성 파일의 기본 섹션은 다음 샘플에서와 같이 DD 이름 및 관련 데이터 세트 연결을 정의합니다.

```
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
ispllib=ISP.SISPLOAD
myDD=HLQ1.LLQ1,HLQ2.LLQ2
```

- 각 DD 정의는 정확히 하나의 행을 사용하며(다중 행은 지원되지 않음) 행 길이 제한은 없습니다.
- 정의는 대소문자를 구분하지 않으며 공백은 무시됩니다.
- 주석 행은 별표(*)로 시작합니다.
- DD 이름 다음에는 등호(=)가 오고 등호 다음에는 데이터 세트 연결이 옵니다. 다중 데이터 세트 이름은 쉼표(,)로 구분됩니다.
- 데이터 세트 연결은 나열된 순서로 검색됩니다.
- 데이터 세트는 완전해야 하며 따옴표(')로 묶지 않고 변수를 사용하지 않습니다.
- 모든 데이터 세트는 DISP=SHR을 사용하여 할당됩니다.
- 새 DD 이름을 마음대로 추가할 수 있지만 DD 이름에 대한 (JCL) 규칙을 따라야 하며 ISPF.conf의 다른 구성 매개변수와 충돌할 수 있습니다. 또한 ISPPROF는 TSO/ISPF Client Gateway 서비스에 의해 동적으로 할당됩니다(DISP=NEW,DELETE).

기존 ISPF 프로파일 사용

기본적으로 TSO/ISPF Client Gateway는 TSO 명령 서비스에 대한 임시 ISPF 프로파일을 작성합니다. 그러나 기존 ISPF 프로파일 사본을 사용하도록 TSO/ISPF Client Gateway z에 지시할 수 있습니다. 여기서 키는 `rsed.envvars`의 `_RSE_ISPF_OPTS` 문입니다.

```
#_RSE_ISPF_OPTS="$ _RSE_ISPF_OPTS&ISPPROF=&SYSUID..ISPPROF"
```

이 기능을 사용하려면 명령문을 주석 해제(선행 파운드 기호(#) 문자를 제거)하고 기존 ISPF 프로파일의 완전한 데이터 세트 이름을 제공하십시오.

데이터 세트 이름에서 사용할 수 있는 변수는 다음과 같습니다.

- &SYSUID. - 개발자의 사용자 ID를 대체
- &SYSPREF. - 개발자의 TSO 접두부를 대체
- &SYSNAME. - IEASYMxx parmlib 멤버에 지정된 대로 시스템 이름을 대체

참고:

- "ISPPROF"에 전달된 데이터 세트 이름이 올바르지 않으면 비어 있는 임시 ISPF 프로파일을 대신 사용합니다.
- ISPF 프로파일(임시 및 복사된 프로파일 둘 다)은 세션 종료 시 삭제됩니다. 프로파일 변경사항은 기존 ISPF 프로파일에 병합되지 않습니다.

할당 exec 사용

ISPF.conf의 `allocjob` 문(기본적으로 주석 처리됨)은 사용자 ID에 의한 추가 데이터 세트 할당을 제공하는 데 사용할 수 있는 `exec`를 가리킵니다.

```
*allocjob = ISP.SISPSAMP(ISPZISP2)
```

이 기능을 사용하려면 명령문을 주석 해제(선행 별표(*) 문자를 제거)하고 할당 `exec`에 대한 완전한 참조를 제공하십시오.

- `exec`는 ISPPROF가 할당되고 DD가 ISPF.conf에 정의된 후에 실행되지만 ISPF가 초기화되기 전에 실행됩니다. 할당 `exec`가 이러한 정의를 실행 취소하지 않는지 확인하십시오.
- 1개의 매개변수 즉, 호출자의 사용자 ID가 `exec`에 전달됩니다.
- FEK.SFEKSAMP(FEKSETUP) 작업을 사용자 정의하고 제출할 때 다른 위치를 지정하지 않았다면 샘플 `exec CRAISPRX`는 샘플 라이브러리 `FEK.#CUST.CNTL`에 제공됩니다. 자세한 내용은 *Host Configuration Guide* (SC23-7658)의 "사용자 정의 설정"을 참조하십시오.

참고: ISPF가 초기화되기 전에 `exec`가 호출되기 때문에 **VPUT** 및 **VGET**을 사용할 수 없습니다. 그러나 PDS(E) 또는 VSAM 파일을 사용하여 이러한 기능 자체 구현을 작성할 수 있습니다.

여러 개의 할당 exec 사용

ISPF.conf가 하나의 할당 exec 호출만 지원하긴 하지만 해당 exec가 다른 exec를 호출하는 것에 대한 제한은 없습니다. 매개변수로 전달되는 클라이언트의 사용자 ID는 개인화된 할당 exec를 호출할 수 있습니다. 예를 들어, USERID'.EXEC(ALLOC)' 멤버가 있는지 여부를 확인하고 실행하십시오.

이 테마를 정교하게 변형하면 다음과 같이 기존 TSO 로그인 프로시저를 사용할 수 있습니다.

- 사용자 특정 구성 파일(예: USERID'.FEKPROF')을 읽으십시오.
- 파일에 언급된 로그인 프로시저를 확인하십시오.
- SYS1.PROCLIB에서 언급된 프로시저를 읽고 구문을 분석하여 그 안의 DD 문과 데이터 세트 할당을 찾으십시오.
- 실제 로그인 프로시저와 유사한 방식으로 데이터 세트를 할당하십시오.

Developer for System z 설정이 여러 개인 다중 ISPF.conf 파일

이전 절에 설명된 할당 exec 시나리오가 사용자의 특정 요구를 처리할 수 없는 경우, Developer for System z의 RSE 통신 서버 인스턴스를 여러 개 작성할 수 있습니다(자체 ISPF.conf 파일을 사용하여 각 인스턴스를 작성). 아래 설명된 방법의 주요 결점은 Developer for System z 사용자가 원하는 TSO 환경을 가져오려면 동일 호스트의 여러 서버에 연결해야 한다는 점입니다.

참고: RSE 서버의 두 번째 인스턴스를 작성하려면 구성 파일, 시작 JCL 및 시작된 태스크 정의를 복제하고 업데이트하면 됩니다. 제품을 새로 설치하지 않아도 되며 코드도 복제되지 않습니다.

```
$ 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
-> change: -Ddaemon.log=/var/rdz/logs/tso2
-> change: -Duser.log=/var/rdz/logs/tso2
-> add at the END:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
$ oedit /etc/rdz/tso2/ISPF.conf
-> change: change as needed
```

이전 예제의 명령은 새로 작성된 tso2 디렉토리 변경이 필요한 Developer for System z 구성 파일을 복사합니다. rsed.envvars의 CGI_ISPCONF 변수를 업데이트하여 새

ISPF.conf 홈 디렉토리를 정의해야 하며, daemon.log 및 user.log를 업데이트하여 새 로그 위치(없으면 자동으로 작성됨)를 정의해야 합니다. _RSE_RSED_PORT를 업데이트하면 기존 및 새 RSE 디먼 둘 다 고유 포트 번호를 사용합니다. CLASSPATH를 업데이트하면 RSE가 tso2로 복사되지 않은 구성 파일을 찾을 수 있습니다. 사용자의 요구에 맞게 ISPF.conf 파일 자체를 업데이트할 수 있습니다. 두 인스턴스 간에 ISPF 작업 영역(rsed.envvars의 CGI_ISPWORK 변수)을 공유할 수 있습니다.

이제 남은 것은 새 포트 번호와 새 /etc/rdz/tso2 구성 파일을 사용하는 RSE에 대해 새 시작된 태스크를 작성하는 것입니다. _RSE_RSED_PORT는 rsed.envvars에서 변경되지 않기 때문에 새 시작된 태스크는 새 포트를 시작 인수로 지정해야 합니다.

이 절에 앞서 표시된 조치에 대한 자세한 정보는 *IBM Rational Developer for System z Host Configuration Guide*(SC23-7658)를 참조하십시오.

제 11 장 다중 인스턴스 실행

예를 들어, 업그레이드를 테스트하는 경우와 같이 동일한 시스템에서 여러 Developer for System z 인스턴스를 활성화하려는 경우가 있습니다. 그러나 TCP/IP 포트와 같은 일부 자원은 공유할 수 없으므로 항상 기본값을 적용할 수 있는 것은 아닙니다. 이 부록의 정보를 사용하면 다른 Developer for System z 인스턴스의 공존을 계획할 수 있으며 이후 이 구성 안내서를 사용하여 사용자 정의할 수 있습니다.

2개 이상의 인스턴스 간에 Developer for System z의 특정 파트를 공유할 수 있지만 해당 소프트웨어 레벨이 동일하고 유일한 변경사항이 구성 멤버인 경우를 제외하고는 공유하지 않는 것이 좋습니다. Developer for System z에는 사용자 정의 공간이 충분하여 여러 인스턴스가 겹쳐지지 않으므로 이 기능을 사용하는 것이 좋습니다.

참고:

- FEK와 /usr/lpp/rdz는 제품 설치 중에 사용되는 상위 레벨 규정자와 경로입니다. FEK.#CUST, /etc/rdz, /var/rdz는 제품을 사용자 정의할 때 사용되는 기본 위치입니다. 자세한 정보는 *Host Configuration Guide (SC23-7658)*의 "사용자 정의 설정"를 참조하십시오.
- 제품의 z/OS UNIX 파트를 쉽게 배치하려면 개인 파일 시스템(HFS 또는 zFS)에 Developer for System z를 설치해야 합니다.
- 개인용 파일 시스템을 사용할 수 없는 경우, z/OS UNIX tar 명령과 같은 아카이빙 도구를 사용하여 z/OS UNIX 디렉토리를 시스템에서 시스템으로 전송해야 합니다. 이렇게 하면 Developer for System z 파일과 디렉토리에 대한 속성(예: 프로그램 제어)을 예약할 수 있습니다.

Developer for System z 설치 디렉토리를 아카이브, 복원하기 위한 다음 샘플 명령에 대한 자세한 정보는 *UNIX System Services Command Reference(SA22-7802)*를 참조하십시오.

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

sysplex에서 동일 설정

몇 가지 가이드라인을 따르면 Developer for System z 구성 파일(및 코드)을 sysplex의 여러 시스템(각 시스템은 Developer for System z 자체 동일 복사본을 실행) 간에 공유할 수 있습니다. 이 정보는 독립형 Developer for System z 인스턴스를 대상으로 합니다. 72 페이지의 『분산 동적 VIPA』에 설명된 대로 분산 동적 VIPA를 사용하여 각각 별도 시스템에 있는 다중 서버를 하나의 가상 서버로 그룹화하는 경우 TCP/IP 설정에 대한 추가 규칙이 적용됩니다.

- 하나의 시스템이 다른 시스템의 정보를 겹쳐쓰지 않게 하려면 로그 파일이 고유 위치에서 끝나야 합니다. rsed.envvars의 daemon.log 및 user.log 지시문을 사용하여 z/OS UNIX 로그를 특정 위치로 라우팅하여 지정된 경로에서 시스템 특정 z/OS UNIX 파일 시스템을 마운트하는 경우 구성 파일을 공유할 수 있습니다. 이런 방법으로 모든 로그는 동일한 논리적 위치에 기록되지만 기저에 공유되지 않는 파일 시스템으로 인해 다른 실제 위치에서 끝납니다.
- Developer for System z는 /etc/rdz/ 및 /var/rdz/pushtoclient/와 같은 구성 유형 디렉토리를 읽기 전용 모드에서 사용하기 때문에 sysplex에서 이를 공유할 수 있습니다.
- 임시 파일 이름은 sysplex에서 인식되지 않기 때문에 /tmp/ 및 /var/rdz/WORKAREA/와 같은 임시 데이터 디렉토리는 시스템별로 고유해야 합니다.
- 코드를 공유하는 경우, 유지보수를 적용한 후 일부 시스템이 동기화되지 않는 일이 없도록 구성 파일도 공유해야 합니다.
- 활성 /etc/rdz/pushtoclient.properties 구성 파일을 공유하는 경우, 관련 메타데이터 디렉토리(/var/rdz/pushtoclient/)도 공유해야 합니다.

동일한 소프트웨어 레벨, 다른 구성 파일

일련의 제한된 상황에서는 (일부) 사용자 정의할 수 있는 파트를 제외한 모두를 공유할 수 있습니다. 한 예는 온사이트 사용에는 비SSL 액세스를, 오프사이트 사용에는 SSL 인코딩된 통신을 제공하는 것입니다.

경고: 공유 설정을 안전하게 사용하여 유지보수, 기술적 미리보기 또는 새 릴리스를 테스트할 수는 없습니다.

활성 Developer for System z 설치의 또 다른 인스턴스를 설정하려면, 현재 설정이 겹치지 않도록 서로 다른 데이터 세트, 디렉토리 및 포트를 사용하여 서로 다른 파트에 대해 사용자 정의 단계를 다시 실행하십시오.

앞서 언급한 SSL 샘플에서는 현재 RSE 디먼 설정을 복제할 수 있으며, 그 후에 복제된 설정을 업데이트할 수 있습니다. 그 다음에는 RSE 디먼 시작 JCL을 복제하고 새 TCP/IP 포트와 업데이트된 구성 파일 위치를 사용하여 사용자 정의할 수 있습니다. SSL

인스턴스와 비SSL 인스턴스 간에 MVS 사용자 정의(JES 작업 모니터 등)를 공유할 수 있습니다. 이로 인해 다음 조치가 발생합니다.

```
$ 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
-> change: _RSE_RSED_PORT=4047
-> change: -Ddaemon.log=/var/rdz/logs/ssl
-> change: -Duser.log=/var/rdz/logs/ssl
-> add at the END:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
$ oedit /etc/rdz/ssl/ssl.properties
-> change: change as needed
```

이전 예제의 명령은 새로 작성된 ssl 디렉토리 변경이 필요한 Developer for System z 구성 파일을 복사합니다. rsed.envvars의 daemon.log 및 user.log 변수를 업데이트하여 새 로그 위치(없으면 자동으로 작성됨)를 정의해야 합니다. CLASSPATH를 업데이트하면 RSE가 ssl로 복사되지 않은 구성 파일을 찾을 수 있습니다. 사용자의 요구에 맞게 ssl.properties 파일 자체를 업데이트할 수 있습니다.

이제 남은 것은 새 포트 번호와 새 /etc/rdz/ssl 구성 파일을 사용하는 RSE에 대해 새 시작된 태스크를 작성하는 것입니다.

이 절에 앞서 표시된 조치에 대한 자세한 정보는 *IBM Rational Developer for System z Host Configuration Guide(SC23-7658)*의 관련 절을 참조하십시오.

자동화된 동기화

앞에서 언급한 SSL 샘플에서는 비SSL 및 SSL 사용 RSE 디먼 사이의 변경이 최소화되므로 해당 rsed.envvars 파일의 동기화 유지 프로세스를 자동화할 수 있습니다. 따라서 하나의 rsed.envvars 파일만 유지보수하면 되므로 서비스 롤아웃이 간소화됩니다.

다음 예제는 RSED 포트 번호를 로그 디렉토리 이름에 추가하고 CLASSPATH를 업데이트하여 복제본이 나머지 구성 파일을 찾을 수 있게 합니다. 그런 다음 이 예제는 프로세스에서 포트 번호를 업데이트하여 시작 시 비SSL RSE 디먼의 rsed.envvars를 복제하도록 SSL 사용 RSE 디먼의 시작된 태스크 JCL을 향상시킵니다. 포트 번호는 로그 디렉토리 이름에 임베드되므로 해당 번호는 양 디먼 간에 자동으로 달라집니다.

1. 마스터 rsed.envvars를 준비하십시오.

```
$ oedit /etc/rdz/rsed.envvars
-> change: -Ddaemon.log=/var/rdz/logs/$RSE_RSED_PORT
-> change: -Duser.log=/var/rdz/logs/$RSE_RSED_PORT
-> add at the END:
# -- NEEDED BY CLONES TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

2. 마스터(비SSL)와 복제본(SSL) 간에 다른 (rsed.envvars 파일이 아닌) 기타 구성 파일을 준비하십시오.

```
$ mkdir /etc/rdz/ssl
$ cp /etc/rdz/ssl.properties /etc/rdz/etc/rdz/ssl
$ oedit /etc/rdz/ssl/ssl.properties
-> change: change as needed
```

3. Create an RSED started task that will clone the base rsed.envvars and alter the RSE daemon port (4035 -> 4034).

```
/*
/* RSE DAEMON - SSL
/*
//RSED      PROC IVP=,                * 'IVP' to do an IVP test
//          HOME='/usr/lpp/rdz',
//          CNFG='/etc/rdz/ssl'
/*
//          SET SED='"/RSED_PORT/s/4035/4034/'
//          SET FILE='rsed.envvars'
/*
/* copy /etc/rdz/rsed.envvars to /etc/rdz/ssl/rsed.envvars
/* and alter 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=*
/*
/* start RSED with the newly created 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
/*
```

기타 모든 상황

코드 변경이 관련된 경우(유지보수, 기술적 미리보기, 새 릴리스) 또는 변경사항이 상당히 복잡한 경우에는 또 다른 Developer for System z 설치를 수행해야 합니다. 이 절에서는 여러 설치 간의 가능한 충돌 포인트를 설명합니다.

다음 목록은 Developer for System z 인스턴스 간에 달라야 하거나 다르게 설정할 것을 강력히 권고하는 항목에 대한 간략한 개요입니다.

- SMP/E CSI
- 설치 라이브러리
- JES 작업 모니터 TCP/IP 포트 및 해당 구성 파일 FEJJCNF
- JES 작업 모니터 시작 JCL
- APPC 트랜잭션 이름
- RSE 구성 파일(rsed.envvars, *.properties 및 *.conf)
- RSE TCP/IP 포트
- RSE 시작 JCL

보다 자세한 개요를 설명하면 다음과 같습니다.

- SMP/E CSI
 1. 별도의 CSI에 Developer for System z 인스턴스를 각각 설치하십시오. SMP/E는 CSI에 동일 FMID를 두 번째 설치하는 것은 방지하지만 다른 FMID 설치 허용합니다. 두 번째 FMID가 새 버전이면 제품의 기존 버전을 삭제합니다. 두 번째 FMID가 이전 버전이면 중복된 파트 이름으로 인해 설치에 실패합니다.
- 설치 라이브러리
 1. 별도의 데이터 세트와 디렉토리에 Developer for System z 인스턴스를 각각 설치하십시오. IBM에서 제공한 기본값(/usr/lpp/rdz) 앞에 붙여야만 z/OS UNIX 경로를 변경할 수 있음을 기억하십시오. 올바른 샘플은 /service/usr/lpp/rdz입니다.
 2. 사용자 정의 설정 작업 FEK.SFEKSAMP(FEKSETUP)은 구성 파일을 저장하는 데 사용되는 데이터 세트와 디렉토리를 작성합니다. 구성 파일은 고유해야 하기 때문에 기존 사용자 정의를 겹쳐쓰지 않으려면 이 작업을 제출할 때 고유 데이터 세트와 디렉토리 이름을 사용해야 합니다.
- 필수 파트
 1. JES 작업 모니터 구성 파일 FEK.#CUST.PARMLIB(FEJJCNF)는 JES 작업 모니터의 TCP/IP 포트 번호를 보유하므로 공유할 수 없습니다. 멤버 자체의 이름을 바꿀 수 있으므로(JCL도 업데이트되는 경우) 설치 데이터 세트에서 업데이트를 수행하지 않는 경우 이 멤버의 사용자 정의 버전을 모두 동일한 데이터 세트에 배치할 수 있습니다.
 2. JES 작업 모니터 시작 JCL FEK.#CUST.PROCLIB(JMON)은 FEJJCNF를 참조하므로 공유할 수 없습니다. 멤버(및 사용자 작업으로 시작하는 경우 JOB 카드) 이름을 바꾼 후에 모든 JCL을 동일한 데이터 세트에 배치할 수 있습니다.
 3. RSE 구성 파일 /etc/rdz/rsed.envvars는 설치 경로, 선택적으로 서버 로그 위치(고유해야 함)에 대한 참조를 보유합니다. 파일 이름은 필수이므로 동일한 디렉토리에 서로 다른 사본을 보관할 수 없습니다.

4. ISPF.conf 구성 파일에는 TSO 명령 서버인 FEK.SFEKPROC(FEKFRSRV)에 대한 참조가 있습니다. 이는 소프트웨어 레벨별이므로 인스턴스당 하나의 ISPF.conf 파일을 작성해야 합니다.
5. 기타 모든 z/OS UNIX 기반 구성 파일(예: *.properties)은 rsed.envvars와 동일한 디렉토리에 상주해야 하므로 공유할 수 없습니다. rsed.envvars가 공유되지 않는 위치에 있어야 하기 때문입니다.
6. RSE 시작 JCL FEK.#CUST.PROCLIB(RSED)는 TCP/IP 포트 번호를 정의하고 설치 및 구성 디렉토리(고유해야 함)에 대한 참조가 있기 때문에 공유할 수 없습니다. 멤버(및 사용자 작업으로 시작하는 경우 JOB 카드) 이름을 바꾼 후에 모든 JCL을 동일한 데이터 세트에 배치할 수 있습니다.

- 선택적 파트

1. 제한사항 없이 REXEC 및 SSH TCP/IP 포트를 공유할 수 있습니다.
2. APPC 트랜잭션에는 TSO 명령 서버인 FEK.SFEKPROC(FEKFRSRV)에 대한 참조가 있습니다. 이는 소프트웨어 레벨별이므로 인스턴스당 하나의 APPC 트랜잭션을 작성해야 합니다. APPC 트랜잭션 이름이 변경되기 때문에 _FEKFSCMD_TP_NAME_ 변수를 rsed.envvars에 정의해야 함을 기억하십시오.
3. 일부 ELAXF* 프로시저에는 Developer for System z 로드 라이브러리인 hlq.SFEKLOAD에 대한 참조가 있습니다. 사용자가 여러 세트를 사용할 수 있게 하는 방법에 대한 가능한 솔루션은 *Host Configuration Guide (SC23-7658)*의 "ELAXF* 원격 빌드 프로시저"에서 JCLLIB에 대한 참고사항을 참조하십시오.
4. 두 개의 DB2® 스토어드 프로시저 인스턴스를 활성화하려면 다음 태스크를 완료해야 합니다. 그러나 이는 지원되지 않는 있는 그대로의 설명임을 참고하십시오.
 - a. hlq.SFEKPROC(ELAXMREX)를 이름이 다르게 지정된 멤버(예: ELAXMRXX)로 복사하십시오.
 - b. 샘플 멤버 hlq.SFEKSAMP(ELAXMSAM)을 이름이 다르게 지정된 멤버(예: ELAXMWDZ)로 복사하십시오.
 - c. 샘플 멤버 hlq.SFEKSAMP(ELAXMJCL)을 변경하여 다음 이름 변경을 반영하십시오. 예를 들어, 다음과 같습니다.

