

**Rational** IBM Rational Developer for System z  
version 7.6.1

*Guide de configuration du système  
hôte*





**Rational** IBM Rational Developer for System z  
version 7.6.1

*Guide de configuration du système  
hôte*



**Important**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques relatives à la documentation pour IBM Rational Developer for System z», à la page 359.

**Remarque**

Certaines illustrations de ce manuel ne sont pas disponibles en français à la date d'édition.

**Quatrième édition - mai 2010**

Réf. US : SC23-7658-04

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France  
17 avenue de l'Europe  
92275 Bois-Colombes Cedex*

© Copyright IBM France 2010. Tous droits réservés.

© Copyright IBM Corporation 2005, 2010.

# Table des matières

<b>Figures . . . . .</b>	<b>ix</b>
--------------------------	-----------

<b>Tableaux . . . . .</b>	<b>xi</b>
---------------------------	-----------

<b>A propos de ce manuel . . . . .</b>	<b>xiii</b>
--	-------------

A qui s'adresse ce guide . . . . .	xiii
Récapitulatif des changements . . . . .	xiii
Description du contenu du document . . . . .	xiv
Planification . . . . .	xiv
Personnalisation de base . . . . .	xiv
(Facultatif) Common Access Repository Manager (CARMA). . . . .	xiv
(Facultatif) Gestionnaire de déploiement d'application . . . . .	xv
(Facultatif) SCLM Developer Toolkit . . . . .	xv
(Facultatif) Tâches de personnalisation alternatives . . . . .	xv
Vérification de l'installation . . . . .	xvi
Commandes de l'opérateur . . . . .	xvi
Identification des incidents liés à la configuration . . . . .	xvi
Remarques relatives à la sécurité . . . . .	xvi
Compréhension de Developer for System z . . . . .	xvi
Remarques à propos de WLM . . . . .	xvi
Remarques relatives à l'optimisation . . . . .	xvi
Remarques relatives aux performances . . . . .	xvii
Remarques à propos de CICSTS . . . . .	xvii
Personnalisation de l'environnement TSO . . . . .	xvii
Exécution de plusieurs instances . . . . .	xvii
Guide de migration . . . . .	xvii
Configuration de l'authentification SSL et X.509 . . . . .	xvii
Configuration de TCP/IP . . . . .	xviii
Configuration d'INETD. . . . .	xviii
Configuration APPC. . . . .	xviii
Conditions requises . . . . .	xviii

## Partie 1. Personnalisation de Developer for System z . . . . . 1

### Chapitre 1. Planification . . . . . 3

Remarques relatives à la migration . . . . .	3
Remarques relatives à la planification . . . . .	3
Présentation du produit . . . . .	3
Compétences requises . . . . .	4
Temps nécessaire . . . . .	4
Remarques relatives à la pré-installation . . . . .	4
Choix de configuration . . . . .	5
Éléments prérequis pour les produits . . . . .	5
Ressources requises . . . . .	5
Préparation de la configuration . . . . .	9
Gestion de la charge de travail . . . . .	9
Utilisation des ressources et limites du système . . . . .	9
Configuration nécessaire des produits requis . . . . .	9
Remarques relatives à l'ID utilisateur . . . . .	9

Remarques relatives au serveur . . . . .	10
Méthode de configuration . . . . .	11
Remarques relatives au prédéploiement . . . . .	12
Liste de contrôle du client . . . . .	13

### Chapitre 2. Personnalisation de base 15

Configuration requise et liste de contrôle . . . . .	15
Configuration personnalisée . . . . .	15
Modifications de PARMLIB . . . . .	16
Définition des limites z/OS UNIX dans BPXPRMxx . . . . .	16
Ajout de tâches démarrées à COMMNDxx . . . . .	18
Définitions LPA dans LPALSTxx . . . . .	18
Autorisations APF dans PROGxx . . . . .	18
Définitions LINKLIST dans PROGxx . . . . .	19
Définitions LINKLIST et LPA prérequis . . . . .	20
Définitions LINKLIST pour les autres produits . . . . .	21
Modifications de PROCLIB . . . . .	21
Moniteur de travaux JES . . . . .	21
Démon RSE . . . . .	22
Démon lock . . . . .	23
Limitations JCL pour la variable PARM . . . . .	24
Procédures de construction à distance ELAXF* . . . . .	25
Définitions de sécurité . . . . .	26
FEJJCNF, fichier de configuration du moniteur de travaux JES . . . . .	27
rsed.envvars - Fichier de configuration RSE . . . . .	32
Définition de PORTRANGE disponible pour un serveur RSE . . . . .	41
Définition de paramètres de démarrage Java supplémentaires avec _RSE_JAVAOPTS . . . . .	42
Définition de paramètres de démarrage Java supplémentaires avec _RSE_CMDSERV_OPTS . . . . .	47
ISPF.conf, fichier de configuration de la passerelle client TSO/ISPF d'ISPF . . . . .	47
Composants facultatifs . . . . .	49
Vérification de l'installation . . . . .	49

### Chapitre 3. (Facultatif) Common Access Repository Manager (CARMA) . 51

Configuration requise et liste de contrôle . . . . .	51
Composants CARMA . . . . .	52
Notes de migration VSAM CARMA . . . . .	53
interface RSE avec CARMA . . . . .	53
Démarrage du serveur CARMA à l'aide de la soumission par lots . . . . .	56
Ajustement de CRASRV.properties. . . . .	56
Ajustement de CRASUBMT . . . . .	56
(Facultatif) Démarrage du serveur CARMA alternatif à l'aide de CRASTART . . . . .	57
Ajustement de CRASRV.properties. . . . .	58
Ajustement de crastart.conf . . . . .	58
(Facultatif) Démarrage du serveur CARMA alternatif à l'aide de la passerelle client TSO/ISPF . . . . .	60
Ajustement de CRASRV.properties. . . . .	60

Ajustement d'ISPF.conf . . . . .	61
(Facultatif) Activation des modèles de RAM (Repository Access Manager) . . . . .	62
Activation du RAM PDS . . . . .	63
Activation du RAM SCLM . . . . .	63
Activation du gestionnaire RAM du squelette . . . . .	63
(Facultatif) Activation du RAM de CA Endevor à l'aide de SCM . . . . .	64
Configuration requise et liste de contrôle . . . . .	64
Définir CA Endevor SCM RAM . . . . .	65
Démarrage de CA Endevor SCM RAM à l'aide de la soumission par lots . . . . .	65
Démarrage de CA Endevor SCM RAM à l'aide de CRASTART . . . . .	68
(Facultatif) Personnaliser CRANDVRA . . . . .	69
(Facultatif) Personnaliser CA Endevor SCM RAM . . . . .	70
(Facultatif) Prise en charge de plusieurs RAM . . . . .	70
Exemple . . . . .	71
(Facultatif) IRXJCL versus CRAXJCL . . . . .	72
Création de CRAXJCL . . . . .	72

## Chapitre 4. (Facultatif) Gestionnaire de déploiement d'application . . . . . 75

Configuration requise et liste de contrôle . . . . .	75
Référentiel de CRD . . . . .	76
Utilitaire d'administration CICS . . . . .	77
RESTful par opposition à Web Service . . . . .	77
Serveur CRD utilisant l'interface RESTful . . . . .	77
Région de connexion CICS primaire . . . . .	77
Régions de connexion CICS secondaires . . . . .	78
(Facultatif) Personnalisation des ID transaction du serveur CRD . . . . .	78
Serveur CRD utilisant l'interface Web Service . . . . .	79
Gestionnaire de message de pipeline . . . . .	79
région de connexion CICS primaire . . . . .	80
régions de connexion CICS secondaires . . . . .	81
(Facultatif) Référentiel de manifestes . . . . .	81

## Chapitre 5. (Facultatif) SCLM Developer Toolkit . . . . . 83

Configuration requise et liste de contrôle . . . . .	83
Conditions préalables . . . . .	84
Mises à jour d'ISPF.conf pour SCLMDT . . . . .	84
Mises à jour de rsed.envvars pour SCLMDT . . . . .	85
(Facultatif) Conversion de nom long/court . . . . .	86
Création du fichier LSTRANS.FILE, VSAM de conversion des noms longs/courts. . . . .	86
Mises à jour de rsed.envvars pour la conversion de noms longs/courts . . . . .	88
(Facultatif) Installation et personnalisation d'Ant . . . . .	88
Mises à jour SCLM pour SCLMDT . . . . .	89
Suppression des fichiers antérieurs du répertoire WORKAREA . . . . .	90