```
//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
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. *Host Configuration Guide* (SC23-7658)의 "(선택사항) DB2 스토어드 프로시저"에 설명된 대로 사용자 정의를 진행하되, 새 멤버를 사용하십시오.
 - e. 클라이언트의 DB2 스토어드 프로시저 마법사에서 새 WLM 환경 이름(예: ELAXMWDZ)을 사용해야 합니다.
- 5. CICS 리전에서의 양방향 지원은 로드 라이브러리 멤버에 의존하므로 릴리스 간에 공유할 수 없습니다. 그러나 모든 인스턴스에 로드 모듈 이름이 동일하면 인스턴스 간에, 심지어 릴리스 간에도 최신 버전을 공유할 수 있습니다. 로드 모듈 이름이 변경된 경우에는 역방향 호환이 불가능합니다.
 - 6. CICS 리전에 포함된 애플리케이션 배치 관리자 로드 모듈은 역방향 호환이 가능하므로 릴리스 간에 최신 버전을 공유할 수 있습니다.
 - 7. 애플리케이션 배치 관리자 CRD VSAM은 역방향 호환이 가능하므로 릴리스 간에 최신 버전을 공유할 수 있습니다.
 - 8. 애플리케이션 배치 관리자 CICS 자원 정의는 역방향 호환이 가능하므로 릴리스 간에 최신 버전을 공유할 수 있습니다.
 - 9. CARMA VSAM은 소프트웨어 레벨 간에 변경될 수 있으므로 공유하지 않는 것이 좋습니다.

제 12 장 구성 문제점 해결

이 장은 Developer for System z 구성 중에 발생할 수 있는 몇 가지 일반적인 문제점을 해결하기 위해 제공되며 다음 절로 구성됩니다.

- 190 페이지의 『FEKLOGS를 사용한 로그 및 설정 분석』
- 190 페이지의 『로그 파일』
- 197 페이지의 『덤프 파일』
- 199 페이지의 『추적』
- 202 페이지의 『z/OS UNIX 권한 비트』
- 206 페이지의 『예약된 TCP/IP 포트』
- 207 페이지의 『주소 공간 크기』
- 209 페이지의 『기타 정보』

Developer for System z Messages and Codes(SC14-7497) 서적은 Developer for System z 컴포넌트가 생성하는 메시지와 리턴 코드를 설명합니다. *Developer for System z Answers to common host configuration and maintenance issues*(SC14-7373)는 다양한 문제점 시나리오와 그 해결 방법에 대해 설명합니다.

자세한 정보는 Developer for System z 웹 사이트(<http://www-03.ibm.com/software/products/us/en/developerforsystemz/>)의 지원 섹션을 참조하십시오. 이 섹션에서는 지원팀의 최신 정보를 제공하는 기술 노트가 있습니다.

웹 사이트의 라이브러리 섹션(<http://www-01.ibm.com/support/docview.wss?uid=swg27038517>)에서는 또한 Developer for System z 문서의 최신 버전과 백서가 함께 제공됩니다.

Developer for System z Information Center(<http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp>)는 Developer for System z 클라이언트와 호스트와의 상호 작용 방법을 설명합니다(클라이언트 관점에서).

z/OS 인터넷 라이브러리(<http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/>)에서도 중요한 정보를 제공합니다.

Developer for System z에 추천하고 싶은 기능에 대한 의견을 보내주십시오. 다음에서 RFE(Request For Enhancement)를 열 수 있습니다.

<https://www.ibm.com/developerworks/support/rational/rfe/>

FEKLOGS를 사용한 로그 및 설정 분석

Developer for System z는 모든 z/OS UNIX 로그 파일과 Developer for System z 설치 및 구성 정보를 수집하는 샘플 작업인 FEKLOGS를 제공합니다.

샘플 작업 FEKLOGS는 FEK.#CUST.JCL에 있습니다(FEK.SFEKSAMP(FEKSETUP) 작업을 사용자 정의, 제출할 때 다른 위치를 지정하지 않은 경우). 자세한 내용은 *Host Configuration Guide* (SC23-7658)의 "사용자 정의 설정"을 참조하십시오.

FEKLOGS 사용자 정의는 JCL에서 설명합니다. 사용자 정의에는 몇몇 주요 변수 제공이 포함됩니다.

참고: SDSF 고객은 SDSF의 XDC 행 명령을 사용하여 데이터 세트에 작업 출력을 저장할 수 있으며 데이터 세트는 IBM 지원 센터에 제공됩니다. 레코드가 잘리지 않도록 하기 위해 출력 데이터 세트를 VB 2051(SDSF의 기본값은 VB 240임)로 할당해야 한다는 점에 유의하십시오.

로그 파일

Developer for System z는 사용자와 IBM 지원 센터가 문제점을 식별, 해결하는 데 도움을 줄 수 있는 로그 파일을 작성합니다. 다음 목록은 z/OS 호스트 시스템에서 작성할 수 있는 로그 파일의 개요입니다. 이러한 제품별 로그 다음에는 SYSLOG에서 관련 메시지를 확인해야 합니다.

MVS 기반 로그는 해당 DD 문으로 찾을 수 있습니다. z/OS UNIX 기반 로그 파일은 다음 디렉토리에 있습니다.

- userlog/\$LOGNAME/

사용자별 로그 파일은 userlog/\$LOGNAME/에 있습니다. 여기서 userlog는 rsed.envvars에서 user.log, DSTORE_LOG_DIRECTORY 지시문의 결합 값이고 \$LOGNAME은 로그인 사용자 ID(대문자)입니다. user.log 지시문이 주석 처리되어 있거나 없으면 사용자의 홈 경로가 사용됩니다. 홈 경로는 사용자 ID의 OMVS 보안 세그먼트에 정의됩니다. DSTORE_LOG_DIRECTORY 지시문이 주석 처리되어 있거나 없으면 user.log 값에 .eclipse/RSE/가 추가됩니다.

- .dstoreMemLogging - 데이터 저장소 메모리 사용량 로깅
- .dstoreTrace - 데이터 저장소 조치 로깅
- .dstoreHashmap.* - 활성 데이터 저장소의 스냅샷
- .dstoreStackTrace.* - 활성 데이터 저장소 스택 및 이 항목이 호출된 위치의 스냅샷
- ffs.log - 기본 MVS 기능을 실행하는 외부 파일 시스템(FFS) 서버의 로그
- ffsget.log - 순차 데이터 세트 또는 PDS 멤버를 읽는 파일 리더의 로그

- ffspout.log - 순차 데이터 세트 또는 PDS 멤버를 작성하는 파일 작성자의 로그
- lock.log - 순차 데이터 세트 또는 PDS 멤버를 잠금/잠금 해제하는 잠금 관리자의 로그
- rmt_class_loader.cache.jar - RSE 원격 클래스 로더가 로드하는 클래스 캐시
- rsecomm.log - 클라이언트와 RSE에 의존하는 모든 서비스의 통신 로깅의 명령을 처리하는 RSE 서버의 로그(Java 예외 스택 추적 포함 가능)

참고:

- .eclipse 디렉토리와 .dstore* 로그 파일은 마침표(.)로 시작되어 숨겨져 있습니다. z/OS UNIX 명령 **ls -IA**를 사용하면 숨겨진 파일과 디렉토리를 나열할 수 있습니다. Developer for System z 클라이언트를 사용하는 경우에는 창 > 참조... > 원격 시스템 > 파일 환경 설정 페이지를 선택하고 "숨겨진 파일 표시"를 사용합니다.

• daemon-home

RSE 디먼과 RSE 스레드 풀별 로그 파일은 daemon-home에 있습니다. 여기서 daemon-home은 rsed.envvars에서 daemon.log 지시문의 값입니다. daemon.log 지시문이 주석 처리되어 있거나 없으면 RSED 시작 태스크에 지정된 사용자 ID의 홈 디렉토리를 사용합니다. 홈 디렉토리는 사용자 ID의 OMVS 보안 세그먼트에 저장됩니다.

- rsedaemon.log - RSE 디먼 로그
- rseserver.log - RSE 스레드 풀 로그
- audit.log - RSE 감사 추적
- serverlogs.count - RSE 스레드 풀 스트림 로깅 카운터
- stderr.*.log - RSE 스레드 풀 표준 오류 스트림
- stdout.*.log - RSE 스레드 풀 표준 출력 스트림

• /tmp

IVP 특정 로그 파일(설치 검증 프로그램)은 TMPDIR이 참조하는 디렉토리에 있습니다(이 변수가 rsed.envvars에 정의된 경우). 이 변수가 정의되지 않으면 파일은 /tmp에 작성됩니다.

- fekfivpi.log - fekfivpi IVP 테스트 로그
- fekfivps.log - fekfivps IVP 테스트 로그
- fekfivpc.log - fekfivpc IVP 테스트의 통신 로그

참고: 언급된 로그 파일 중 일부에 기록되는 데이터의 양을 제어할 수 있는 연산자 명령이 있습니다. 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "운영자 명령" 명령을 참조하십시오.

JES 작업 모니터 로깅

- **SYSOUT DD**

정상 오퍼레이션에 대한 로깅입니다. 샘플 JCL FEK.#CUST.PROCLIB(JMON)에서 기본값은 SYSOUT=*입니다.

- **SYSPRINT DD**

추적 로깅입니다. 샘플 JCL FEK.#CUST.PROCLIB(JMON)에서 기본값은 SYSOUT=*입니다. 추적은 -TV 매개변수를 사용하여 활성화됩니다. 자세한 내용은 199 페이지의 『JES 작업 모니터 추적』을 참조하십시오.

RSE 디먼 및 스레드 풀 로깅

- **STDOUT DD**

RSE 디먼의 Java 표준 출력인 stdout의 경로 재지정 데이터. 샘플 JCL FEK.#CUST.PROCLIB(RSED)의 기본값은 SYSOUT=*입니다.

- **STDERR DD**

RSE 디먼의 Java 표준 오류 출력인 stderr의 경로 재지정 데이터. 샘플 JCL FEK.#CUST.PROCLIB(RSED)의 기본값은 SYSOUT=*입니다.

- **daemon-home**

RSE 디먼과 RSE 스레드 풀별 로그 파일은 daemon-home에 있습니다. 여기서 daemon-home은 rsed.envvars에서 daemon.log 지시문의 값입니다. daemon.log 지시문이 주석 처리되어 있거나 없으면 RSED 시작 태스크에 지정된 사용자 ID의 홈 디렉토리를 사용합니다. 홈 디렉토리는 사용자 ID의 OMVS 보안 세그먼트에 저장됩니다.

- rsedaemon.log - RSE 디먼 로그
- rseserver.log - RSE 스레드 풀 로그
- audit.log - RSE 감사 추적
- serverlogs.count - RSE 스레드 풀 스트림 로깅 카운터
- stderr.*.log - RSE 스레드 풀 표준 오류 스트림
- stdout.*.log - RSE 스레드 풀 표준 출력 스트림

참고:

- `serverlogs.count`, `stderr.*.log`와 `stdout.*.log`는 `rse.envvars`의 `enable.standard.log` 지시문이 활성화 상태이거나 **modify rsestandardlog on** 연산자 명령으로 함수가 동적으로 활성화된 경우에만 작성됩니다.
- `stderr.*.log`와 `stdout.*.log`의 *는 기본적으로 1입니다. 그러나 RSE 스레드 풀이 여러 개 있을 수 있고 이 경우 고유 파일 이름을 유지하기 위해 새 RSE 스레드 풀마다 숫자가 증가합니다.
- 언급된 로그 파일 중 일부에 기록되는 데이터의 양을 제어할 수 있는 연산자 명령이 있습니다. 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "운영자 명령" 명령을 참조하십시오.
- `rse.envvars`에 `keep.last.log=true`가 지정된 경우 `rse*.log` 파일은 ".log" 확장자 대신 ".last" 확장자로 존재할 수도 있습니다. 기본적으로 ".last" 로그 파일은 작성되지 않습니다.
- `keep.all.logs=true`가 `rse.envvars`에 지정되어 있으면 `rse*.log` 파일이 확장 이름을 갖습니다. 기본적으로 확장 이름이 사용됩니다. 다음은 샘플 확장 이름입니다: `rseserver.RSED#yyyymmddhhmmss.log`. 여기서 RSED는 RSE 디먼의 주소 공간 이름을 나타내고 yyyymmddhhmmss는 날짜 및 시간소인(년, 월, 일, 시, 분, 초)입니다.

RSE 사용자 로깅

- `userlog/$LOGNAME/`

RSE와 관련된 컴포넌트로 작성되는 로그 파일이 여러 개 있습니다. 모든 파일은 `userlog/$LOGNAME/`에 있습니다. 여기서 `userlog`는 `rse.envvars`에서 `user.log`, `DSTORE_LOG_DIRECTORY` 지시문의 결합 값이고 `$LOGNAME`은 로그인 사용자 ID(대문자)입니다. `user.log` 지시문이 주석 처리되어 있거나 없으면 사용자의 홈 경로가 사용됩니다. 홈 경로는 사용자 ID의 OMVS 보안 세그먼트에 정의됩니다. `DSTORE_LOG_DIRECTORY` 지시문이 주석 처리되어 있거나 없으면 `user.log` 값에 `.eclipse/RSE/`가 추가됩니다.

- `.dstoreMemLogging` - 데이터 저장소 메모리 사용량 로깅
- `.dstoreTrace` - 데이터 저장소 조치 로깅
- `.dstoreHashmap.*` - 활성 데이터 저장소의 스냅샷
- `.dstoreStackTrace.*` - 활성 데이터 저장소 스레드 및 이 항목이 호출된 위치의 스냅샷
- `ffs.log` - 기본 MVS 기능을 실행하는 외부 파일 시스템(FFS) 서버의 로그
- `ffsget.log` - 순차 데이터 세트 또는 PDS 멤버를 읽는 파일 리더의 로그
- `ffsput.log` - 순차 데이터 세트 또는 PDS 멤버를 작성하는 파일 작성자의 로그

- lock.log - 순차 데이터 세트 또는 PDS 멤버를 잠금 또는 잠금 해제하는 잠금 관리자의 로그
- rmt_class_loader.cache.jar - RSE 원격 클래스 로더가 로드하는 클래스 캐시
- rsecomm.log - 클라이언트와 RSE에 의존하는 모든 서비스의 통신 로깅의 명령을 처리하는 RSE 서버의 로그(Java 예외 스택 추적 포함 가능)

참고:

- .eclipse 디렉토리와 .dstore* 로그 파일은 마침표(.)로 시작되어 숨겨져 있습니다. z/OS UNIX 명령 **ls -lA**를 사용하면 숨겨진 파일과 디렉토리를 나열할 수 있습니다. Developer for System z 클라이언트를 사용하는 경우에는 창 > 참조... > 원격 시스템 > 파일 환경 설정 페이지를 선택하고 "숨겨진 파일 표시"를 사용합니다.
- .dstore* 로그 파일 작성은 **-DDSTORE_*** Java 시작 옵션으로 제어됩니다(*Host Configuration Guide* (SC23-7658)의 "_RSE_JAVAOPTS를 사용하여 추가 Java 시작 매개변수 정의" 설명 참조).
- .dstore* 로그 파일은 ASCII로 작성됩니다. z/OS UNIX 명령 **iconv -f ISO8859-1 -t IBM-1047 .dstore***를 사용하면 EBCDIC로 표시됩니다(코드 페이지 IBM-1047을 사용하는 경우).
- 모든 *.log 파일과 달리, .dstore* 로그 파일은 클라이언트 재연결 시 자동으로 제거되지 않습니다. 이 파일의 제거는 수동으로 수행합니다.
- 언급된 로그 파일 중 일부에 기록되는 데이터의 양을 제어할 수 있는 연산자 명령이 있습니다. 자세한 정보는 *Host Configuration Guide* (SC23-7658)의 "운영자 명령" 명령을 참조하십시오.
- rsed.envvars에 keep.last.log=true가 지정된 경우 ffs*.log, lock.log 및 rsecomm.log 파일은 ".log" 확장자 대신 ".last" 확장자로 존재할 수도 있습니다. 기본적으로 ".last" 로그 파일은 작성되지 않습니다.
- keep.all.logs=true가 rsed.envvars에 지정되어 있으면 ffs*.log, lock.log 및 rsecomm.log 파일이 확장 이름을 갖습니다. 기본적으로 확장 이름이 사용됩니다. 다음은 샘플 확장 이름입니다: ffs.RSEDx#yyyymmddhhmmss.log. 여기서 RSEDx는 스레드 풀의 주소 공간 이름을 나타내고 yyyymmddhhmmss는 날짜 및 시간소인(년, 월, 일, 시, 분, 초)입니다.

SCLM 개발자 툴킷 로깅

- userlog/\$LOGNAME/rsecomm.log

SCLM 개발자 툴킷에 대한 통신 로깅입니다(여기서 userlog는 rsed.envvars의 user.log 및 DSTORE_LOG_DIRECTORY 지시문을 결합한 값이고 \$LOGNAME은 로그인 사용자 ID(대문자)임). user.log 지시문이 주석 처리되어 있거나 없으면 사용자

의 홈 경로가 사용됩니다. 홈 경로는 사용자 ID의 OMVS 보안 세그먼트에 정의됩니다. DSTORE_LOG_DIRECTORY 지시문이 주석 처리되어 있거나 없으면 user.log 값에 .eclipse/RSE/가 추가됩니다.

CARMA 로깅

- **CARMA 서버 작업**

일괄처리 인터페이스를 사용하여 CARMA와의 연결을 열면, FEK.#CUST.SYSPROC (CRASUBMT)가 CRAport 서버 작업(사용자의 사용자 ID를 소유자로 사용)을 시작합니다(여기서 port는 사용되는 TCP/IP 포트임).

- **CARMALOG DD**

CARMALOG DD 문이 선택된 CARMA 시작 메소드에 지정된 경우, CARMA 로깅은 서버 작업의 이 DD 문으로 경로 재지정됩니다. 그렇지 않으면, SYSPRINT로 이동합니다.

- **SYSPRINT DD**

CARMALOG DD 문이 정의되지 않은 경우 서버 작업의 SYSPRINT DD는 CARMA 로깅을 보유합니다.

- **SYSTSPRT DD**

서버 작업의 SYSPRINT DD는 CARMA 서버 시작에 대한 시스템(TSO) 메시지를 보유합니다.

- **userlog/\$LOGNAME/rsecomm.log**

CARMA에 대한 통신 로깅입니다(여기서 userlog는 rsed.envvars의 user.log 및 DSTORE_LOG_DIRECTORY 지시문을 결합한 값이고 \$LOGNAME은 로그인 사용자 ID(대문자)임). user.log 지시문이 주석 처리되어 있거나 없으면 사용자의 홈 경로가 사용됩니다. 홈 경로는 사용자 ID의 OMVS 보안 세그먼트에 정의됩니다. DSTORE_LOG_DIRECTORY 지시문이 주석 처리되어 있거나 없으면 user.log 값에 .eclipse/RSE/가 추가됩니다.

fekfivpc IVP 테스트 로깅

- **/tmp/fekfivpc.log**

fekfivpc 명령(CARMA 관련 IVP 테스트)은 RSE와 CARMA 간의 통신을 문서화하기 위해 fekfivpc.log 파일을 작성합니다. 이 로그는 TMPDIR이 rsed.envvars에 정의된 경우 이 변수가 참조하는 디렉토리에 작성됩니다. 이 변수가 정의되어 있지 않은 경우, 파일은 /tmp에 작성됩니다.

fekfivpi IVP 테스트 로깅

- /tmp/fekfivpi.log

fekfivpi -file 명령(TSO/ISPF Client Gateway 관련 IVP 테스트)의 출력입니다. 이 로그는 TMPDIR이 rsed.envvars에 정의된 경우 이 변수가 참조하는 디렉토리에 작성됩니다. 이 변수가 정의되어 있지 않은 경우, 파일은 /tmp에 작성됩니다.

fekfivps IVP 테스트 로깅

- /tmp/fekfivps.log

fekfivps -file 명령(SCLMDT 관련 IVP 테스트)의 출력입니다. 이 로그는 TMPDIR이 rsed.envvars에 정의된 경우 이 변수가 참조하는 디렉토리에 작성됩니다. 이 변수가 정의되어 있지 않은 경우, 파일은 /tmp에 작성됩니다.

코드 검토 로깅

- SYSTSPRT DD

코드 검토 프로시저를 호출하는 단계의 SYSTSPRT DD는 코드 분석 프로세스를 구동하는 프론트 엔드의 메시지를 보관합니다.

- WORKSPCE DD

코드 검토 프로시저를 호출하는 단계의 WORKSPCE DD는 코드 분석 프로세스의 Eclipse 작업공간 로그 메시지를 보관합니다.

- ERRMSGs DD

코드 검토 프로시저를 호출하는 단계의 ERRMSGs DD는 코드 분석 프로세스의 stderr 출력을 보관합니다.

코드 적용 로깅

- SYSTSPRT DD

코드 검토 프로시저를 호출하는 단계의 SYSTSPRT DD는 코드 분석 프로세스를 구동하는 프론트 엔드의 메시지를 보관합니다.

- WORKSPCE DD

코드 검토 프로시저를 호출하는 단계의 WORKSPCE DD는 코드 분석 프로세스의 Eclipse 작업공간 로그 메시지를 보관합니다.

- ERRMSGs DD

코드 검토 프로시저를 호출하는 단계의 ERRMSGs DD는 코드 분석 프로세스의 stderr 출력을 보관합니다.

덤프 파일

제품이 비정상적으로 종료되면 문제점 판별을 지원하기 위해 스토리지 덤프가 작성됩니다. 이러한 덤프의 가용성과 위치는 사이트별 설정에 따라 상당히 다릅니다. 덤프가 작성되지 않거나 다음 섹션에 언급된 것과 다른 위치에 덤프가 작성될 수 있습니다.

MVS 덤프

프로그램이 MVS에서 실행 중인 경우, 시스템 덤프 파일을 검사하고 JCL에 다음 DD 문(제품에 따라 다름)이 있는지 확인하십시오.

- SYSABEND
- SYSMDUMP
- SYSUDUMP
- CEEDUMP
- SYSPRINT
- SYSOUT

이러한 DD 문에 대한 자세한 정보는 *MVS JCL Reference*(SA22-7597) 및 *Language Environment Debugging Guide*(GA22-7560)를 참조하십시오.

Java 덤프

z/OS UNIX에서 대부분의 Developer for System z 덤프는 JVM(Java Virtual Machine)에 의해 제어됩니다.

JVM은 초기화 중에 기본적으로 덤프 에이전트 세트(SYSTDUMP 및 JAVADUMP)를 작성합니다. JAVA_DUMP_OPTS 환경 변수를 사용하여 이 덤프 에이전트 세트를 대체할 수 있으며, 명령행에서 -Xdump를 사용하여 이 세트를 추가로 대체할 수 있습니다. JVM 명령행 옵션은 rsed.envvars의 _RSE_JAVA0PTS 지시문에 정의됩니다. IBM 지원 센터에서 지시하지 않는 한 덤프 설정을 변경하지 마십시오.

참고: 명령행에서 -Xdump:what 옵션을 사용하여 시작 완료 시 존재하는 덤프 에이전트를 결정할 수 있습니다.

작성할 수 있는 덤프 유형은 다음과 같습니다.

SYSTDUMP

Java 트랜잭션 덤프입니다. z/OS에서 생성한 형식화되지 않은 스토리지 덤프입니다.

이 덤프는 %uid.JVM.TDUMP.%job.D%ym%d.T%H%M%S 양식의 기본 이름을 사용하여 또는 JAVA_DUMP_TDUMP_PATTERN 환경 변수 설정을 통해 결정된 대로 순차 MVS 데이터 세트에 기록됩니다.

참고: JAVA_DUMP_TDUMP_PATTERN에서는 변수(트랜잭션 덤프가 수행될 때 실제 값으로 변환됨)를 사용할 수 있습니다.

표 38. JAVA_DUMP_TDUMP_PATTERN 변수

변수	사용법
%uid	사용자 ID
%job	작업 이름
%y	연도(두 자리 숫자)
%m	월(두 자리 숫자)
%d	일(두 자리 숫자)
%H	시(두 자리 숫자)
%M	분(두 자리 숫자)
%S	초(두 자리 숫자)

CEEDUMP

LE(Language Environment) 덤프입니다. JVM 프로세스에 있는 각 스레드에 대한 스택 추적을 레지스터 정보 및 각 레지스터에 대한 간략한 스토리지 덤프와 함께 표시하는 형식화된 요약 시스템 덤프입니다.

이 덤프는 CEEDUMP.yyyymmdd.hhmmss.pid라는 z/OS UNIX 파일에 기록됩니다(여기서 yyyymmdd는 현재 날짜이고 hhmmss는 현재 시간이며, pid는 현재 프로세스 ID임). 이 파일의 가능한 위치는 199 페이지의 『z/OS UNIX 덤프 위치』에 설명되어 있습니다.

HEAPDUMP

Java 힙에 있는 오브젝트에 대한 형식화된 덤프(목록)입니다.

이 덤프는 HEAPDUMP.yyyymmdd.hhmmss.pid.TXT라는 z/OS UNIX 파일에 기록됩니다(여기서 yyyymmdd는 현재 날짜이고 hhmmss는 현재 시간이며, pid는 현재 프로세스 ID임). 이 파일의 가능한 위치는 199 페이지의 『z/OS UNIX 덤프 위치』에 설명되어 있습니다.

Developer for System z에서 이 덤프를 트리거할 연산자 명령을 제공합니다. 세부사항은 *Host Configuration Guide*(SC23-7658)에서 "연산자 명령" 장을 참조하십시오.

JAVADUMP

JVM에 대한 형식화된 분석입니다. JVM 및 Java 애플리케이션과 관련된 진단 정보(예: 애플리케이션 환경, 스레드, 원시 스택, 잠금, 메모리)를 포함합니다.

이 덤프는 JAVADUMP.yyyymmdd.hhmmss.pid.TXT라는 z/OS UNIX 파일에 기록됩니다(여기서 yyyymmdd는 현재 날짜이고 hhmmss는 현재 시간이며, pid는 현재 프로세스 ID임). 이 파일의 가능한 위치는 199 페이지의 『z/OS UNIX 덤프 위치』에 설명되어 있습니다.

Developer for System z에서 이 덤프를 트리거할 연산자 명령을 제공합니다. 세부사항은 *Host Configuration Guide*(SC23-7658)에서 "연산자 명령" 장을 참조하십시오.

JVM 덤프에 대한 자세한 정보는 *Java Diagnostic Guide*(SC34-6358)를 참조하고 LE 특정 정보는 *Language Environment Debugging Guide*(GA22-7560)를 참조하십시오.

z/OS UNIX 덤프 위치

JVM은 다음 위치 각각의 존재 여부와 쓰기 권한을 확인하고 CEEDUMP, HEAPDUMP, JAVADUMP 파일을 사용 가능한 첫 번째 위치에 저장합니다. 덤프 파일을 올바르게 작성하려면 디스크 여유 공간이 충분해야 합니다.

1. 환경 변수 `_CEE_DMPTARG`의 디렉토리(있는 경우). 이 변수는 `rsed.envvars`에서 `/tmp`로 설정됩니다. 이 변수를 `/dev/null`로 변경하면 덤프 파일이 작성되지 않습니다.
2. 현재 작업 디렉토리(디렉토리가 루트 디렉토리(/)가 아니고 디렉토리가 쓰기 가능한 경우).
3. 환경 변수 `TMPDIR`(`/tmp` 이외의 임시 디렉토리 위치를 나타내는 환경 변수)의 디렉토리(있는 경우)
4. `/tmp` 디렉토리
5. 이전에 언급한 위치에 덤프를 저장할 수 없으면 `stderr`에 덤프가 배치됩니다.

추적

JES 작업 모니터 추적

JES 작업 모니터 추적은 시스템 운영자가 제어합니다(*Host Configuration Guide* (SC23-7658)의 "운영자 명령" 설명 참조).

- `PRM=-TV` 매개변수로 `JMON` 시작 태스크를 시작하면 상세 모드(추적)가 활성화됩니다.
- 추적 수정 및 메시지 수정 연산자 명령을 사용하면 로그 메시지의 원하는 레벨을 선택할 수 있습니다.

RSE 추적

RSE와 관련된 컴포넌트로 작성되는 로그 파일이 여러 개 있습니다. 대부분의 로그 파일은 `userlog/$LOGNAME/`에 있습니다. 여기서 `userlog`는 `rsed.envvars`에서 `user.log`, `DSTORE_LOG_DIRECTORY` 지시문의 결합 값이고 `$LOGNAME`은 로그인한 사용자 ID(대문자)입니다. `user.log` 지시문이 주석 처리되어 있거나 없으면 사용자의 홈 경로가 사용됩니다. 홈 경로는 사용자 ID의 OMVS 보안 세그먼트에 정의됩니다. `DSTORE_LOG_DIRECTORY` 지시문이 주석 처리되어 있거나 없으면 `user.log` 값에 `.eclipse/RSE/`가 추가됩니다.

ffs*.log, lock.log와 rsecomm.log에 기록되는 데이터의 양은 **modify rsecommlog** 연산자 명령으로 또는 `debug_level` in `rsecomm.properties`를 설정하여 제어됩니다. 세부사항은 *Host Configuration Guide* (SC23-7658)의 "운영자 명령"과 *Host Configuration Guide* (SC23-7658)의 "(선택사항) RSE 추적"을 참조하십시오.

.dstore* 로그 파일 작성은 `-DDSTORE_*` Java 시작 옵션으로 제어됩니다(*Host Configuration Guide* (SC23-7658)의 "`_RSE_JAVAOPTS`를 사용하여 추가 Java 시작 매개변수 정의" 설명 참조).

참고:

- .eclipse 디렉토리와 .dstore* 로그 파일은 마침표(.)로 시작되어 숨겨져 있습니다. z/OS UNIX 명령 `ls -lA`를 사용하면 숨겨진 파일과 디렉토리를 나열할 수 있습니다. Developer for System z 클라이언트를 사용하는 경우에는 창 > 참조... > 원격 시스템 > 파일 환경 설정 페이지를 선택하고 "숨겨진 파일 표시"를 사용합니다.
- .dstore* 로그 파일은 ASCII로 작성됩니다. z/OS UNIX 명령 `iconv -f ISO8859-1 -t IBM-1047 .dstore*`를 사용하면 EBCDIC로 표시됩니다(코드 페이지 IBM-1047을 사용하는 경우).
- 모든 *.log 파일과 달리, .dstore* 로그 파일은 클라이언트 재연결 시 자동으로 제거되지 않습니다. 이 파일의 제거는 수동으로 수행합니다.

RSE 디먼과 RSE 스레드 풀별 로그 파일은 `daemon-home`에 있습니다. 여기서 `daemon-home`은 `rsed.envvars`에서 `daemon.log` 지시문의 값입니다. `daemon.log` 지시문이 주석 처리되어 있거나 없으면 RSED 시작 태스크에 지정된 사용자 ID의 홈 디렉토리를 사용합니다. 홈 디렉토리는 사용자 ID의 OMVS 보안 세그먼트에 저장됩니다.

`rsedaemon.log`와 `rseserver.log`에 기록되는 데이터의 양은 **modify rsedaemonlog**, **modify rseserverlog** 연산자 명령으로 또는 `rsecomm.properties`의 `debug_level`을 설정하여 제어됩니다. 세부사항은 *Host Configuration Guide* (SC23-7658)의 "운영자 명령"과 *Host Configuration Guide* (SC23-7658)의 "(선택사항) RSE 추적"을 참조하십시오.

`serverlogs.count`, `stderr.*.log`, `stdout.*.log`는 `rsed.envvars`의 `enable.standard.log` 지시문이 활성 상태이거나 **modify rsestandardlog on** 연산자 명령으로 함수가 동적으로 활성화된 경우에만 작성됩니다.

CARMA 추적

사용자는 클라이언트의 CARMA 연결 특성 탭에 추적 레벨을 설정하여 CARMA가 생성하는 추적 정보의 양을 제어할 수 있습니다. 추적 레벨 선택사항은 다음과 같습니다.

- 로깅 사용 안함
- 오류 로깅
- 경고 로깅
- 정보 로깅
- 디버그 로깅

기본값은 다음과 같습니다.

오류 로깅

로그 파일 위치에 대한 자세한 정보는 190 페이지의 『로그 파일』을 참조하십시오.

오류 피드백 추적

다음 프로시저는 원격 빌드 프로시저의 오류 피드백 문제점을 진단하는 데 필요한 정보 수집을 허용합니다. 이 추적은 성능 저하를 초래하므로 IBM 지원 센터의 지시 하에서만 수행해야 합니다. 이 절에서 hlq에 대한 모든 참조는 Developer for System z 설치 중에 사용되는 상위 레벨 규정자를 나타냅니다. 설치 기본값은 FEK이지만 사용자의 사이트에는 적용되지 않을 수 있습니다.

1. 활성 ELAXFCOC 컴파일 프로시저의 백업 사본을 작성하십시오. 이 프로시저는 hlq.SFEKSAMP 데이터 세트에 기본적으로 제공되지만 *Host Configuration Guide* (SC23-7658)의 "ELAXF* 원격 빌드 프로시저"에 설명된 대로 다른 위치(예: SYS1.PROCLIB)로 복사되었을 수 있습니다.
2. EXIT(ADEXIT(ELAXMGUX)) 컴파일 옵션에 'MAXTRACE' 문자열이 포함되도록 활성 ELAXFCOC 프로시저를 변경하십시오.

```
//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)
```

참고: MAXTRACE 주위에 어포스트로피를 이중으로 사용해야 합니다. 이제 옵션은 EXIT(ADEXIT('MAXTRACE',ELAXMGUX))입니다.

3. 자세한 추적을 원하는 COBOL 프로그램에 대한 원격 구문 검사를 수행하십시오.
4. JES 출력의 SYSOUT 파트는 SIDEFILE1, SIDEFILE2, SIDEFILE3, SIDEFILE4에 대한 데이터 세트 이름을 나열하는 것으로 시작합니다.

```
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 SIDEFIL3 - NAME = 'uid.DT021207.TT110823.M0000174.C00000002'
ABOUT TOO OPEN SIDEFIL4 - NAME = 'uid.DT021207.TT110823.M0000236.C00000003'
SUCCESSFUL OPEN SIDEFIL4 - NAME = 'uid.DT021207.TT110823.M0000236.C00000003'
```

참고: 설정에 따라 SIDEFIL1 및 SIDEFIL2는 DD 문을 가리킵니다(SUCCESSFUL OPEN SIDEFIL1 - NAME = DD:WSEDSF1). 출력의 JESJCL 파트(SYSOUT 파트 앞 에 있음)를 참조하여 실제 데이터 세트 이름을 가져오십시오.

```
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. 이 4개의 데이터 세트를 PC로 복사하십시오(예를 들어, Developer for System z 에 로컬 COBOL 프로젝트를 작성하고 SIDEFIL1->4 데이터 세트를 추가).
6. 전체 JES 작업 로그를 PC로 복사하십시오(예를 들어, Developer for System z에 서 작업 출력을 열고 파일 > 다른 이름으로 저장 ...을 선택하여 로컬 프로젝트에 저장).
7. 변경을 실행 취소(컴파일 옵션에서 "MAXTRACE", 문자열 제거)하거나 백업을 복원하 여 ELAXFCOC 프로시저를 원래 상태로 복원하십시오.
8. 수집된 파일(SIDEFIL1->4 및 작업 로그)을 IBM 지원 센터로 보내십시오.

z/OS UNIX 권한 비트

Developer for System z에서는 z/OS UNIX 파일 시스템과 일부 z/OS UNIX 파일 에 특정 권한 비트가 설정되어야 합니다.

SETUID 파일 시스템 속성

RSE(Remote Systems Explorer)는 호스트에 클라이언트 연결과 같은 코어 서비스를 제공하는 Developer for System z 컴포넌트입니다. 사용자의 보안 환경 작성과 같은 태스크를 수행할 수 있어야 합니다.

SETUID 권한 비트(시스템 기본값임)를 사용하여 Developer for System z가 설치된 파 일 시스템(HFS 또는 zFS)을 마운트해야 합니다. NOSETUID 매개변수를 사용하여 파일 시스템을 마운트하면 Developer for System z가 사용자의 보안 환경을 작성하지 못하 며 연결 요청에 실패합니다. 이 설정 문제점을 나타내는 기타 지표는 다음과 같습니다.

- "FEK999E fekfomvs 모듈이 APF 권한 부여된 것으로 표시되어야 함" 콘솔 메시 지
- PassTicket IVP가 실패하고 "ICH409I 282-010 ABEND DURING RACHECK PROCESSING"이 표시됨

Java 또는 z/OS UNIX 2진을 호스팅하는 파일 시스템이 NOSETUID 매개변수를 사 용하여 마운트되면 BPXP014I 및 BPXP015I 메시지와 같은 유사한 오류를 예상할 수 있습니다.

SETUID 비트의 현재 상태를 나열하려면 TSO ISHELL 명령을 사용하십시오. ISHELL 패널에서 **File_systems > 1. 마운트 테이블...**을 선택하여 마운트된 파일 시스템을 나열하십시오. **a** 행 명령은 선택된 파일 시스템의 속성을 표시합니다(여기서 “Ignore SETUID” 필드는 0이어야 함).

프로그램 제어 권한

RSE(Remote Systems Explorer)는 호스트에 클라이언트 연결과 같은 코어 서비스를 제공하는 Developer for System z 컴포넌트입니다. 클라이언트의 사용자 ID로 전환과 같은 태스크를 수행하려면 프로그램 제어를 실행해야 합니다.

21 페이지의 제 2 장 『보안 고려사항』에 설명된 대로 보안 제품에 대한 Java 인터페이스의 경우를 제외하고 z/OS UNIX 프로그램 제어 비트는 필요한 경우 SMP/E 설치 중에 설정됩니다. Developer for System z 디렉토리 수동 복사 중에 보존하지 않은 경우에는 이 권한 비트가 유실될 수 있습니다.

프로그램 제어해야 하는 Developer for System z 파일은 다음과 같습니다.

- /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

다음 샘플(\$는 z/OS UNIX 프롬프트임)에 표시된 대로 문자 p를 사용하여 프로그램 제어 비트가 표시되는 확장 속성을 나열하려면 z/OS UNIX 명령 **ls -E**를 사용하십시오.

```
$ cd /usr/lpp/rdz
$ ls -E lib/fekf*
-rwxr-xr-x -ps- 2 user      group      94208 Jul  8 12:31 lib/fekfdir.dll
```

다음 샘플(\$ 및 #은 z/OS UNIX 프롬프트임)에 표시된 대로 프로그램 제어 비트를 수동으로 설정하려면 z/OS UNIX 명령 **extattr +p**를 사용하십시오.

```
$ 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
```

참고: **extattr +p** 명령을 사용할 수 있으려면, 최소한 보안 소프트웨어의 FACILITY 클래스에 있는 BPX.FILEATTR.PROGCTL 프로파일에 대한 READ 액세스 권한이 있거나 이 프로파일이 정의되지 않은 경우 슈퍼유저(UID 0)여야 합니다. 자세한 정보는 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.

APF 권한 부여

RSE(Remote Systems Explorer)는 호스트에 클라이언트 연결과 같은 코어 서비스를 제공하는 Developer for System z 컴포넌트입니다. 자세한 프로세스 자원 사용량 표시와 같은 태스크를 수행하려면 APF 권한 부여를 실행해야 합니다.

z/OS UNIX APF 비트는 필요한 경우 SMP/E 설치 중에 설정됩니다. Developer for System z 디렉토리 수동 복사 중에 보존하지 않은 경우에는 이 권한 비트가 유실될 수 있습니다.

다음 Developer for System z 파일은 APF 권한 부여해야 합니다.

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

다음 샘플(\$는 z/OS UNIX 프롬프트임)에 표시된 대로 문자 a를 사용하여 APF 비트가 표시되는 확장 속성을 나열하려면 z/OS UNIX 명령 **ls -E**를 사용하십시오.

```
$ cd /usr/lpp/rdz
$ ls -E bin/fekfriwp
-rwxr-xr-x aps- 2 user      group      114688 Sep 17 06:41 bin/fekfriwp
```

다음 샘플(\$ 및 #은 z/OS UNIX 프롬프트임)에 표시된 대로 APF 비트를 수동으로 설정하려면 z/OS UNIX 명령 **extattr +a**를 사용하십시오.

```
$ 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
```

참고: **extattr +a** 명령을 사용할 수 있으려면, 최소한 보안 소프트웨어의 FACILITY 클래스에 있는 BPX.FILEATTR.APF 프로파일에 대한 READ 액세스 권한이 있거나 이 프로파일이 정의되지 않은 경우 슈퍼유저(UID 0)여야 합니다. 자세한 정보는 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.

스티키(Sticky) 비트

일부 선택적 Developer for System z 서비스에서는 z/OS UNIX에 MVS 로드 모듈을 사용할 수 있어야 합니다. z/OS UNIX에 "스티키(Sticky)" 비트가 있는 스텝(더미 파일)을 작성하여 이를 수행합니다. 스텝이 실행될 때 z/OS UNIX는 동일한 이름을 가진 MVS 로드 모듈을 검색하고 로드 모듈을 대신 실행합니다.

z/OS UNIX 스티키(Sticky) 비트는 필요한 경우 SMP/E 설치 중에 설정됩니다. Developer for System z 디렉토리 수동 복사 중에 보존하지 않은 경우에는 이 권한 비트가 유실될 수 있습니다.

스티키(Sticky) 비트가 있어야 하는 Developer for System z 파일은 다음과 같습니다.

- /usr/lpp/rdz/bin/
 - AZUTSTRN
 - BWBTSOW
 - BWBTRANT
 - CRASTART

다음 샘플(\$는 z/OS UNIX 프롬프트임)에 표시된 대로 문자 t를 사용하여 스티키(Sticky) 비트가 표시되는 권한을 나열하려면 z/OS UNIX 명령 **ls -l**를 사용하십시오.

```
$ cd /usr/lpp/rdz
$ ls -l bin/CRA*
-rwxr-xr-t  2 user      group      71 Jul  8 12:31 bin/CRASTART
```

다음 샘플(\$ 및 #은 z/OS UNIX 프롬프트임)에 표시된 대로 스티키(Sticky) 비트를 수동으로 설정하려면 z/OS UNIX 명령 **chmod +t**를 사용하십시오.

```
$ 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
```

참고: **chmod** 명령을 사용할 수 있으려면, 최소한 보안 소프트웨어의 UNIXPRIV 클래스에 있는 SUPERUSER.FILESYS.CHANGEPERMS 프로파일에 대한 READ 액세스 권한이 있거나 이 프로파일이 정의되지 않은 경우 슈퍼유저(UID 0)여야 합니다. 자세한 정보는 *UNIX System Services Planning*(GA22-7800)을 참조하십시오.

예약된 TCP/IP 포트

netstat 명령(TSO 또는 z/OS UNIX)을 사용하면 현재 사용 중인 포트에 대한 개요 정보를 얻을 수 있습니다. 이 명령의 출력은 다음 예와 같습니다. 사용되는 포트는 "Local Socket" 열에서 마지막 숫자(".." 뒤)입니다. 이러한 포트는 이미 사용 중이므로 Developer for System z 구성에 사용할 수 없습니다.

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		

존재할 수 있는 다른 제한사항은 예약된 TCP/IP 포트입니다. TCP/IP 포트는 다음과 같은 2가지 공통 위치에 예약할 수 있습니다.

- **PROFILE.TCPIP**

TCP/IP 시작 태스크의 PROFILE DD 문이 참조하는 데이터 세트입니다 (SYS1.TCPPARMS(TCPPROF)).

- PORT: 지정된 작업 이름에 포트를 예약합니다.

- PORTRANGE: 지정된 작업 이름에 포트 범위를 예약합니다.

이러한 명령문에 대한 자세한 정보는 *Communications Server: IP Configuration Guide*(SC31-8775)를 참조하십시오.

• **SYS1.PARMLIB(BPXPRMxx)**

- INADDRANYPORT: 시스템이 PORT 0, INADDR_ANY 바인드에 사용하기 위해 예약하는 포트 번호 범위에 대한 시작 포트 번호를 지정합니다. 이 값은 CINET(단일 호스트에서 활성화된 여러 TCP/IP 스택)에만 필요합니다.
- INADDRANYCOUNT: 시스템이 예약하는 포트의 수를 지정합니다 (INADDRANYPORT 매개변수에 지정된 포트 번호로 시작). 이 값은 CINET(단일 호스트에서 활성화된 여러 TCP/IP 스택)에만 필요합니다.

이러한 명령문에 대한 자세한 정보는 *UNIX System Services Planning* (GA22-7800)과 *MVS Initialization and Tuning Reference*(SA22-7592)를 참조하십시오(아래 설명 참조).

이러한 예약 포트는 다음과 같은 예의 출력을 작성하는 **netstat portl** 명령(TSO 또는 z/OS UNIX)으로 나열할 수 있습니다.

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		

NETSTAT 명령에 대한 자세한 정보는 *Communications Server: IP System Administrator's Command*(SC31-8781)를 참조하십시오.

참고: **NETSTAT** 명령은 BPXPRMxx 정의를 대체해야 하는 PROFILE.TCPIP에 정의된 정보만 표시합니다. 잘 모르거나 문제가 있는 경우에는 BPXPRMxx parmlib 멤버를 확인하여 해당 예약 포트를 확인하십시오.

주소 공간 크기

z/OS UNIX Java 프로세스인 RSE 디먼이 기능을 수행하려면 리전 크기가 커야 합니다. 따라서 OMVS 주소 공간에 대해 스토리지 한계를 높게 설정하는 것이 중요합니다.

시작 JCL 요구사항

RSE 디먼은 BPXBATSL(리전 크기는 0이어야 함)을 사용하여 JCL에 의해 시작됩니다.

SYS1.PARMLIB(BPXPRMxx)에 설정된 제한사항

기본 OMVS 주소 공간(프로세스) 리전 크기를 정의하는 SYS1.PARMLIB(BPXPRMxx)의 MAXASSIZE를 2G로 설정하십시오. 이는 허용된 최대 크기입니다. 이는 시스템 전체 한계이므로 모든 z/OS UNIX 주소 공간에 대해 활성화됩니다. 이를 원하지 않을 경우, 보안 소프트웨어에 Developer for System z에 대해서만 한계를 설정할 수도 있습니다.

MVS System Commands(GC28-1781)에 설명된 대로 다음 콘솔 명령을 사용하여 (다음 IPL까지) 이 값을 동적으로 확인하고 설정할 수 있습니다.

1. DISPLAY OMVS,0
2. SETOMVS MAXASSIZE=2G

보안 프로파일에 저장된 제한사항

SYS1.PARMLIB(BPXPRMxx) 값을 사용하려면 디먼의 사용자 ID OMVS 세그먼트에서 ASSIZEMAX를 확인하고 2147483647 또는 가급적 NONE으로 설정하십시오.

RACF를 사용하는 경우, *Security Server RACF Command Language Reference* (SA22-7687)에 설명된 대로 다음 TSO 명령을 사용하여 이 값을 확인하고 설정할 수 있습니다.

1. LISTUSER userid NORACF OMVS
2. ALTUSER userid OMVS(NOASSIZEMAX)

시스템 종료에 의해 강제 실행된 제한사항

IEFUSI 또는 IEALIMIT 시스템 종료가 OMVS 주소 공간 리전 크기를 제어할 수 있게 하지 마십시오. 이를 수행하는 가능한 방법은 SYS1.PARMLIB(SMFPRMxx)의 SUBSYS(OMVS,NOEXITS)를 코딩하는 것입니다.

MVS System Commands(GC28-1781)에 설명된 대로 다음 콘솔 명령을 사용하여 SYS1.PARMLIB(SMFPRMxx) 값을 확인하고 활성화할 수 있습니다.

1. DISPLAY SMF,0
2. SET SMF=xx

64비트 주소 지정에 대한 제한사항

SYS1.PARMLIB(SMFPRMxx)의 키워드 MEMLIMIT은 64비트 태스크가 2GB 막대 위로 할당할 수 있는 가상 스토리지의 크기를 제한합니다. MEMLIMIT=0M은 JCL의 REGION 매개변수와 달리 프로세스가 막대 이상의 가상 스토리지를 사용할 수 없음을 의미합니다.

SMFPRMxx에서 MEMLIMIT이 지정되지 않는 경우 기본값은 0M이므로 태스크가 막대 아래 (31비트) 2GB로 바인드됩니다. z/OS 1.10에서는 기본값이 2G로 변경되어 64비트 태스크가 최대 4GB(막대 아래 2GB와 MEMLIMIT이 부여하는 막대 위 2GB)를 사용할 수 있습니다.

MVS System Commands(GC28-1781)에 설명된 대로 다음 콘솔 명령을 사용하여 SYS1.PARMLIB(SMFPRMxx) 값을 확인하고 활성화할 수 있습니다.

1. DISPLAY SMF,0
2. SET SMF=xx

MEMLIMIT은 또한 JCL의 EXEC 카드에서 매개변수로 지정할 수 있습니다. MEMLIMIT 매개변수를 지정하지 않는 경우 기본값은 SMF에 정의된 값입니다(기본값이 NOLIMIT 인 REGION=0M이 지정되는 경우 제외).

기타 정보

오류 피드백 B37 공간 이상 종료

사용자가 컴파일 조치 중에 오류 피드백을 선택하면 Developer for System z가 몇 개의 임시 데이터 세트를 작성합니다. 이러한 데이터 세트 중 하나의 공간이 부족하면 컴파일 작업이 종료되고 B37-04 공간 이상 종료로 표시됩니다.

이 문제점을 경험하게 되면 FEK.SFEKPROC(FEKFERRF)에서 공간 할당을 조정하십시오. 기본값은 SPACE(200,40) TRACKS입니다.

시스템 한계

SYS1.PARMLIB(BPXPRMxx)는 다수의 z/OS UNIX 관련 제한사항을 정의하는데, 몇 개의 Developer for System z 클라이언트가 활성화되어 있을 때 이에 도달할 수 있습니다. SETOMVS 및 SET OMVS 콘솔 명령을 사용하여 대부분의 BPXPRMxx 값을 동적으로 변경할 수 있습니다.

BPXPRMxx 한계에 도달하려고 할 때 z/OS UNIX가 콘솔 메시지(BPXI040I)를 표시하게 하려면 SETOMVS LIMMSG=ALL 콘솔 명령을 사용하십시오.

연결이 거부됨

각 RSE 연결은 영구적으로 활성화된 몇 개의 프로세스를 시작합니다. SYS1.PARMLIB(BPXPRMxx)에 설정된 프로세스 수에 대한 한계로 인해, 특히 사용자가 동일한 UID를 공유하는 경우(예: 기본 OMVS 세그먼트를 사용하는 경우) 새 연결을 거부할 수 있습니다.

- UID당 한계는 MAXPROCUSER 키워드를 사용하여 설정되며 기본값은 25입니다.
- 시스템 전체 한계는 MAXPROCSYS 키워드를 사용하여 설정되며 기본값은 200입니다.

다른 거부된 연결 소스는 활성 z/OS 주소 공간 양과 z/OS UNIX 사용자 수에 대한 한계입니다.

- 최대 주소 공간 ID(ASID) 수는 MAXUSER 키워드를 사용하여 SYS1.PARMLIB (IEASYSxx)에 정의되며 기본값은 255입니다.
- 최대 z/OS UNIX 사용자 ID(UID) 수는 MAXUIDS 키워드를 사용하여 SYS1.PARMLIB (BPXPRMxx)에 정의되며 기본값은 200입니다.

OutOfMemoryError

RSE 스레드 풀이 실패하고 OutOfMemoryError 메시지가 로그됩니다. 이 오류는 Java 힙 크기와 관련이 있으며 이 스레드 풀의 활성 사용자가 예상보다 많은 자원을 사용하는 경우 발생할 수 있습니다. 이 오류의 일반적인 원인은 다음과 같습니다.

- 원격 시스템 탐색기에서 대형 데이터 세트 필터 확장
- 멤버가 많은 PDS(E) 열기
- 대규모 멤버 또는 순차 파일 열기

이 문제를 해결할 수 있는 방법은 다음과 같습니다.

- 최대 Java 힙 크기를 제어하는 rsed.envvars의 -Xmx 지시문을 늘립니다. Java 힙은 주소 공간 한계에 포함되어야 합니다.
- 단일 스레드 풀에 배치되어 단일 Java 힙을 공유할 수 있는 사용자 수를 제어하는 rsed.envvars의 -Dmaximum.clients 지시문을 줄입니다.

호스트 연결 에뮬레이터

- 호스트 연결 에뮬레이터는 RSE 서버가 아닌 TN3270 Telnet을 사용하여 호스트에 연결합니다.
- 보안 Telnet(SSL)을 사용하고 잘 알려진 CA에서 서명하지 않은 인증서에 대해 작업하는 경우, 모든 클라이언트는 신뢰할 수 있는 CA의 호스트 연결 에뮬레이터 목록에 CA 인증서를 추가해야 합니다.
- SNA 기능 확장을 사용하지 않으려면 TCP/IP TELNETPARMS의 NOSNAEXT 옵션이 필요할 수도 있습니다. NOSNAEXT가 지정된 경우, TN3270 Telnet 서버는 경합 해결 및 SNA 감지 기능을 위해 협상하지 않습니다.

제 13 장 SSL 및 X.509 인증 설정

이 부록은 SSL(Secure Socket Layer)을 설정할 때 또는 기존 설정을 확인하거나 수정할 때 발생할 수 있는 일반적인 문제점 해결에 유용한 정보를 제공합니다. 이 부록은 사용자가 X.509 인증서를 사용하여 스스로를 인증하는 것을 지원할 수 있도록 샘플 설정도 제공합니다.

보안 통신은 통신 파트너의 적격성을 확인하고 다른 사용자가 데이터를 쉽게 가로채고 읽을 수 없는 방법으로 정보를 전달하는 것을 의미합니다. SSL은 이 기능을 TCP/IP 네트워크에서 제공합니다. SSL은 디지털 인증서를 사용하여 사용자를 식별하고 공개 키 프로토콜을 사용하여 통신을 암호화합니다. 디지털 인증서와 SSL이 사용하는 공개 키 프로토콜에 대한 자세한 정보는 *Security Server RACF Security Administrator's Guide*(SA22-7683)를 참조하십시오.

Developer for System z에 SSL 통신을 설정하는 데 필요한 조치는 정확한 요구, 사용된 RSE 통신 방법, 사이트에서 이미 제공되는 정보에 따라 사이트별로 다릅니다.

이 부록에서는 현재 RSE 정의를 복제하므로 SSL을 사용하는 두 번째 RSE 디먼 연결을 갖게 됩니다. 또한 RSE 연결의 다른 파트가 사용할 자체 보안 인증서를 작성합니다.

- 212 페이지의 『SSL 또는 TLS를 암호화 방법으로 사용하도록 결정』
- 212 페이지의 『개인 키 및 인증서 저장 위치 결정』
- 213 페이지의 『RACF를 사용하여 키 링 작성』
- 215 페이지의 『기존 RSE 설정 복제』
- 216 페이지의 『rsed.envvars를 업데이트하여 공존 사용』
- 216 페이지의 『ssl.properties를 업데이트하여 SSL 사용』
- 217 페이지의 『새 RSE 디먼을 작성하여 SSL 활성화』
- 218 페이지의 『연결 테스트』
- 221 페이지의 『(선택사항) X.509 클라이언트 인증 지원 추가』
- 222 페이지의 『(선택사항) gskkyman을 사용하여 키 데이터베이스 작성』
- 225 페이지의 『(선택사항) keytool을 사용하여 키 저장소 작성』

이 부록에서는 다음과 같은 단일 이름 지정 규칙이 사용됩니다.

- 인증서: rdzrse
- 키 및 인증서 스토리지: rdzssl.*
- 비밀번호: rsessl

- 디먼 사용자 ID : stcrse

다음 절에서 설명하는 일부 태스크를 수행하려면 사용자가 z/OS UNIX에서 활성 상태여야 합니다. 이 작업은 TSO 명령 **OMVS**를 실행하여 수행할 수 있습니다. **exit** 명령을 사용하면 TSO로 리턴됩니다.

SSL 또는 TLS를 암호화 방법으로 사용하도록 결정

rsed.envvars의 `_RSE_JAVAOPTS` 지시문에서 `DSTORE_SSL_ALGORITHM` 변수를 사용하면 SSL과 해당 계층자 TLS(Transport Layer Security) 중에 암호화 방법을 선택할 수 있습니다. 이 내용은 *호스트 구성 안내서(SC23-7658)*의 "`_RSE_JAVAOPTS`로 추가 Java 시작 매개변수 정의"에 설명되어 있습니다.

개인 키 및 인증서 저장 위치 결정

SSL에서 사용되는 ID 인증서와 암호화/암호 해독 키는 키 파일에 저장됩니다. 애플리케이션 유형에 따라 이 키 파일 구현이 다릅니다.

그러나 모든 구현은 샘플 원칙을 따릅니다. 명령이 키 쌍(공개 키 및 연관된 개인 키)을 생성합니다. 그런 다음 명령은 단일 요소 인증서 체인으로 저장된 X.509 자체 서명 인증서에 공개 키를 래핑합니다. 이 인증서 체인과 개인 키는 키 파일에 키 파일에 (별명으로 식별된) 항목으로 저장됩니다.

RSE 디먼은 시스템 SSL 애플리케이션이며 키 데이터베이스 파일을 사용합니다. 이 키 데이터베이스는 gskkyman이 작성한 실제 파일 또는 SAF 준수 보안 소프트웨어(예: RACF)가 관리하는 키 링입니다. 디먼에 의해 시작되는 RSE 서버는 Java SSL 애플리케이션이며 keytool이 작성한 키 저장소 파일 또는 보안 소프트웨어가 관리하는 키 링입니다.

표 39. SSL 인증서 스토리지 메커니즘

인증서 스토리지	작성자 및 관리자	RSE 디먼	RSE 서버
키 링	SAF 준수 보안 제품	지원됨	지원됨
키 데이터베이스	z/OS UNIX의 gskkyman	지원됨	/
키 저장소	Java의 keytool	/	지원됨

SSL을 통해 연결하려면 z/OS UNIX 파일 또는 SAF 준수 키 링으로 키 저장소와 키 데이터베이스가 둘 다 필요합니다.

- 키 저장소(RACF 또는 keytool)
- 키 데이터베이스(RACF 또는 gskkyman)

참고:

- SAF 준수 키 링은 인증서 관리를 위해 선호하는 방법입니다.

- RSE 디먼과 RSE 서버가 동일한 인증서 관리 방법을 사용하는 경우에는 공유 인증서를 사용할 수 있습니다.
- RSE 디먼은 프로그램 제어를 실행해야 합니다. 시스템 SSL 사용은 보안 소프트웨어가 SYS1.SIEALNKE를 프로그램 제어해야 함을 의미합니다.
- 시스템 SSL 애플리케이션(디먼 연결)을 실행하려면 SYS1.SIEALNKE가 LINKLIST 또는 STEPLIB에 있어야 합니다. STEPLIB 방법을 선호할 경우, rsed.envvars 끝에 다음 명령문을 추가하십시오.

```
STEPLIB=$STEPLIB:SYS1.SIEALNKE
```

그러나 다음을 알아 두십시오.

- z/OS UNIX에서 STEPLIB를 사용하면 성능이 저하됩니다.
- 하나의 STEPLIB 라이브러리에 APF 권한이 부여된 경우, 모두에게 권한을 부여해야 합니다. 라이브러리는 STEPLIB의 권한이 부여되지 않은 라이브러리와 혼합되면 APF 권한을 유실합니다.
- 시스템 SSL은 사용 가능한 경우 ICSF(Integrated Cryptographic Service Facility)를 사용합니다. ICSF는 시스템 SSL 소프트웨어 알고리즘 대신 사용될 하드웨어 암호화 지원을 제공합니다. 자세한 정보는 *System SSL Programming(SC24-5901)*을 참조하십시오.

RACF 및 디지털 인증서에 대한 자세한 정보는 *Security Server RACF Security Administrator's Guide(SA22-7683)*를 참조하십시오. gskkyman 문서는 *System SSL Programming(SC24-5901)*에 있으며 keytool 문서는 <http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html>에 있습니다.

RACF를 사용하여 키 링 작성

gskkyman을 사용하여 RSE 디먼 키 데이터베이스를 작성하고 keytool을 사용하여 RSE 서버 키 저장소를 작성하는 경우 이 단계를 실행하지 마십시오.

RACDCERT 명령은 RACF에 개인 키와 인증서를 설치하고 유지보수합니다. RACF는 여러 개의 개인 키와 인증서를 그룹으로 관리하도록 지원합니다. 이 그룹을 키 링이라고 합니다.

RACDCERT 명령에 대한 자세한 내용은 *Security Server RACF Command Language Reference(SA22-7687)*를 참조하십시오.

```

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))

```

이전 샘플은 필수 프로파일을 작성하고 해당 사용자 ID가 소유한 키 링과 인증서에 대한 사용자 ID STCRSE 액세스를 허용하는 것으로 시작합니다. 사용된 사용자 ID는 SSL RSE 디먼을 실행하는 데 사용된 사용자 ID와 일치해야 합니다. 다음 단계는 rdzrse 레이블을 사용하여 새로운 자체 서명 인증서를 작성하는 것입니다. 비밀번호는 필요하지 않습니다. 그런 다음 이 인증서는 새로 작성된 키 링(rdzssl.racf)에 추가됩니다. 인증서와 마찬가지로 키 링의 경우에도 비밀번호가 필요하지 않습니다.

다음 list 옵션을 사용하여 결과를 확인할 수 있습니다.

```

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.0=IBM.L=Raleigh.SP=NC.C=US<
Subject's Name:
    >CN=rdz rse ssl.OU=rdz.0=IBM.L=Raleigh.SP=NC.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
    Ring Owner: STCRSE
    Ring:
        >rdzssl.racf<

```

(선택사항) 서명 인증서 사용

인증서는 자체 서명하거나 인증 기관(CA)에서 서명할 수 있습니다. CA에서 서명한 인증서는 인증서 소유자가 권리자임을 CA에서 보증함을 의미합니다. 서명 프로세스는 인증서에 CA 신임 정보(역시 인증서임)를 추가하여 다중 요소 인증서 체인을 만듭니다.

CA에서 서명한 인증서를 사용하는 경우, Developer for System z 클라이언트가 CA를 이미 신뢰하고 있으면 클라이언트의 신뢰 유효성 검증 질문을 피할 수 있습니다.

CA 서명 인증서를 작성하여 사용하려면 다음 단계를 따르십시오.

1. 자체 서명 인증서를 작성하십시오.

```
RACDCERT ID(stcrse) GENCERT WITHLABEL('rdzrse') . . .
```

2. 이 인증서에 대한 서명 요청을 작성하십시오.

```
RACDCERT ID(stcrse) GENREQ (LABEL('rdzrse')) DSN(dsn)
```


3. 선택한 CA로 서명 요청을 보내십시오.

4. CA 신임 정보(역시 인증서임)가 이미 알려져 있는지 여부를 확인하십시오.

```
RACDCERT CERTAUTH LIST
```

5. CA 인증서를 신뢰할 수 있음으로 표시하십시오.

```
RACDCERT CERTAUTH ALTER(LABEL('CA cert')) TRUST
```

또는 CA 인증서를 데이터베이스에 추가하십시오.

```
RACDCERT CERTAUTH ADD(dsn) WITHLABEL('CA cert') TRUST
```

6. 서명 인증서를 데이터베이스에 추가하십시오. 이는 자체 서명 인증서를 대체합니다.

```
RACDCERT ID(stcrse) ADD(dsn) WITHLABEL('rdzrse') TRUST
```

참고: 대체 전에 자체 서명 인증서를 삭제하지 마십시오. 삭제하면 인증서와 함께 제공되는 개인 키를 유실하여 인증서가 무용지물이 됩니다.

7. 키 링을 작성하십시오.

```
RACDCERT ID(stcrse) ADDRING(rdzssl.racf)
```

8. 서명 인증서를 키 링에 추가하십시오.

```
RACDCERT ID(stcrse) CONNECT(ID(stcrse) LABEL('rdzrse')  
RING(rdzssl.racf))
```

9. CA 인증서를 키 링에 추가하십시오.

```
RACDCERT ID(stcrse) CONNECT(CERTAUTH LABEL('CA cert')  
RING(rdzssl.racf))
```

인증서에 서명하는 데 사용되는 CA 인증서에도 다른 상위 레벨 CA 인증서가 다시 서명할 수 있음에 유의하십시오. 이 경우, 상위 레벨 CA 인증서도 키 링에 추가해야 합니다. 이 프로세스는 상위 레벨 CA 인증서가 루트 CA 인증서(항상 자체 서명 인증서임)가 될 때까지 반복됩니다.

기존 RSE 설정 복제

이 단계에서는 RSE 구성 파일의 새 인스턴스가 작성되므로 SSL 설정을 기존 설정과 병렬로 실행할 수 있습니다. 다음 샘플 명령은 구성 파일이 *Host Configuration Guide* (SC23-7658)의 "사용자 정의 설정"에서 사용되는 기본 위치인 `/etc/rdz`에 있다고 예상합니다.

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

이전 예제에 나열된 z/OS UNIX 명령은 ssl 서브디렉토리를 작성하고 변경이 필요한 구성 파일로 채웁니다. SSL에 특정하지 않기 때문에 기타 구성 파일, 설치 디렉토리, MVS 컴포넌트를 공유할 수 있습니다.

기존 구성 파일의 대부분을 재사용하여 SSL 설정에 실제로 필요한 변경사항에 초점을 맞추고 전체 RSE 설정 수행을 방지할 수 있습니다(예를 들어, ISPF.conf의 새 위치를 정의하지 않을 수 있음).

rsed.envvars를 업데이트하여 공존 사용

지금까지 정의는 현재 설정의 정밀한 복사본이며, 이는 새 RSE 디먼의 로그가 현재 서버 로그 파일을 오버레이함을 의미합니다. RSE는 ssl 디렉토리에 복사되지 않은 구성 파일의 위치도 알아야 합니다. rsed.envvars를 조금만 변경하여 두 문제를 모두 해결할 수 있습니다.

```
$ oedit /etc/rdz/ssl/rsed.envvars
-> change: _RSE_RSED_PORT=4047
-> change: -Ddaemon.Log=/var/rdz/logs/ssl
-> change: -Duser.log=/var/rdz/logs/ssl
-> add at the END:
# -- NEEDED TO FIND THE REMAINING CONFIGURATION FILES
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

이전 예제의 변경사항은 새 로그 위치(로그 위치가 없으면 RSE 디먼이 작성)를 정의합니다. 변경사항은 CLASSPATH도 업데이트하므로 SSL RSE 프로세스는 먼저 현재 디렉토리(/etc/rdz/ssl)에서 구성 파일을 검색한 후 원래 디렉토리(/etc/rdz)를 검색합니다.

ssl.properties를 업데이트하여 SSL 사용

ssl.properties를 업데이트하여 SSL 암호화된 통신 사용을 시작하도록 RSE에 지시합니다.

```
$ oedit /etc/rdz/ssl/ssl.properties
-> change: enable_ssl=true
-> uncomment and change: daemon_keydb_file=rdzssl.racf
-> uncomment and change: daemon_key_label=rdzrse
-> uncomment and change: server_keystore_file=rdzssl.racf
-> uncomment and change: server_keystore_label=rdzrse
-> uncomment and change: server_keystore_type=JCERACFKS
```

이전 예제의 변경사항은 SSL을 사용할 수 있게 하고 rdzssl.racf 키 링의 rdzrse 레이블 아래에 (공유) 인증서가 저장되어 있음을 RSE 디먼과 RSE 서버에 알립니다. JCERACFKS 키워드는 SAF 준수 키 링이 키 저장소로 사용됨을 RSE 서버에 알립니다.

디먼이 사용하는 시스템 SSL은 사용 가능한 경우 System z 암호 하드웨어에 대한 인터페이스인 ICSF를 항상 사용합니다. ICSF 사용 시 디먼 정의를 서버와 공유하려면, server_keystore_type JCECCARACFKS를 지정해야 합니다. 여기서 SAF 준수 키 링은 공개 키에 대한 키 저장소로도 사용되지만 개인 키는 ICSF에 저장됩니다. *Cryptographic Services ICSF Administrator's Guide*(SA22-7521)에 설명된 대로 ICSF는 CSFKEYS 및 CSFSERV 보안 클래스의 프로파일을 사용하여 암호화 키 및 서비스를 사용할 수 있는 사용자를 제어합니다.

새 RSE 디먼을 작성하여 SSL 활성화

앞서 설명한 바와 같이 SSL을 사용할 두 번째 연결을 작성하며 이는 새 RSE 디먼 작성을 의미합니다. RSE 디먼은 시작된 태스크 또는 사용자 작업입니다. 초기(테스트) 설정에 사용자 작업 방법을 사용합니다. 다음 지시사항은 샘플 JCL이 FEK.#CUST.PROCLIB(RSED)(*Host Configuration Guide* (SC23-7658)의 "사용자 정의 설정"에서 사용되는 기본 위치)에 있다고 예상합니다.

1. 새 멤버 FEK.#CUST.PROCLIB(RSEDSSL)을 작성하고 샘플 JCL FEK.#CUST.PROCLIB(RSED)에 복사하십시오.
2. 맨 위에 작업 카드를, 맨 아래에 실행문을 추가하여 RSEDSSL을 사용자 정의하십시오. 다음 코드 샘플에 표시된 대로 SSL 관련 구성 파일(/etc/rdz/ssl)의 위치도 제공하십시오. 사용자 ID STCRSE에는 이전 단계에서 인증서와 키 링에 대한 적절한 액세스 권한이 부여되었기 때문에 이 사용자 ID를 강제로 사용합니다.

```
//RSEDSSL JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),USER=STCRSE
//*
//* RSE DAEMON - SSL
//*
//RSED      PROC TMPDIR=,
//          PORT=,
//          IVP=,                * 'IVP' to do an 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
//*
```

그림 30. RSEDSSL - SSL에 대한 RSE 디먼 사용자 작업

참고: RSEDSSL 작업에 지정된 사용자 ID에는 원래 RSE 디먼과 동일한 권한이 있어야 합니다. FACILITY 프로파일 BPX.SERVER 및 PTKTDATA 프로파일 IRRPTAUTH.FEKAPPL.*은 여기서 주요 요소입니다.

연결 테스트

SSL 호스트 구성이 완료되면 이전에 작성된 FEK.#CUST.PROCLIB(RSEDSSL) 작업을 제출하여 SSL의 RSE 디먼을 시작할 수 있습니다.

이제 Developer for System z 클라이언트와 연결하여 새 설정을 테스트할 수 있습니다. 기존 구성을 복제하여 SSL이 사용할 새 구성을 작성했으므로 RSE 디먼에 포트 4047을 사용하여 클라이언트에 새 연결을 정의해야 합니다.

연결 시 보안 경로를 설정하기 위해 호스트와 클라이언트가 핸드셰이킹을 시작합니다. 이 핸드셰이킹에는 인증서 교환이 포함됩니다. Developer for System z 클라이언트가 호스트 인증서 또는 인증서를 서명한 CA를 인식하지 못하면 Developer for System z 클라이언트가 사용자에게 해당 인증서를 신뢰할 수 있는지 여부를 묻는 프롬프트를 표시합니다.

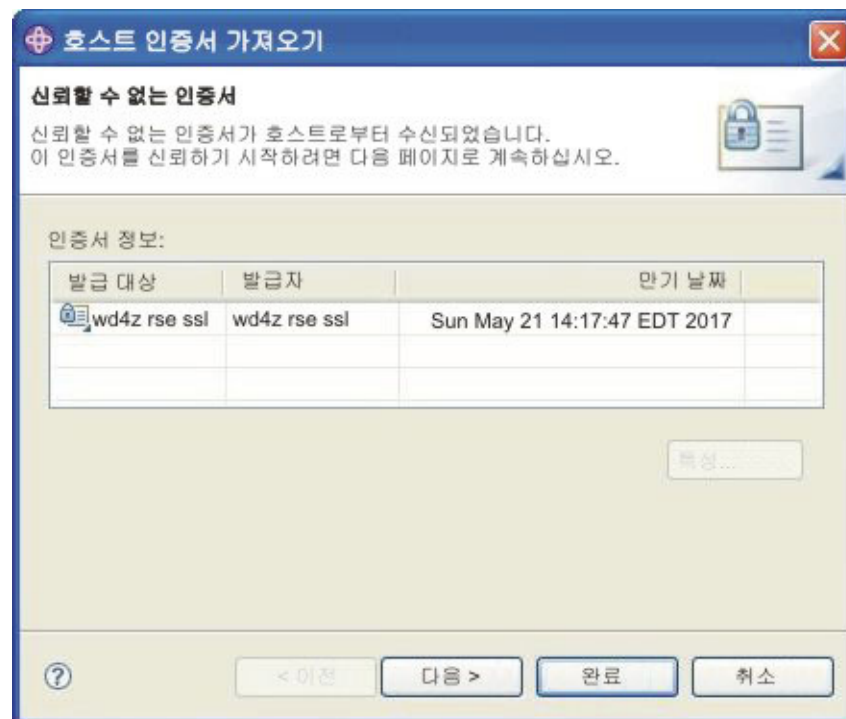


그림 31. 호스트 인증서 가져오기 대화 상자

완료 단추를 클릭하면 사용자가 이 인증서를 신뢰하는 데 동의할 수 있으며 이후 연결 초기화가 계속 진행됩니다.

참고: RSE 디먼과 RSE 서버는 서로 다른 2가지 인증서 위치를 사용하므로 인증서와 확인 모두 서로 다른 2가지가 존재합니다.

클라이언트가 인증서를 인식하면 이 대화 상자가 다시 표시되지 않습니다. 신뢰 인증서 목록은 창 > 환경 설정... > 원격 시스템 > **SSL**을 선택하여 관리할 수 있습니다. 다음과 같은 대화 상자가 표시됩니다.

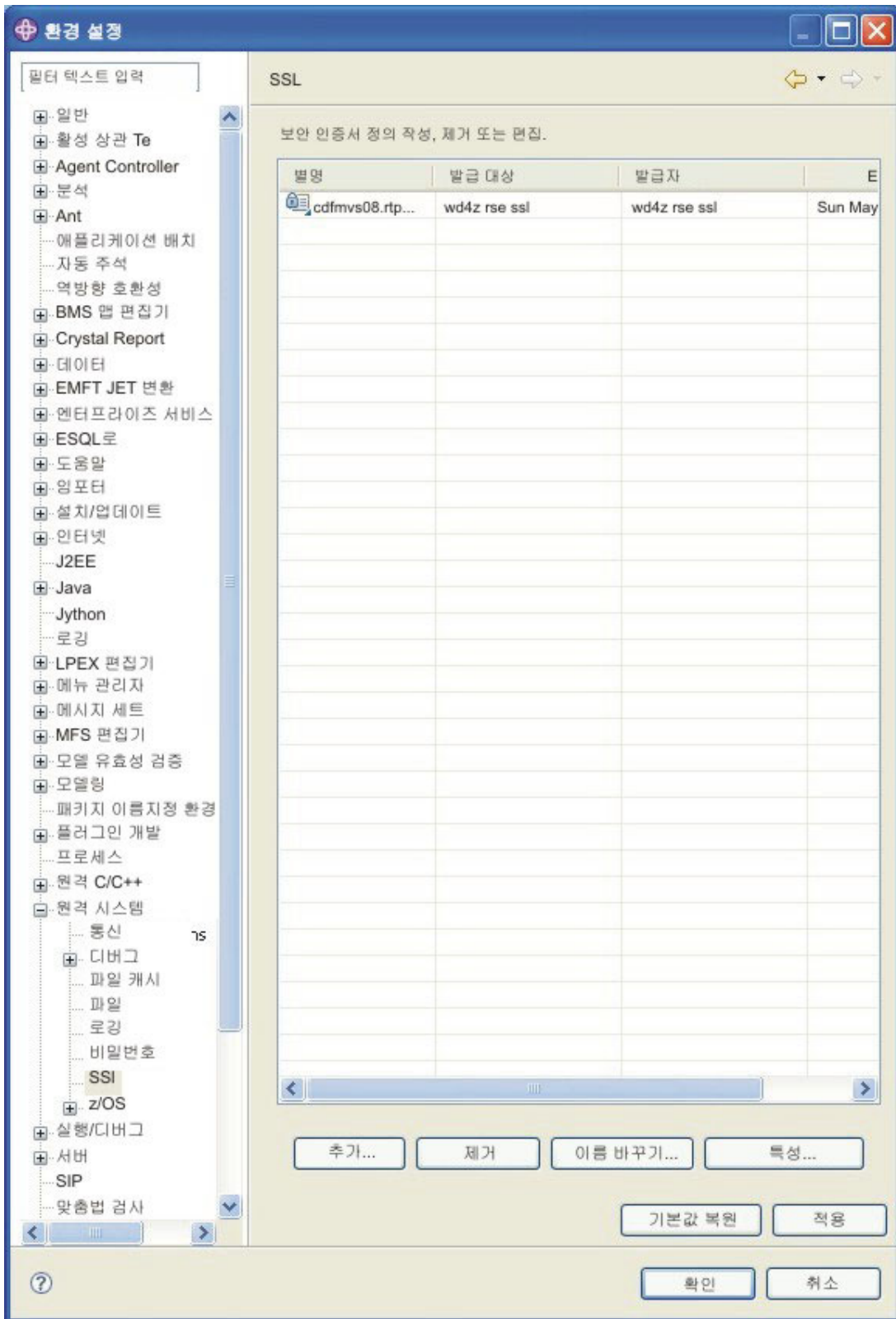


그림 32. 환경 설정 대화 상자 - SSL

SSL 통신이 실패하면 클라이언트가 오류 메시지를 리턴합니다. 자세한 정보는 다른 서버 및 사용자 로그 파일을 참조하십시오(192 페이지의 『RSE 디먼 및 스프레드 폴 로깅』 및 193 페이지의 『RSE 사용자 로깅』의 설명 참조).

(선택사항) X.509 클라이언트 인증 지원 추가

RSE 디먼은 사용자가 X.509 인증서를 사용하여 스스로를 인증하는 것을 지원합니다. SSL 암호화된 통신은 SSL에서 사용되는 인증서를 사용하여 호스트 인증을 확장한 것이기 때문에 이 기능을 작동하려면 SSL 암호화된 통신을 사용하는 것이 전제조건입니다.

35 페이지의 『X.509 인증서를 사용한 클라이언트 인증』에 설명된 대로 사용자에게 대한 인증서 인증을 수행하는 여러 가지 방법이 있습니다. 다음 단계는 보안 소프트웨어가 HostIdMappings 인증서 확장을 사용하여 인증서를 인증하는 방법을 지원하는 데 필요한 설정을 설명합니다.

1. 클라이언트 인증서에 서명하는 데 사용된 인증 기관(CA)을 식별하는 인증서를 신뢰성이 높은 CA 인증서로 변경하십시오. 인증서 유효성 검증에는 TRUST 상태면 충분하지만 로그인 프로세스의 인증서 인증 파트에 사용되기 때문에 HIGHTRUST로의 변경이 수행됩니다.

```
RACDCERT CERTAUTH ALTER(LABEL('HighTrust CA')) HIGHTRUST
```

2. 클라이언트 인증서의 유효성을 검증할 수 있도록 rdzssl.racf 키 링에 CA 인증서를 추가하십시오.

```
RACDCERT ID(stcrse) CONNECT(CERTAUTH LABEL('HighTrust CA') +  
RING(rdzssl.racf))
```

이렇게 하면 CA 인증서에 대한 보안 소프트웨어 설정이 마무리됩니다.

3. 클라이언트 인증서의 HostIdMappings 확장에 정의된 호스트 이름 (IRR.HOST.hostname)에 대해 SERVAUTH 클래스에 자원 (CDFMVS08.RALEIGH.IBM.COM 형식)을 정의하십시오.

```
RDEFINE SERVAUTH IRR.HOST.CDFMVS08.RALEIGH.IBM.COM UACC(NONE)
```

4. RSE 시작된 태스크 사용자 ID(STCRSE)에 READ 권한을 사용하여 이 자원에 대한 액세스를 부여하십시오.

```
PERMIT IRR.HOST.CDFMVS08.RALEIGH.IBM.COM CLASS(SERVAUTH) +  
ACCESS(READ) ID(stcrse)
```

5. 변경사항을 SERVAUTH 클래스에 활성화하십시오. SERVAUTH 클래스가 아직 활성화되어 있지 않으면 첫 번째 명령을 사용하십시오. 활성 설정을 새로 고치려면 두 번째 명령을 사용하십시오.

```
SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)  
or  
SETROPTS RACLIST(SERVAUTH) REFRESH
```

이렇게 하면 HostIdMappings 확장에 대한 보안 소프트웨어 설정이 마무리됩니다.

6. RSE 시작된 태스크를 다시 시작하여 X.509 인증서를 사용한 클라이언트 로그인 허용을 시작하십시오.

(선택사항) gskkyman을 사용하여 키 데이터베이스 작성

RSE 디먼 키 데이터베이스에 SAF 준수 키 링을 사용하는 경우 이 단계를 실행하지 마십시오.

gskkyman은 개인 키, 인증서 요청, 인증서를 포함하는 z/OS UNIX 파일을 작성하여 채우고 관리하는 z/OS UNIX 셸 기반 메뉴 방식 프로그램입니다. 이 z/OS UNIX 파일을 키 데이터베이스라고 합니다.

참고: gskkyman에 대한 환경을 설정하려면 다음 명령문이 필요합니다. 이에 대한 자세한 정보는 >System SSL Programming(SC24-5901)을 참조하십시오.

```
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

이전 샘플은 비밀번호 **rsessl**을 사용하여 **rdzssl.kdb**라는 키 데이터베이스를 작성하는 것으로 시작합니다. 데이터베이스가 있으면 약 10년간(윤일은 계산하지 않음) 유효한 자체 서명 인증서를 새로 작성하여 데이터베이스를 채웁니다. 인증서는 **rdzrse** 레이블 아래에 저장되며 키 데이터베이스에 사용되는 것과 동일한 비밀번호(**rsessl**)를 사용하여 저장됩니다(**RSE** 필수조건임).

gskkyman은 (매우 안전한) 600 권한 비트 마스크(소유자만 액세스 권한을 가짐)를 사용하여 키 데이터베이스를 할당합니다. 디먼이 키 데이터베이스 작성자와 동일한 사용자 ID를 사용하지 않는 경우 권한을 덜 제한적으로 설정해야 합니다. 644(소유자는 읽기/쓰기, 모든 사람은 읽기 권한을 가짐)는 **chmod** 명령에 사용 가능한 마스크입니다.

다음과 같이 키 및 인증서 관리 하위 메뉴에서 인증서 정보 표시 옵션을 선택하여 결과를 확인할 수 있습니다.

\$ 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

다음 ssl.properties 샘플은 daemon_* 지시문이 앞서 표시된 SAF 키 링 샘플과 다름을 보여줍니다.

```
$ oedit /etc/rdz/ssl/ssl.properties
-> change: enable_ssl=true
-> uncomment and change: daemon_keydb_file=rdzssl.kdb
-> uncomment and change: daemon_keydb_password=rsessl
-> uncomment and change: daemon_key_label=rdzrse
```

```
-> uncomment and change: server_keystore_file=rdzssl.racf
-> uncomment and change: server_keystore_label=rdzrse
-> uncomment and change: server_keystore_type=JCERACFKS
```

이전 변경사항은 SSL을 사용할 수 있게 하고 비밀번호 rsessl을 사용하여 rdzssl.kdb 키 데이터베이스의 rdzrse 레이블 아래에 인증서가 저장되어 있음을 RSE 디먼에 알립니다. RSE 서버는 여전히 SAF 준수 키 링을 사용 중입니다.

(선택사항) keytool을 사용하여 키 저장소 작성

RSE 서버 키 저장소에 SAF 준수 키 링을 사용하는 경우 이 단계를 실행하지 마십시오.

"keytool -genkey"는 개인 키 쌍 및 일치하는 자체 서명 인증서((새) 키 저장소 파일에 (별명으로 식별된) 항목으로 저장됨)를 생성합니다.

참고: 명령 검색 디렉토리에 Java가 포함되어야 합니다. keytool을 실행하려면 다음 명령문이 필요합니다. 여기서 /usr/lpp/java/J5.0은 Java가 설치된 디렉토리 (PATH=\$PATH:/usr/lpp/java/J5.0/bin)입니다.

모든 정보를 매개변수로 전달할 수 있지만 명령행 길이 제한사항으로 인해 다음과 같이 일부 대화식 작업이 필요합니다.

```
$ 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
```

이전 예제에서 작성된 자체 서명 인증서는 약 10년간 유효합니다(윤일은 계산하지 않음). 이 인증서는 별명 rdzrse를 사용하여 /etc/rdz/ssl/rdzssl.jks에 저장됩니다. 인증서 비밀번호(rsessl)는 키 저장소 비밀번호와 동일합니다(RSE 필수조건임).

다음과 같이 list 옵션을 사용하여 결과를 확인할 수 있습니다.

```
$ 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
```

다음 ssl.properties 샘플은 server_* 지시문이 앞서 표시된 SAF 키 링 샘플과 다름을 보여줍니다.

```
$ oedit /etc/rdz/ssl/ssl.properties
-> change: enable_ssl=true
-> uncomment and change: daemon_keydb_file=rdzssl.racf
-> uncomment and change: daemon_key_label=rdzrse
-> uncomment and change: server_keystore_file=rdzssl.jks
-> uncomment and change: server_keystore_password=rsessl
-> uncomment and change: server_keystore_label=rdzrse
-> optionally uncomment and change: server_keystore_type=JKS
```

이전 변경사항은 SSL을 사용할 수 있게 하고 비밀번호 rsessl을 사용하여 rdzssl.jks 키 저장소의 rdzrse 레이블 아래에 인증서가 저장되어 있음을 RSE 서버에 알립니다. RSE 디먼은 여전히 SAF 준수 키 링을 사용 중입니다.

제 14 장 TCP/IP 설정

이 부록은 TCP/IP를 설정할 때 또는 기존 설정을 확인하거나 수정할 때 발생할 수 있는 일반적인 문제점 해결에 유용한 정보를 제공합니다.

TCP/IP 구성에 대한 추가 정보는 *Communications Server: IP Configuration Guide*(SC31-8775) 및 *Communications Server: IP Configuration Reference* (SC31-8776)를 참조하십시오.

호스트 이름 종속성

TSO 명령 서비스에 APPC를 사용하는 경우 Developer for System z는 초기화 시 올바른 호스트 이름을 갖는 TCP/IP에 종속됩니다. 이는 다른 TCP/IP 및 분석기 구성 파일이 올바르게 설정되어 있어야 함을 의미합니다.

fekfivpt VIP(Installation Verification Program)를 사용하여 TCP/IP 구성을 테스트할 수 있습니다. 이 명령은 이 샘플의 출력과 유사한 출력을 리턴해야 합니다(\$는 z/OS UNIX 프롬프트임).

```
$ 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) ResolveVia         = UDP
(*) Options NDots      = 1
(*) SockNoTestStor     = NO
(L) ResolverTimeout    = 10
(L) ResolverUdpRetries = 1
(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
Tcpi 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

```

분석기 이해

분석기는 이름을 주소로 또는 주소를 이름으로 분석하기 위해 이름 서버에 액세스하는 클라이언트로서 프로그램을 대신합니다. 요청 프로그램에 대한 조회를 분석하기 위해 분석기는 사용 가능한 이름 서버에 액세스하고 로컬 정의(예: /etc/resolv.conf, /etc/hosts, /etc/ipnodes, HOSTS.SITEINFO, HOSTS.ADDRINFO 또는 ETC.IPNODES)를 사용하거나 둘 다를 조합하여 사용합니다.

분석기 주소 공간이 시작되면, 분석기 JCL 프로시저에서 SETUP DD 카드가 가리키는 선택적 분석기 설정 데이터 세트를 읽습니다. 설정 정보가 제공되지 않은 경우, 분석기는 GLOBALTCPIPDATA, DEFAULTTCPIPDATA, GLOBALIPNODES, DEFAULTIPNODES 또는 COMMONSEARCH 정보 없이 적용 가능한 원시 MVS 또는 z/OS UNIX 검색 순서를 사용합니다.

구성 정보 검색 순서 이해

TCP/IP 기능이 사용하는 구성 파일 검색 순서와 기본 검색 순서를 환경 변수, JCL 또는 사용자가 제공하는 기타 변수로 대체할 수 있는 시기를 이해하는 것이 중요합니다. 이를 알고 있으면 로컬 데이터 세트 및 HFS 파일 이름 지정 표준을 수용할 수 있으며, 문제점 진단 시 사용 중인 구성 데이터 세트 또는 HFS 파일을 알면 유용합니다.

유의해야 할 또 다른 중요한 점은 구성 파일에 검색 순서가 적용될 때 첫 번째 파일을 찾으면 검색이 종료된다는 것입니다. 따라서 다른 파일이 검색 순서에 조기에 있거나 애플리케이션이 선택한 검색 순서에 파일이 포함되어 있지 않기 때문에 결코 찾을 수 없는 파일에 구성 정보를 배치하면 예기치 않은 결과가 발생할 수 있습니다.

구성 파일을 검색할 때 JCL 프로시저에서 DD 문을 사용하거나 환경 변수를 설정하여 대부분의 구성 파일이 있는 위치를 TCP/IP에 명시적으로 알려 줄 수 있습니다. 그렇지

않으면, *Communications Server: IP Configuration Guide*(SC31-8775)에 설명된 검색 순서를 기반으로 TCP/IP가 구성 파일 위치를 동적으로 결정하게 할 수 있습니다.

TCP/IP 스택의 구성 컴포넌트는 TCP/IP 스택 초기화 중에 TCPIP.DATA를 사용하여 스택의 HOSTNAME을 결정합니다. 값을 가져오기 위해 z/OS UNIX 환경 검색 순서가 사용됩니다.

참고: 분석기가 사용 중인 TCPIP.DATA 값과 이 값을 읽은 위치를 결정하려면 추적 분석기 기능을 사용하십시오. 동적으로 추적을 시작하는 방법에 대한 정보는 *Communications Server: IP Diagnosis Guide*(GC31-8782)를 참조하십시오. 추적이 활성화되면, TSO **NETSTAT HOME** 명령 및 z/OS UNIX 셸 **netstat -h** 명령을 실행하여 값을 표시하십시오. TSO 및 z/OS UNIX 셸에서 호스트 이름에 대한 PING을 실행하면 구성할 수 있는 DNS 서버에 대한 활동도 표시됩니다.

z/OS UNIX 환경에서 사용하는 검색 순서

검색하는 특정 파일 또는 테이블은 분석기 구성 설정과 시스템의 특정 파일 존재 여부에 따라 MVS 데이터 세트 또는 HFS 파일입니다.

기본 분석기 구성 파일

기본 분석기 구성 파일은 TCPIP.DATA 문을 포함합니다. 이 절에 지정된 일부 구성 파일에 액세스할 때 사용될 데이터 세트 접두부(DATASETPREFIX 문 값)를 결정하기 위해 분석기 지시문 이외에 기본 분석기 구성 파일도 참조합니다.

기본 분석기 구성 파일에 액세스하는 데 사용되는 검색 순서는 다음과 같습니다.

1. GLOBALTCPIPDATA

정의된 경우, 분석기 GLOBALTCPIPDATA 설정 명령문 값이 사용됩니다(228 페이지의 『분석기 이해』도 참조). 추가 구성 파일에 대해 검색이 계속됩니다. 다음 파일을 찾으면 검색이 종료됩니다.

2. RESOLVER_CONFIG 환경 변수 값

이 환경 변수 값이 사용됩니다. 파일이 없거나 파일이 다른 데서 독점적으로 할당된 경우 검색에 실패합니다.

3. /etc/resolv.conf

4. //SYSTCPD DD 카드

DD 이름 SYSTCPD에 할당된 데이터 세트가 사용됩니다. z/OS UNIX 환경에서 하위 프로세스는 SYSTCPD DD에 액세스할 수 없습니다. 그 이유는 SYSTCPD 할당이 fork() 또는 exec 함수 호출을 통해 상위 프로세스에서 상속되지 않기 때문입니다.

5. userid.TCPIP.DATA

userid는 현재 보안 환경(주소 공간, 태스크 또는 스레드)과 연관된 사용자 ID입니다.

6. **jobname.TCPIP.DATA**

jobname은 일괄처리 작업의 경우 JOB JCL 명령문에 지정된 이름 또는 시작된 프로시저의 경우 프로시저 이름입니다.

7. **SYS1.TCPPARMS(TCPDATA)**

8. **DEFAULTTCPIPDATA**

정의된 경우, 분석기 DEFAULTTCPIPDATA 설정 명령문 값이 사용됩니다(228 페이지의 『분석기 이해』도 참조).

9. **TCPIP.TCPIP.DATA**

변환 테이블

사용될 변환 데이터 세트를 결정하기 위해 변환 테이블(EBCDIC에서 ASCII로, ASCII에서 EBCDIC로)을 참조합니다. 이 구성 파일에 액세스하는 데 사용되는 검색 순서는 다음과 같습니다. 검색 순서는 찾은 첫 번째 파일에서 종료됩니다.

1. **X_XLATE** 환경 변수 값. 이 환경 변수 값은 TSO CONVXLAT 명령이 작성한 변환 테이블의 이름입니다.

2. **userid.STANDARD.TCPXLBIN**

userid는 현재 보안 환경(주소 공간 또는 태스크/스레드)과 연관된 사용자 ID입니다.

3. **jobname.STANDARD.TCPXLBIN**

jobname은 일괄처리 작업의 경우 JOB JCL 명령문에 지정된 이름 또는 시작된 프로시저의 경우 프로시저 이름입니다.

4. **hlq.STANDARD.TCPXLBIN**

hlq는 기본 분석기 구성 파일(있는 경우)에 지정된 DATASETPREFIX 문 값을 나타냅니다. 그렇지 않으면, hlq는 기본적으로 TCPIP입니다.

5. 테이블을 찾을 수 없는 경우, 분석기는 SEZATCPX(STANDARD) 데이터 세트 멤버에 나열된 테이블과 동일한 하드 코드화된 기본 테이블을 사용합니다.

로컬 호스트 테이블

기본적으로 분석기는 먼저 분석 요청에 구성된 도메인 이름 서버를 사용하려고 시도합니다. 분석 요청을 충족시킬 수 없으면, 로컬 호스트 테이블이 사용됩니다. 분석기 동작은 TCPIP.DATA 문을 사용하여 제어됩니다.

TCPIP.DATA 분석기 명령문은 도메인 이름 서버 사용 여부와 방법을 정의합니다. LOOKUP TCPIP.DATA 문을 사용해서도 도메인 이름 서버 및 로컬 호스트 테이블 사용 방법을 제어할 수 있습니다. TCPIP.DATA 문에 대한 자세한 정보는 *Communications Server: IP Configuration Reference*(SC31-8776)를 참조하십시오.

분석기는 getnetbyname API 호출을 위해 무조건적으로 사이트 이름 정보에 대한 Ipv4 고유 검색 순서를 사용합니다. 사이트 이름 정보에 대한 Ipv4 고유 검색 순서는 다음과 같습니다. 검색은 찾은 첫 번째 파일에서 종료됩니다.

1. **X_SITE** 환경 변수 값

이 환경 변수 값은 TSO **MAKESITE** 명령이 작성한 HOSTS.SITEINFO 정보 파일의 이름입니다.

2. **X_ADDR** 환경 변수 값

이 환경 변수 값은 TSO **MAKESITE** 명령이 작성한 HOSTS.ADDRINFO 정보 파일의 이름입니다.

3. **/etc/hosts**

4. **userid.HOSTS.SITEINFO**

userid는 현재 보안 환경(주소 공간 또는 태스크/스레드)과 연관된 사용자 ID입니다.

5. **jobname.HOSTS.SITEINFO**

jobname은 일괄처리 작업의 경우 JOB JCL 명령문에 지정된 이름 또는 시작된 프로시저의 경우 프로시저 이름입니다.

6. **hlq.HOSTS.SITEINFO**

hlq는 기본 분석기 구성 파일(있는 경우)에 지정된 DATASETPREFIX 문 값을 나타냅니다. 그렇지 않으면, hlq는 기본적으로 TCPIP입니다.

Developer for System z에 이 설정 정보 적용

앞서 설명한 바와 같이 APPC를 사용하는 경우 Developer for System z는 초기화 시 올바른 호스트 이름을 갖는 TCP/IP에 종속됩니다. 이는 다른 TCP/IP 및 분석기 구성 파일이 올바르게 설정되어 있어야 함을 의미합니다.

다음 예제는 TCP/IP 및 분석기에 대한 일부 구성 태스크에 초점을 둡니다. 여기서는 TCP/IP 또는 분석기의 전체 설정을 다루지 않으며 사용자의 사이트에 적용할 수 있는 몇 가지 주요 측면만 강조합니다.

1. 다음 JCL에서는 TCP/IP가 SYS1.TCPPARMS(TCPDATA)를 사용하여 스택의 호스트 이름을 결정함을 알 수 있습니다.

```
//TCPIP    PROC PARMS='CTRACE(CTIEZB00)',PROF=TCPPROF,DATA=TCPDATA
//*
//* TCP/IP NETWORK
//*
//TCPIP    EXEC PGM=EZBTCPIP,REGION=0M,TIME=1440,PARM=&PARMS
//PROFILE DD DISP=SHR,DSN=SYS1.TCPPARMS(&PROF)
//SYSTCPD DD DISP=SHR,DSN=SYS1.TCPPARMS(&DATA)
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//ALGPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CFGPRINT DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSOUT   DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=132,BLKSIZE=136)
//SYSERROR DD SYSOUT=*
```

2. SYS1.TCPPARMS(TCPDATA)는 시스템 이름을 호스트 이름으로 사용하고 도메인 이름 서버(DNS)를 사용하지 않음을 알려 줍니다. 모든 이름은 사이트 테이블 검색을 통해 분석됩니다.

```
; HOSTNAME specifies the TCP host name of this system. If not
; specified, the default HOSTNAME will be the node name specified
; in the IEFSSNxx PARMLIB member.
;
; HOSTNAME
;
; DOMAINORIGIN specifies the domain origin that will be appended
; to host names passed to the resolver. If a host name contains
; any dots, then the DOMAINORIGIN will not be appended to the
; host name.
;
DOMAINORIGIN  RALEIGH.IBM.COM
;
; NSINTERADDR specifies the IP address of the name server.
; LOOPBACK (14.0.0.0) specifies your local name server. If a name
; server will not be used, then do not code an NSINTERADDR statement.
; (Comment out the NSINTERADDR line below). This will cause all names
; to be resolved via site table lookup.
;
; NSINTERADDR  14.0.0.0
;
; TRACE RESOLVER will cause a complete trace of all queries to and
; responses from the name server or site tables to be written to
; the user's console. This command is for debugging purposes only.
;
; TRACE RESOLVER
```

3. 분석기 JCL에서는 SETUP DD 문이 사용되지 않음을 알 수 있습니다. 228 페이지의 『분석기 이해』에 설명된 대로 이는 GLOBALTCPIPDATA 및 기타 변수가 사용되지 않음을 의미합니다.

```
//RESOLVER PROC PARMS='CTRACE(CTIRES00)'
//*
//* IP NAME RESOLVER – START WITH SUB=MSTR
//*
//RESOLVER EXEC PGM=EZBREINI,REGION=0M,TIME=1440,PARM=&PARMS
//*SETUP    DD DISP=SHR,DSN=USER.PROCLIB(RESSETUP),FREE=CLOSE
```

4. RESOLVER_CONFIG 환경 변수가 설정되어 있지 않다고 가정하면, 분석기가 /etc/resolv.conf를 기본 구성 파일로 사용함을 표 40에서 알 수 있습니다.

```
TCPIPJOBNAME TCPIP
DomainOrigin RALEIGH.IBM.COM
HostName CDFMVS08
```

229 페이지의 『z/OS UNIX 환경에서 사용하는 검색 순서』에 설명된 대로 기본 구성 파일은 TCPIP.DATA 문을 포함합니다. 시스템 이름이 CDFMVS08이면(시스템 이름이 호스트 이름으로 사용됨을 TCPDATA가 설명), /etc/resolv.conf가 SYS1.TCPPARMS(TCPDATA)와 동기화됨을 알 수 있습니다. DNS 정의가 없으므로 사이트 테이블 검색이 사용됩니다.

5. 표 40는 기본 ASCII-EBCDIC 변환 테이블을 사용하기 위해 아무것도 수행할 필요가 없음도 알려줍니다.
6. TSO **MAKESITE** 명령이 사용되지 않는다고 가정하면(X_SITE 및 X_ADDR 변수를 작성할 수 있음) /etc/hosts가 이름 검색에 사용되는 사이트 테이블입니다.

```
# Resolver /etc/hosts file cdfmvs08
9.42.112.75    cdfmvs08                # CDFMVS08 Host
9.42.112.75    cdfmvs08.raleigh.ibm.com    # CDFMVS08 Host
127.0.0.1      localhost
```

이 파일의 최소 콘텐츠는 현재 시스템에 대한 정보입니다. 이전 샘플에서는 cdfmvs08과 cdfmvs08.raleigh.ibm.com 둘 다 z/OS 시스템의 IP 주소에 올바른 이름으로 정의됩니다.

도메인 이름 서버(DNS)를 사용 중이면, DNS가 /etc/hosts 정보를 보유하며 /etc/resolv.conf 및 SYS1.TCPPARMS(TCPDATA)에는 시스템에 DNS를 식별하는 명령문이 있습니다.

혼란을 방지하기 위해 TCP/IP 및 분석기 구성 파일을 서로 동기화해야 합니다.

표 40. 분석기에 사용 가능한 로컬 정의

파일 유형 설명	영향받는 API	후보 파일
기본 분석기 구성 파일	모든 API	1. GLOBALTCPIPDATA 2. RESOLVER_CONFIG 환경 변수 3. /etc/resolv.conf 4. SYSTCPD DD 이름 5. userid.TCPIP.DATA 6. jobname.TCPIP.DATA 7. SYS1.TCPPARMS(TCPDATA) 8. DEFAULTTCPIPDATA 9. TCPIP.TCPIP.DATA

표 40. 분석기에 사용 가능한 로컬 정의 (계속)

파일 유형 설명	영향받는 API	후보 파일
변환 테이블	모든 API	<ol style="list-style-type: none"> 1. X_XLATE 환경 변수 2. userid.STANDARD.TCPXLBIN 3. jobname.STANDARD.TCPXLBIN 4. hlq.STANDARD.TCPXLBIN 5. 분석기 제공 변환 테이블, SEZATCPX의 STANDARD 멤버
로컬 호스트 테이블	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. X_SITE 환경 변수 2. X_ADDR 환경 변수 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. RESOLVER_IPNODES 환경 변수 3. userid.ETC.IPNODES 4. jobname.ETC.IPNODES 5. hlq.ETC.IPNODES 6. DEFAULTIPNODES 7. /etc/ipnodes

참고: 233 페이지의 표 40는 *Communications Server: IP Configuration Guide*(SC31-8775)의 표를 일부 복사한 것입니다. 전체 표는 해당 매뉴얼을 참조하십시오.

호스트 주소가 올바르게 분석되지 않음

TCP/IP 분석기가 호스트 주소를 올바르게 분석할 수 없는 문제점이 나타나는 경우, 분석기 구성 파일이 누락되었거나 불완전하기 때문일 가능성이 높습니다. 이러한 문제가 있음을 분명히 나타내는 것은 lock.log의 다음 메시지입니다.

```
clientip(0.0.0.0) <> callerip(<host IP address>)
```

이를 확인하려면 *Host Configuration Guide* (SC23-7658)의 "설치 검증"에 설명된 대로 fekfivpt TCP/IP IVP를 실행하십시오. 출력의 분석기 구성 섹션은 다음 샘플과 유사합니다.

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
```

“Local Tcp/Ip Dataset”가 참조하는 파일(또는 데이터 세트)의 정의가 올바른지 확인하십시오.

IP 분석기 파일(z/OS UNIX 검색 순서 사용)에 기본 이름을 사용하지 않는 경우 이 필드는 공백입니다. 이 경우, `rsed.envvars`에 다음 명령문을 추가하십시오. 여기서 `<resolver file>` 또는 `<resolver data>`는 IP 분석기 파일의 이름을 나타냅니다.

```
RESOLVER_CONFIG=<resolver file>
```

또는

```
RESOLVER_CONFIG='<resolver data set>'
```

제 2 부 부록

참고 문헌

참조된 서적

이 책에 참조된 서적은 다음과 같습니다.

표 41. 참조된 서적

책 제목	주문 번호	참조	참조 웹 사이트
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 전제조건	SC23-7659	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z 호스트 구성 빠른 시작	GA30-4183	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z Host Configuration Guide	SC23-7658	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z 호스트 구성 참조서	SA30-4501	Developer for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Rational Developer for System z 호스트 구성 유틸리티 안내서	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 Answers to common host configuration and maintenance issues	SC14-7373	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 전제조건	SC23-7659	Developer for System z	http://www.ibm.com/software/rational/products/developer/systemz/library/index.html
Rational Developer for System z 호스트 구성 빠른 시작	GA30-4183	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	백서	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Using ISPF Client Gateway to provide CARMA services	SC14-7292	백서	http://www-01.ibm.com/support/docview.wss?uid=swg27038517

표 41. 참조된 서적 (계속)

책 제목	주문 번호	참조	참조 웹 사이트
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/
I MVS 진단: 도구 및 서비스 지원	GA22-7589	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL Reference	SA22-7597	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning APPC/MVS Management	SA22-7599	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning Workload Management	SA22-7602	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS System Commands	SA22-7627	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E Customization	SA22-7783	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
TSO/E REXX Reference	SA22-7790	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
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/

표 41. 참조된 서적 (계속)

책 제목	주문 번호	참조	참조 웹 사이트
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	CICSTS 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	CICSTS 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 for z/OS	http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html

이 책에 참조된 웹 사이트 다음과 같습니다.

표 42. 참조된 웹 사이트

설명	참조 웹 사이트
Developer for System z Information Center	http://pic.dhe.ibm.com/infocenter/ratdevz/v9r0/index.jsp
Developer for System z 라이브러리	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Developer for System z 홈 페이지	http://www-03.ibm.com/software/products/us/en/developerforsystemz/
Developer for System z 권장 서비스	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335
Developer for System z 향상 요청	https://www.ibm.com/developerworks/support/rational/rfe/
z/OS 인터넷 라이브러리	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
CICSTS Information Center	https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp

표 42. 참조된 웹 사이트 (계속)

설명	참조 웹 사이트
IBM Tivoli Directory Server	http://www-01.ibm.com/software/tivoli/products/directory-server/
문제점 판별 도구 플러그인	http://www-01.ibm.com/software/awdtools/deployment/pdtpplugins/
Apache Ant 다운로드	http://ant.apache.org/
Java keytool 문서	http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html
CA 지원 홈 페이지	https://support.ca.com/

정보 서적

다음 서적은 필수 호스트 시스템 컴포넌트에 대한 설정 문제점을 이해하는 데 도움이 됩니다.

표 43. 정보 서적

책 제목	주문 번호	참조	참조 웹 사이트
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/

용어집

조치 ID(Action ID)

조치에 대한 숫자 ID(0 - 999)

애플리케이션 서버(Application Server)

1. 브라우저 기반 컴퓨터와 조직의 백엔드 비즈니스 애플리케이션 또는 데이터베이스 사이에서 모든 애플리케이션 조작을 처리하는 프로그램. Java EE 표준을 준수하는 Java 기반의 appserver라는 특수 클래스가 있습니다. Java EE 코드는 이 appserver 간에 쉽게 이식이 가능합니다. appserver는 동적 웹 콘텐츠를 위한 JSP 및 Servlet을, 트랜잭션 및 데이터베이스 액세스를 위한 EJB를 지원할 수 있습니다.
2. 원격 애플리케이션으로부터의 요청 대상. DB2 환경에서 애플리케이션 서버 기능은 분산 데이터 기능에 의해 제공되며 원격 애플리케이션에서 DB2 데이터에 액세스하는 데 사용됩니다.
3. 애플리케이션의 실행 환경을 제공하는 분산 네트워크의 서버 프로그램
4. 애플리케이션 요청자로부터의 요청 대상. 애플리케이션 서버 사이트의 데이터베이스 관리 시스템(DBMS)에서 요청된 데이터를 제공합니다.
5. 콘텐츠 관리자의 자산 및 조회를 요청하는 클라이언트와의 통신을 처리하는 소프트웨어

양방향(Bidirectional, bi-di)

아랍어, 히브리어와 같이 일반적으로 오른쪽에서 왼쪽으로 실행되는 스크립트와 관련됩니다(왼쪽에서 오른쪽으로 실행되는 숫자는 제

외). 이 정의는 LISA(Localization Industry Standards Association) 용어집에서 차용한 것입니다.

양방향 속성(Bidirectional Attribute)

텍스트 유형, 텍스트 방향, 숫자 스와핑 및 대칭형 스와핑

빌드 요청(Build Request)

빌드 트랜잭션을 수행하기 위한 클라이언트의 요청

빌드 트랜잭션(Build Transaction)

클라이언트로부터 빌드 요청을 받은 후 빌드를 수행하기 위해 MVS에서 시작된 작업

컴파일(Compile)

1. ILE(Integrated Language Environment) 언어에서 프로그램 또는 서비스 프로그램에 바인드할 수 있는 모듈로 소스 명령문을 변환하는 것
2. 고급 언어로 표현된 프로그램 모두 또는 일부를 중간 언어 즉, 어셈블리 언어나 기계어로 표현된 컴퓨터 프로그램으로 변환하는 것

컨테이너(Container)

1. CoOperative Development Environment/400에서 소스 파일을 포함 및 구성하는 시스템 오브젝트. 컨테이너의 예로는 i5/OS™ 라이브러리 또는 MVS 파티션된 데이터 세트가 있습니다.
2. Java EE에서 라이프사이클 관리, 보안, 배치, 런타임 서비스를 컴포넌트에 제공하는 엔티티. (Sun) 컨테이너 유형(EJB, 웹,

JSP, Servlet, 애플릿 및 애플리케이션 클라이언트)마다 컴포넌트별로 서비스도 제 공합니다.

3. 백업 복구 및 매체 서비스에서 상자, 케 이스, 결이 등 매체를 저장, 이동하는 데 사용되는 실제 오브젝트
4. 가상 테이프 서버(VTS)에서 내보낸 논리 적 볼륨(LVOL)을 하나 이상 저장할 수 있는 저장소. 하나 이상의 LVOL이 있고 VTS 라이브러리 밖에 상주하는 스택된 볼륨은 해당 볼륨의 컨테이너로 간주됩 니다.
5. 데이터의 실제 저장 위치(예: 파일, 디렉 토리 또는 디바이스)
6. 페이지에서 포틀릿이나 다른 컨테이너의 레이아웃을 배열하기 위해 사용되는 열 또는 행
7. 오브젝트를 보유하는 사용자 인터페이스 의 요소. 폴더 관리자의 경우 다른 폴더 또는 문서를 포함하는 오브젝트에 해당합 니다.

데이터베이스(Database)

하나 이상의 애플리케이션을 서비스하기 위해 함께 저장되는 상호 관련 또는 독립된 데이 터 항목의 컬렉션

데이터 정의 보기(Data Definition View)

데이터베이스와 그 오브젝트의 로컬 표시가 포함되어 있고 이러한 오브젝트를 조작하여 원격 데이터베이스로 내보내는 기능을 제공합 니다.

데이터 세트(Data Set)

규정된 여러 배열 중 하나의 데이터 컬렉션 으로 구성되고 시스템이 액세스할 수 있는 제 어 정보로 설명되는 데이터 저장 및 검색의 주요 단위

디버그(Debug)

프로그램에서 오류를 발견, 진단, 제거하는 것

디버깅 세션(Debugging Session)

개발자가 디버거를 시작하는 시간과 종료하는 시간 사이에 발생하는 디버깅 활동

오류 버퍼(Error Buffer)

오류 출력 정보를 임시로 보유하는 데 사용 되는 스토리지의 한 부분

게이트웨이(Gateway)

1. 웹 서비스 호출 중 인터넷과 인트라넷 환 경을 연결하는 미들웨어 컴포넌트
2. 엔드포인트와 나머지 Tivoli 환경 사이에 서 서비스를 제공하는 소프트웨어
3. VoIP와 회로 스위치 환경 사이에 브릿지 를 제공하는 VoIP(Voice over Internet Protocol)의 컴포넌트
4. 다른 네트워크 아키텍처와 네트워크 또는 시스템을 연결하는 데 사용되는 디바이스 또는 프로그램. 시스템은 다른 통신 프로 토콜, 다른 네트워크 아키텍처 또는 다른 보안 정책 등 서로 다른 특성을 가질 수 있으며 이 경우 게이트웨이는 연결 역할 뿐 아니라 변환 역할도 수행하게 됩니다.

ISPF(Interactive System Productivity Facility)

전체 화면 편집기와 대화 상자 관리자 역할 을 하는 IBM에 라이선스가 있는 프로그램. 애플리케이션 프로그램을 작성하는 데 사용되 며 애플리케이션 프로그래머와 터미널 사용자 간에 대화식 대화 상자와 표준 화면 패널을

생성하는 수단을 제공합니다. ISPF를 구성하는 4가지 기본 컴포넌트는 DM, PDF, SCLM, C/S입니다. DM(Dialog Manager) 컴포넌트는 대화 상자와 최종 사용자에게 서비스를 제공합니다. PDF 컴포넌트는 대화 상자 또는 애플리케이션 개발자를 지원하는 서비스를 제공하는 프로그램 개발 기능입니다. SCLM(Software Configuration Library Manager) 컴포넌트는 애플리케이션 개발자에게 해당 애플리케이션 개발 라이브러리를 관리하는 서비스를 제공합니다. C/S(Client/Server) 컴포넌트는 프로그래밍 가능한 워크스테이션에서 ISPF를 실행하고 워크스테이션 운영 체제의 표시 기능을 사용하여 패널을 표시하며 호스트 도구 및 데이터와 워크스테이션 도구 및 데이터를 통합할 수 있습니다.

해석기(Interpreter)

고급 프로그래밍 언어의 지시사항을 각각 변환하고 실행한 후 다음 지시사항을 변환하고 실행하는 프로그램

동형(Isomorphic)

루트에서 시작하는 XML 인스턴스 문서의 구성된 각 요소(즉, 다른 요소를 포함하는 요소)에는 중첩 두께가 XML 동등 항목의 중첩 두께와 동일한 COBOL 그룹 항목이 하나씩만 있습니다. 상위에서 시작하는 XML 인스턴스 문서의 각 비복합 요소(즉 기타 요소를 포함하지 않는 요소)에는 중첩 깊이가 대응되는 XML의 중첩 깊이와 동일하며 런타임 시 메모리 주소가 정확하게 동일한 하나의 대응되는 COBOL 기초 항목이 있습니다.

연계 섹션(Linkage Section)

활성화 단위(프로그램 또는 메소드)에서 사용 가능한 데이터 항목을 설명하며, 활성화된 단위(호출된 프로그램 또는 호출된 메소드)의 Data Division에 있는 섹션. 이러한 데이터 항목은 활성화된 단위와 활성화 단위 모두에서 참조할 수 있습니다.

로드 라이브러리(Load Library)

로드 모듈이 있는 라이브러리

잠금 조치(Lock Action)

구성원을 잠급니다.

네비게이터 보기(Navigator View)

워크벤치에서 자원의 계층 구조 보기를 제공합니다.

이형(Non-Isomorphic)

모양이 동일하지 않은(이형) XML 문서와 COBOL 그룹에 속하는 XML 요소와 COBOL 항목의 단순 매핑. 이형 매핑은 동형 구조의 이형 요소 사이에서도 작성될 수 있습니다.

결과물 콘솔 보기(Output Console View)

프로세스의 결과물을 표시하며 프로세스에 키보드 입력을 제공할 수 있습니다.

결과물 보기(Output View)

작업 중인 오브젝트와 관련된 메시지, 매개변수, 결과를 표시합니다.

퍼스펙티브(Perspective)

워크벤치에서 자원의 다양한 측면을 보여주는 보기의 그룹. 워크벤치 사용자는 태스크에 따

라 퍼스펙티브를 전환하고 퍼스펙티브 내에서 보기와 편집기의 레이아웃을 사용자 정의할 수 있습니다.

RAM 저장소 액세스 관리자

원격 파일 시스템(Remote File System)

별도의 서버 또는 운영 체제에 상주하는 파일 시스템

원격 시스템(Remote System)

한 시스템에서 통신할 수 있는 네트워크의 다른 시스템

원격 시스템 퍼스펙티브(Remote Systems Perspective)

ISPF와 유사한 규칙을 사용하여 원격 시스템을 관리할 수 있는 인터페이스를 제공합니다.

저장소(Repository)

1. 데이터 저장 영역. 모든 저장소에는 이름과 연관된 비즈니스 항목 유형이 있습니다. 기본적으로 이 이름은 비즈니스 항목의 이름과 같습니다. 예를 들어, 송장의 저장소 이름은 Invoices가 됩니다. 정보 저장소는 로컬(프로세스별)과 글로벌(재사용 가능)이라는 2가지 유형이 있습니다.
2. BTS 프로세스 상태가 저장되는 VSAM 데이터 세트. 프로세스가 BTS 제어하에 실행되지 않는 경우에는 해당 상태(및 구성 활동의 상태)를 저장소 데이터 세트에 기록하여 유지됩니다. 특정 프로세스 유형에 해당하는 모든 프로세스(및 해당 활동 인스턴스)의 상태는 같은 저장소 데이터 세트에 저장됩니다. 여러 프로세스 유형에 대한 레코드를 같은 저장소에 기록할 수 있습니다.
3. 소스 코드와 기타 애플리케이션 자원의 영구 저장 영역. 팀 프로그래밍 환경에서는

공유 저장소를 통해 다중 사용자가 애플리케이션 자원에 액세스할 수 있습니다.

4. 클러스터 구성원인 큐 관리자에 대한 정보의 컬렉션. 이 정보에는 큐 관리자 이름과 해당 위치, 해당 채널 및 호스트 큐 등이 포함됩니다.

저장소 인스턴스(Repository Instance)

SCM에 존재하는 프로젝트 또는 컴포넌트

저장소 보기(Repositories View)

워크벤치에 추가된 CVS 저장소 위치를 표시합니다.

응답 파일(Response File)

1. 프로그램에서 요청하는 질문에 대한 사전 정의된 응답 세트를 포함하며, 해당 값을 한 번에 하나씩 입력하는 대신 사용되는 파일
2. 설치를 자동화하는 설치 및 구성 데이터로 사용자 정의할 수 있는 ASCII 파일. 대화식 설치 중에 설정 및 구성 데이터를 입력해야 하는 경우도 있지만 응답 파일을 사용하면 사용자 개입 없이 설치를 진행할 수 있습니다.

서버 보기(Servers View)

모든 서버와 해당 서버와 연관된 구성에 대한 목록을 표시합니다.

셸(Shell)

명령 및 사용자 상호작용을 해석하고 이를 운영 체제와 통신하는 사용자와 운영 체제 간 소프트웨어 인터페이스. 컴퓨터에는 다양한 레벨의 사용자 상호작용에 대한 여러 셸 계층이 존재할 수 있습니다.

셸 이름(Shell Name)

셸 인터페이스의 이름

셸 스크립트(Shell Script)

셸에서 해석할 수 있는 명령을 포함하는 파

일. 사용자가 셸 명령 프롬프트에서 스크립트 파일의 이름을 입력하면 셸이 스크립트 명령을 실행합니다.

사이드덱(Sidedeck)

DLL 프로그램의 기능을 공개하는 라이브러리. 소스 코드가 컴파일된 후 이 라이브러리에 항목 이름과 모듈 이름이 저장됩니다.

자동 설치(Silent Installation)

콘솔로 메시지를 보내지 않고 로그 파일에 메시지와 오류를 저장하는 설치. 또한 자동 설치하는 데이터 입력 시 응답 파일을 사용할 수 있습니다.

자동 설치 제거(Silent Uninstallation)

uninstall 명령이 호출된 후 콘솔로 메시지를 보내지 않고 로그 파일에 메시지와 오류를 저장하는 설치 제거 프로세스

태스크 목록(Task List)

단일 제어 플로우로 실행할 수 있는 프로시저 목록

URL URL(Uniform Resource Locator)

IBM Rational Developer for System z의 문서 주의사항

© Copyright IBM Corporation 2009, 2013.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan, Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현 상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함)간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 IBM이 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스에서부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 반드시 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확인할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보는 계획 수립 목적으로만 사용됩니다. 이 정보는 기술된 제품이 GA(General Availability)되기 전에 변경될 수 있습니다.

이 정보에는 비즈니스 오퍼레이션에 사용되는 데이터 및 보고서의 예제가 포함되어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용 없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 일체의 보증 없이 "현상상태로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 어떤 손해에 대해서도 책임을 지지 않습니다.

이러한 샘플 프로그램의 모든 사본 또는 일부 또는 파생된 작업에는 다음과 같은 저작권 주의사항이 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. 2009, 2013.

이 정보를 소프트카피로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

상표 정보

IBM, the IBM 로고 및 ibm.com은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹(www.ibm.com/legal/copytrade.shtml)에 있습니다.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 및/또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록 상표 또는 상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.

기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다.

저작권 라이선스

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용 없이 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 일체의 보증 없이 "현상태대로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 어떤 손해에 대해서도 책임을 지지 않습니다.

상표 정보

IBM, IBM 로고 및 `ibm.com`은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. 현재 IBM 상표 목록은 웹(www.ibm.com/legal/copytrade.shtml)에 있습니다.

CA Endeavor는 CA Technologies의 등록상표입니다.

Rational은 미국 또는 기타 국가에서 사용되는 International Business Machines Corporation과 Rational Software Corporation의 상표입니다.

Intel 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표입니다.

Microsoft, Windows 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표 또는 등록상표입니다.

Java 및 모든 Java 기반 등록 상표와 로고는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc의 상표 또는 등록 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

색인

[가]

가변 사용자 ID, 실행 170
감사 데이터
 로그된 조치 28
감사 로깅, RSE 디먼이 관리 26
감사 제어
 audit.* 옵션 27
 daemon.log 27
 enable.audit.log 27
 _RSE_HOST_CODEPAGE 27
감사 처리
 modify switch 27
개발, 애플리케이션 128
개인 키 및 인증서 저장 위치 212
개인 키 및 인증서, 저장 위치 결정 212
거부된 연결 209
검색 순서, z/OS UNIX 환경 229
고려사항, 보안 21
고려사항, 성능 127
고정 Java 힙 크기 130
공간 사용량, z/OS UNIX 파일 시스템 107
공간 사용량, 메타데이터 138
공존, rsed.envvars를 업데이트하여 공존 사용 216
관리 유틸리티 메시지 164
관리 유틸리티, 마이그레이션 참고사항 164
관리, 워크로드 129
구성 문제점 해결 189
구성 문제점, 문제점 해결 189
구성 정보 검색 순서 228
구성 정보, 검색 순서 228
구성 파일이 다른 동일한 소프트웨어 레벨 182
구성 파일, Developer for System z 44
구성 파일, 기본 분석기 229
구성 파일, 동일한 소프트웨어 레벨, 다른 182
권한 비트, z/OS UNIX 202
그룹 메타데이터 위치 142
그룹 선택, LDAP 기반 144
그룹 선택, SAF 기반 150
그룹 연결 141
기본 대 비1차 연결 리전 156
기본 분석기 구성 파일 229

기본 시스템 136
기본 TCP/IP 동작 대체 70
기본 TCP/IP 동작, 대체 70
기존 ISPF 프로파일 사용 177
기존 RSE 설정 복제 215

[나]

내부 통신 69
네트워크, 모니터링 120

[다]

다양한 자원 정의 114
 EXEC 카드, 서버 JCL 114
 FEJJCNGF 114
 SYS1.PARMLIB(ASCHPMxx) 115
 SYS1.PARMLIB(IEASYSxx) 115
 SYS1.PARMLIB(IVTPRMxx) 115
다중 인스턴스 실행 181
다중 인스턴스, 실행 181
다중 ISPF.conf 파일 178
덤프 위치, z/OS UNIX 199
덤프 파일 197
덤프, Java 197
덤프, MVS 197
데이터 세트 프로파일, 정의 58
동기화, 자동화 183
디렉토리 구조, z/OS UNIX
 그래픽 표시 16
디버거, 통합 11
디버거 관리자 인증 23
디버거 보안 42
디버거, CICS 트랜잭션 167
디스크 공간, JVM(Java Virtual Machine) 132

[라]

라이브러리, LE(Language Environment) 런타임 128
라이브러리, 시스템에 대한 액세스 향상 128
런타임 라이브러리, LE(Language Environment) 128

로그 파일
 audit.log 190
 fa.log 190
 fekfivpi.log 190
 fekfivps.log 190
 ffsget.log 190
 ffsput.log 190
 ffs.log 190
 lock.log 190
 rmt_class_loader.cache.jar 190
 rsecomm.log 190
 rsedaemon.log 190
 rseserver.log 190
 serverlogs.count 190
 stderr.log 190
 stdout.log 190
 .storeMemLogging 190
 .storeTrace 190
로깅, CARMA 195
로깅, fekfivpi IVP 테스트 196
로깅, JES 작업 모니터 192
로깅, RSE 디먼 192
로깅, RSE 사용자 193
로깅, SCLM 개발자 툴킷 194
로깅, 스프레드 풀 192
로깅, 코드 검토 196
로깅, 코드 적용 196
로컬 호스트 테이블 230

[마]

마이그레이션 참고사항, 관리 유틸리티 164
메모리 부족 오류 210
메시지, 관리 유틸리티 164
메타데이터 공간 사용 138
메타데이터 보안 138
메타데이터 위치 137
메타데이터, 클라이언트로 푸시 137
명령 보안, JES 정의 56
모니터링, 네트워크 120
목표 설정, WLM 81
목표, WLM의 설정 81

[바]

방법, 인증 22
변경사항 거부, 유예 기간 152
변환 테이블 230
보안 검사 성능 향상 129
보안 검사 성능, 향상 129
보안 검사, 성능 향상 129
보안 고려사항 21
보안 명령, 유용
 ADDGROUP 19
 ALTUSER 19
 CONNECT 19
보안 설정 및 클래스, 활성화 50
보안 설정 확인 64
보안 설정, 확인 64
보안 소프트웨어를 사용한 인증 37
보안 소프트웨어, 인증 37
보안 정의 48, 151
보안 정의, 체크리스트 48
보안 프로파일, 저장된 제한사항 208
보안 z/OS UNIX 서버, RSE 정의 53
보안, CICSTS 43
보안, JES 29
보안, JES 명령 정의 56
보안, SCLM 44
보안, 디버그 42
보안, 애플리케이션 배치 관리자(ADM) 157
보안, 연결 23
보안, 자원 159
보안, 트랜잭션 157
보안, 파이프라인 157
복수 개발자 그룹 140
분류 규칙, WLM 80
분산 동적 VIPA
 EZBEPOR 72
 PORT 72
 PORTRANGE 72
 SERVERWLM 72
 SYSPLEXPORTS 72
 VIPADISTRIBUTE 72
분석기에 사용 가능한 로컬 정의 233
분석기에 사용 가능한 정의 233
분석기, 사용 가능한 로컬 정의 233
분석기, 이해 228
비밀번호와 사용자 ID 22
비시스템 관리자, 업데이트 권한 18

[사]

사용자 로깅, RSE 193
사용자 엑시트 고려사항 xv, 169
사용자 엑시트 루틴, 기록 169
사용자 엑시트 특성 169
사용자 엑시트 활성화 169
사용자 엑시트점, 사용 가능한 172
사용자 엑시트, 콘솔 메시지 170
사용자 정의 - ISPF.conf, 176
사용자 ID와 비밀번호 22
사용자 ID와 일회성 비밀번호 22
사용자 ID, 실행 170
샘플 설정 120
 스레드 풀 개수 121
 최소 한계 결정 121
 한계 정의 122
샘플 설정, LDAP 그룹 선택 146
샘플 설정, SAF 기반 그룹 선택 151
샘플 스토리지 사용량 분석 103
샘플 스토리지, 사용량 분석 103
서명 인증서, 자체 서명 또는 인증 기관에서 서명 214
서버 선택, LDAP 145
서버 위치, LDAP 146
서브시스템 유형
 ASCH 80
 CICS 80
 JES 80
 OMVS 80
 STC 80
서적, 참조된 239
설정 단계 143
설정 및 클래스, 보안 활성화 50
설정, sysplex에서 동일 182
설치 로깅, CICS 자원 157
성능 고려사항 127
세그먼트, OMVS 정의 51
소개, 클라이언트로 푸시 고려사항 135
소프트웨어 레벨이 동일한 다른 구성 파일 182
소프트웨어 레벨, 다른 구성 파일에서 동일한 182
스레드 개수 97
스레드 풀 로깅 192
스토리지 사용량 101
스티키(Sticky) 비트, z/OS UNIX에 대한 MVS 로드 모듈 가용성 205

스플 파일에 대한 액세스, 조건부 32
스플 파일에 대한 조건부 액세스 32
스플 파일, 조건부 액세스 32
시스템 라이브러리 액세스, 향상 128
시스템 라이브러리에 대한 액세스 향상 128
시스템 라이브러리, 액세스 향상 128
시스템 종료, 강제 실행된 제한사항 208
시스템 한계 209
시작 태스크, Developer for System z에 대한 정의
 JMON 시작된 태스크 51
 RSED 시작된 태스크 51
시작 JCL 요구사항 208
실행 제한사항, 작업에 대한 조치 30
씨드파티 및 X.509 인증 22

[아]

암호화 통신
 통합 디버거 34
암호화된 통신, SSL 43, 159
암호화된 통신, SSL/TLS 33
암호화, SSL 또는 TLS 212
애플리케이션 개발 128
애플리케이션 배치 관리자 보안 157
애플리케이션 배치 관리자 사용자 정의 155
애플리케이션 배치 관리자(ADM) 4
애플리케이션 배치 관리자, CICS 자원 정의 서버 155
애플리케이션 배치 관리자, CICS 자원 정의 편집기 155
애플리케이션 배치 관리자, 사용자 정의 155
엑세스 방법, TSO 175
엑세스 방법, TSO/ISPF Client Gateway 사용 176
업데이트 권한, 비시스템 관리자 18
에뮬레이터, 호스트 연결 210
여러 개의 할당 exec, TSO/ISPF 178
여러 개의 System z용 Developer 설정, 다중 ISPF.conf 파일 사용 178
연결 리전, 기본 대 비기본 156
연결 보안 23
연결 플로우 9
 그래픽 표시 9
연결이 거부됨 209
예약된 TCP/IP 포트 206
예약, TCP/IP 포트 69
오류 피드백 추적 201

- 외부 통신 68
- 외부 통신 제한, 지정된 포트 24
- 요구사항, 시작 JCL 208
- 워크로드 관리 129
- 워크로드 관리자 79
- 워크로드 분류, WLM 79
- 웹 서비스 인터페이스 156
- 웹 소유 리전 156
- 유예 기간, 변경사항 거부 152
- 인증 기관 유효성 검증
 - gskkyman 36
 - SAF 키 링 36
 - TRUST, HIGHTRUST 36
- 인증 방법 22
- 인증서 폐기 목록(CRL) 조회
 - CRL 환경 변수 37
 - rsed.envvars 37
- 인증서 폐기 목록(CRL), 조회
 - CRL 환경 변수 37
 - rsed.envvars 37
- 인증서, X.509를 사용한 클라이언트 인증 35
- 인증, JES 작업 모니터 23
- 인증, SSL 및 X.509 설정 211
- 인증, X.509 22
- 인증, 디버그 관리자 23
- 일회성 비밀번호와 사용자 ID 22
- 임시 자원 사용량 100

[자]

- 자동화된 동기화 183
- 자원 보안 159
- 자원 사용량, 개요 90
- 자원 사용량, 임시 100
- 자원 사용량, 튜닝 89
- 자원 설치 로깅, CICS 157
- 자원 정의, 다양 114
- 작업공간 바인딩 141
- 작업에 대한 조건부 조치 29
- 작업에 대한 조치 - 실행 제한사항 30
- 작업, 조건부 조치 29
- 잠금 디먼 14
- 잠금 디먼 플로우
 - 그래픽 표시 14
- 잠금 디먼(LOCKD) 4
- 잠금 해제
 - RSE, modify cancel 명령 15
- 저장소 보안, CRD 157

- 정의, 보안 48
- 종료점, 사용 가능한 172
- 종속성, 호스트 이름 227
- 주소 공간 계수 91
- 주소 공간 크기 207
- 주소 공간 크기 한계 101
- 지정된 포트에 대한 외부 통신, 제한 24

[차]

- 참조된 서적 239
- 초기 LDAP 그룹 설정 148
- 추적 199
- 추적, CARMA 200
- 추적, JES 작업 모니터 199
- 추적, RSE 199
- 추적, 오류 피드백 201

[카]

- 캐시 관리 유틸리티, JVM(Java Virtual Machine) 132
- 캐시 보안, JVM(Java Virtual Machine) 131
- 캐시 크기 한계, JVM(Java Virtual Machine) 131
- 컴포넌트 개요, Developer for System z
 - 그래픽 표시 4
- 코드 검토 로깅 196
- 코드 적용 로깅 196
- 콘솔 메시지, 사용자 엑시트 170
- 크기 예측, 가이드라인 102
- 크기 한계, Java 힙 101
- 크기 한계, 주소 공간 101
- 크기, 주소 공간 207
- 클라이언트 구성 제어 139
- 클라이언트 기능, 변경 40
- 클라이언트 버전 제어 139
- 클라이언트 인증 지원, X.509 추가 221
- 클라이언트로 푸시 41
- 클라이언트로 푸시 고려사항 135
- 클라이언트로 푸시 메타데이터 137
- 클라이언트로 푸시 백엔드, LDAP에 추가 147

- 클래스 공유 사용, JVM(Java Virtual Machine) 131
- 클래스 공유, JVM(Java Virtual Machine)에서 사용 131

- 키 데이터베이스, gskkyman을 사용하여 작성 222
- 키 링, RACF를 사용하여 작성 213
- 키 자원 정의 110
 - rsed.envvars 110
 - SYS1.PARMLIB(BPXPRMxx) 111

[타]

- 태스크 소유자 7
- 테스트 로깅, fekfivpc IVP 195
- 테스트 로깅, fekfivpi IVP 196
- 테이블, 로컬 호스트 230
- 테이블, 변환 230
- 통신, SSL 암호화 159
- 통신, SSL/TLS 암호화 33
- 통신, 내부 69
- 통신, 외부 68
- 통합 디버거 11
 - 암호화 통신 34

- 튜닝 고려사항 89
- 트랜잭션 덤프 패턴 변수 198
- 트랜잭션 보안 157

[파]

- 파이프라인 보안 157
- 파일 시스템 공간 사용량, z/OS UNIX 107
- 파일 시스템 속성, SETUID 202
- 파일 시스템, zFS 127
- 포트 선택, 제한 74
- 포트 예약, TCP/IP 69
- 포트, CARMA 및 TCP/IP 69
- 포트, TCP/IP 67
- 포트, 예약된 TCP/IP 206
- 포트, 지정으로 외부 통신 제한 24
- 프로그램 제어 권한 203
- 프로세스 개수 94
- 프로젝트, 호스트 기반 153
- 프로파일, 데이터 세트 정의 58
- 피드백 추적, 오류 201

[하]

- 한계, 시스템 209
- 할당 exec 사용 177
- 할당 exec, 사용 177
- 호스트 기반 프로젝트 153

호스트 연결 에뮬레이터 210
 호스트 이름 종속성 227
 호스트 이름, Developer for System z에 적용 231
 호스트 주소가 분석되지 않음, TCP/IP 분석기 lock.log 234
 호스트 테이블, 로컬 230
 확인 응답, 지연 70
 활성화 140
 힙 크기 한계, Java 101

A

ACEE, 관리 44
 ACK 지연 70
 ACK, 지연 70
 ADNJSAPU, 관리 유틸리티 159
 APF 권한 부여 FEK.SFEKAUTH 59
 APF, 권한 부여 204
 AQEZPCM 23
 ASCHPMxx MAX 115
 ASSIZEMAX 52
 audit.action, 사용자 엑시트 172
 audit.log 191

B

BPXPRMxx 122
 INADDRANYCOUNT 114
 MAXASSIZE 52, 112, 208
 MAXFILEPROC 112
 MAXMMAPAREA 112
 MAXPROCSYS 111, 209
 MAXPROCUSER 111, 209
 MAXSOCKETS 114
 MAXTHREADS 111
 MAXTHREADTASKS 111
 MAXUIDS 112, 210

C

CARMA 로깅 rsecomm.log 195
 CARMA 및 TCP/IP 포트 69
 CARMA 추적 200

CEE.SCEELPA SYS1.PARMLIB(LPALSTxx) 128
 CICS 관리자용 관리 유틸리티 제공된 기능 159
 CICS 자원 설치 로깅 157
 CICS 자원 정의(CRD) 서버, 애플리케이션 배치 관리자 155
 CICS 자원 정의(CRD) 편집기, 애플리케이션 배치 관리자 155
 CICS 자원 정의, 개발자 155
 CICS 자원 정의, 관리자 155
 CICS 트랜잭션 43
 CICS 트랜잭션 디버그 167
 CICSplex SM BAS(Business Application Services) 156
 CICSTS 고려사항 155
 CICSTS 보안 43
 CLASSPATH 183
 Client Gateway 액세스 방법, TSO/ISPF 사용 176
 COBOL 원격 검사 201
 Common Access Repository Manager 로깅 195
 CRD 저장소 43
 CRD 저장소 보안 157

D

Developer for System z 시작 태스크, 정의 51
 Developer for System z 이해 3
 Developer for System z, 이해 3
 Developer for System z, 컴포넌트 개요 그래픽 표시 4

F

fa.log 190
 FEJJCNFG 69, 122, 185
 CONSOLE_NAME 31
 MAX_THREADS 114
 FEJJCNFG, JES 작업 모니터 45
 FEKAPPL 23
 fekfivpc IVP 테스트 로깅 fekfivpc.log 195
 fekfivpc.log 191

fekfivpi IVP 테스트 로깅 fekfivpi.log 196
 fekfivpi.log 191
 fekfivpi.log, IVP 테스트 로깅 196
 fekfivps.log 191
 fekfivps.log, IVP 테스트 로깅 196
 FEKLOGS를 사용한 로그 및 설정 분석 190
 FEKLOGS, 사용한 로그 및 설정 분석 190
 FEKRACF, 보안 정의 48
 fekrivp 204
 ffsgget.log 190
 ffspout.log 190
 ffs.log 190

G

GATE, 트래싱 43
 gskkyman, 키 데이터베이스 작성 222

I

IEASYSxx 123
 MAXUSER 115, 210
 ISPF TSO/ISPF Client Gateway ISP.SISPLOAD 54
 ISPF 프로파일, 기존 사용 177
 ISPF, 여러 개의 할당 exec 사용 178
 ISPF.conf 파일, 다중 설정과 함께 사용 178
 ISPF.conf, 기본 사용자 정의 176
 ISP.SISPLOAD
 ISPF TSO/ISPF Client Gateway 54
 IVP 테스트 로깅 fekfivpi.log 196
 fekfivps.log 196
 IVTPRMxx
 ECSA MAX 115
 FIXED MAX 115

J

Java Xquickstart 옵션 130
 Java 덤프 197
 Java 애플리케이션으로서의 RSE 그래픽 표시 6
 Java 힙 크기 한계 101
 Java 힙 크기, 고정 130
 JAVA_DUMP_TDUMP_PATTERN 198
 JCL 요구사항, 시작 208

JES JMON

GEN_CONSOLE_NAME 32

JES 명령 보안, 정의 56

JES 보안 29

JES 작업 모니터 구성

GEN_CONSOLE_NAME 32

JES 작업 모니터 로깅 192

JES 작업 모니터 인증 23

JES 작업 모니터 추적 199

JES 작업 모니터(JMON) 4

JES 작업 모니터, FEJCNFG 45

JMON 57, 185

JVM(Java Virtual Machine) 간에 클래스 공유 130

JVM(Java Virtual Machine), 클래스 공유 130

JVM, 클래스 공유 130

K

keytool을 사용하여 키 저장소, 작성 225

keytool, 키 저장소 작성 225

L

LDAP 고려사항 70

LDAP 그룹 설정, 초기 148

LDAP 그룹, 개발자 추가 148

LDAP 서버 선택 145

LDAP 서버 위치 146

LDAP 스키마 144

LE(Language Environment) 런타임 라이브러리 128

LIMIT_COMMANDS 29

LIMIT_VIEW 32

lock.log 190

logon.action, 사용자 엑시트 173

LPALSTxx 128

M

MVS 덤프 197

N

netstat 206

O

OFF.REMOTECOPY.MVS 41

OMVS 세그먼트, 정의 51

OutOfMemoryError 210

P

PassTicket 사용 25

PassTicket, 사용 25

POE 확인 25, 39

POE(Port Of Entry) 확인 25, 39

PORTRANGE 206

pushtoclient.properties 149, 152

Q

quickstart, Java 옵션(-Xquickstart) 130

R

RACF

허용 60

RACF, 키 링 작성 213

RESTful 인터페이스 156

RESTful 인터페이스 대 웹 서비스 인터페이스 156

rmt_class_loader_cache.jar 190

RSE 대한 라이브러리, MVS 정의 54

RSE 대한 제어 라이브러리, MVS 정의 54

RSE 대한 MVS 프로그램 제어 라이브러리 정의 54

RSE 대한 MVS 프로그램 제어 라이브러리, 정의 54

RSE 대한 UNIX 프로그램 제어 파일, 정의 64

RSE 대한 z/OS UNIX 프로그램 제어 파일 정의 64

RSE 대한 z/OS UNIX 프로그램 제어 파일, 정의 64

RSE 디먼 68

RSE 디먼 로그 파일

audit.log 192

rsedaemon.log 192

rseserver.log 192

serverlogs.count 192

stderr.*.log 192

stdout.*.log 192

RSE 디먼 로깅 192

RSE 디먼 및 감사 로깅 26

RSE 디먼을 사용한 인증 38

RSE 디먼(RSED) 4

RSE 디먼, 인증 38

RSE 모니터링 116

RSE 사용자 로깅

ffsget.log 193

ffsput.log 193

ffs.log 193

lock.log 193

rmt_class_loader.cache.jar 193

rsecomm.log 193

stderr.log 193

stdout.log 193

.dstoreMemLogging 193

.dstoreTrace 193

RSE 서버 68

RSE 서버를 보안 z/OS UNIX로 정의 53

RSE 서버의 스레드 보안

PassTicket 25

RSE 설정, 기존 복제 215

RSE 스레드 풀 로그 파일

audit.log 192

rsedaemon.log 192

rseserver.log 192

serverlogs.count 192

stderr.*.log 192

stdout.*.log 192

RSE 추적 199

RSE , POE(Port Of Entry) 확인 정의 39

rsecomm.log 190

SCLM 개발자 툴킷 로깅 194

rsecomm.properties 200

rsedaemon.log 190, 191

rsed.envvars 109, 149, 152, 183

Dmaximum.clients 110

Dmaximum.threadpool.process 110

Dmaximum.threads 110

Dminimum.threadpool.process 110

DSTORE_LOG_DIRECTORY 195, 199

STEPLIB 33

Xms 110

Xmx 110

_CMDSERV_CONF_HOME 178

_RSE_JAVAOPTS 176, 197

_RSE_PORTRANGE 24

rsed.envvars, 업데이트하여 공존 사용 216

rseserver.log 190, 191
 RSE에 대한 애플리케이션 보호, 정의 56
 RSE에 대한 지원, PassTicket 정의 55
 RSE에 대한 PassTicket 지원 정의 55
 RSE에 대한 PassTicket 지원, 정의 55
 RSE에 대한 POE(Port Of Entry) 확인 정의 39
 RSE, MVS 프로그램 제어 라이브러리 정의 54
 RSE, PassTicket 지원 정의 55
 RSE, pushtoclient.properties 47
 RSE, rsed.envvars
 _RSE_JAVAOPTS 45
 RSE, ssl.properties 46
 RSE, z/OS UNIX 프로그램 제어 파일 정의 64
 RSE, 모니터링 116
 RSE, 보안 z/OS UNIX 서버로 정의 53
 RSE, 애플리케이션 보호 정의 56

S

SCLM 개발자 툴킷 54
 SCLM 개발자 툴킷 로깅
 rsecomm.log 194
 SCLM 개발자 툴킷(SCLMDT) 4
 SCLM 보안 44
 serverlogs.count 190
 SETUID 파일 시스템 속성 202
 SMP/E 설치, 스티키(Sticky) 비트 205
 SSL 암호화된 통신 43, 159
 SSL 호스트 구성 연결 테스트 218
 SSL 호스트 구성 연결, 테스트 218
 SSL(Secure Socket Layer), 사용한 통신 암호화 24
 SSL(Secure Socket Layer), 설정 211
 SSL을 사용한 암호화, 통신 24
 SSL을 사용한 통신 암호화 24
 SSL, 사용한 통신 암호화 24
 SSL, 설정 211
 SSL, 암호화 212
 ssl.properties, 새 RSE 디먼을 작성하여 SSL 활성화 217
 ssl.properties, 업데이트하여 SSL 활성화 216
 SSL/TLS 암호화된 통신 33
 stderr.log 190

stderr.*.log 190
 stdout.log 190
 stdout.*.log 190
 STEPLIB 사용, 방지 127
 STEPLIB, 사용 방지 127
 SYS1.PARMLIB(BPXPRMxx) 122
 MAXASSIZE 52, 208
 MAXPROCSYS 209
 MAXPROCUSER 209
 MAXUIDS 210
 SYS1.PARMLIB(BPXPRMxx), JVM(Java Virtual Machine) 132
 SYS1.PARMLIB(BPXPRMxx), 설정된 제한사항 208
 SYS1.PARMLIB(IEASYSxx) 123
 MAXUSER 210
 sysplex에서 동일 설정 182
 sysplex, 동일 설정 182

T

TCP/IP 동작, 기본값 대체 70
 TCP/IP 분석기, 호스트 주소가 분석되지 않음
 lock.log 234
 TCP/IP 포트 67
 TCP/IP 포트 예약 69
 TCP/IP 포트, 그래픽 표시 67
 TCP/IP 포트, 예약됨 206
 TCP/IP, Developer for System z에 적용 231
 TCP/IP, 분석기에 사용 가능한 로컬 정의 233
 TCP/IP, 설정 227
 TLS를 사용한 암호화, 통신 24
 TLS를 사용한 통신 암호화 24
 TLS, 사용한 통신 암호화 24
 TLS, 암호화 212
 TSO 명령 서비스 4, 175
 TSO 액세스 방법 175
 TSO 환경 사용자 정의 175
 TSO 환경, 사용자 정의 175
 TSO/ISPF Client Gateway 액세스 방법, 사용 176
 TSO/ISPF, 기존 ISPF 프로파일 사용 177
 TSO/ISPF, 다중 설정과 함께 사용 178
 TSO/ISPF, 사용자 정의 - ISPF.conf, 176
 TSO/ISPF, 여러 개의 할당 exec 사용 178
 TSO/ISPF, 할당 exec 사용 177

U

UNIX 덤프 위치 199
 UNIX 서버, RSE 정의 53
 UNIX 환경, 사용되는 검색 순서 229

V

VIPA, 분산 동적 72

W

WLM 고려사항 xiv, 79
 WLM 분류 규칙 80

X

Xquickstart, Java 옵션 130
 X.509 인증 22
 X.509 인증서를 사용한 클라이언트 인증 35
 X.509 인증서, 클라이언트 인증 35
 x.509 인증, 설정 211
 X.509, 클라이언트 인증 지원 추가 221

Z

zFS 파일 시스템, 사용 127
 z/OS UNIX REXX exec 171
 z/OS UNIX 권한 비트 202
 z/OS UNIX 덤프 위치 199
 z/OS UNIX 디렉토리 구조
 그래픽 표시 16
 z/OS UNIX 명령, 유용
 chgrp 19
 chmod 19
 chown 19
 ls 19
 z/OS UNIX 모니터링 117
 z/OS UNIX 서버, RSE 정의 53
 z/OS UNIX 셸 스크립트 171
 z/OS UNIX 파일 시스템 공간 사용량 107
 z/OS UNIX 파일 시스템 모니터링 120
 z/OS UNIX 파일 시스템, 모니터링 120
 z/OS UNIX 환경, 사용되는 검색 순서 229
 z/OS UNIX, 모니터링 117

[특수 문자]

.dstoreMemLogging 190

.dstoreTrace 190

/var/rdz/pushtoclient/*install 149, 152

_RSE_PORTRANGE 24

고객 의견서

IBM Rational Developer for System z

버전 9.0.1

호스트 구성 참조 안내서

SA30-4501-05

성명

주소

회사 또는 단체명

전화번호

고객 의견서
SA30-4501-05



선을 따라
자르거나
접으십시오

접어서 붙이십시오

스테이플러를 사용하지 마시오

접어서 붙이십시오

우 표
붙이는
곳

IBM
Corporation
Building 501
P.O Box 12195
Research Triangle Park, NC
USA 27709-2195

접어서 붙이십시오

스테이플러를 사용하지 마시오

접어서 붙이십시오

SA30-4501-05

선을 따라
자르거나
접으십시오



Printed in Korea

SA30-4501-05