## Chapitre 6. (Facultatif) Tâches de personnalisation alternatives . . . . . 91

(Facultatif) Procédure mémorisée DB2 . . . . .	91
Modifications de Workload Manager (WLM) . . . . .	91
Modifications de PROCLIB . . . . .	92
Modifications de DB2 . . . . .	92

(Facultatif) Support EST (Enterprise Service Tools) . . . . .	93
(Facultatif) Support de langue bidirectionnelle CICS . . . . .	94
(Facultatif) Messages d'erreur de diagnostic IRZ . . . . .	95
(Facultatif) Chiffrement SSL RSE . . . . .	95
(Facultatif) Fonction de trace RSE . . . . .	98
(Facultatif) Groupes de propriétés basés sur l'hôte . . . . .	100
(Facultatif) Projets basés sur l'hôte . . . . .	101
(Facultatif) Intégration de File Manager. . . . .	102
(Facultatif) Caractères non éditables . . . . .	103
(Facultatif) Utilisation de REXEC (ou SSH) . . . . .	104
Actions à distance (basées sur l'hôte) dans les sous-projets z/OS UNIX. . . . .	105
Méthode de connexion RSE alternative . . . . .	105
Configuration de REXEC (ou SSH) . . . . .	105
(Facultatif) Transaction APPC pour le service Commandes TSO . . . . .	106
Préparation . . . . .	106
Implémentation . . . . .	108
Remarques relatives à l'utilisation d'APPC. . . . .	109
(Facultatif) Nettoyage du répertoire WORKAREA . . . . .	110

## Chapitre 7. Vérification de l'installation 111

Vérification des tâches démarrées. . . . .	111
Moniteur de travaux JES (JMON). . . . .	111
LOCKD, démon Lock. . . . .	111
Démon RSED, RSE . . . . .	111
Vérification des services . . . . .	113
Initialisation de programme de vérification de l'installation . . . . .	114
Disponibilité des ports . . . . .	115
Configuration TCP/IP . . . . .	116
Connexion du démon RSE . . . . .	117
Connexion du moniteur de travaux JES. . . . .	117
Connexion du démon lock . . . . .	118
Connexion de la passerelle client TSO/ISPF d'ISPF . . . . .	118
(Facultatif) Connexion au service Commandes TSO à l'aide d'APPC . . . . .	120
(Facultatif) Connexion à SCLMDT . . . . .	120
(Facultatif) Connexion à REXEC . . . . .	122
(Facultatif) Script de shell REXEC/SSH. . . . .	123

## Partie 2. Informations relatives à Developer for System z . . . . . 125

### Chapitre 8. Commandes de l'opérateur 127

Start (S) . . . . .	127
Moniteur de travaux JES . . . . .	127
Démon RSE . . . . .	128
Démon lock . . . . .	128
Modify (F) . . . . .	129
Moniteur de travaux JES . . . . .	129
Démon RSE . . . . .	130
Démon lock . . . . .	133
Stop (P) . . . . .	133
Messages de la console . . . . .	134
Moniteur de travaux JES . . . . .	134
Démon RSE, serveur de pool d'unités d'exécution RSE et démon lock . . . . .	134
Comment lire un diagramme de syntaxe . . . . .	136

Symboles . . . . .	136
Opérandes . . . . .	136
Exemple de syntaxe . . . . .	137
Caractères non alphanumériques et espaces . . . . .	137
Sélection de plusieurs opérandes . . . . .	137
Diagramme sur plusieurs lignes . . . . .	137
Fragments de syntaxe . . . . .	137

## Chapitre 9. Identification des incidents liés à la configuration . . . . . 139

Journal et analyse de configuration à l'aide de FEKLOGS . . . . .	139
Fichiers journaux . . . . .	140
Journalisation du moniteur de travaux JES . . . . .	141
Consignation du démon lock . . . . .	141
Journalisation du démon RSE et du pool d'unités d'exécution . . . . .	141
Journalisation pour l'utilisateur RSE . . . . .	142
Journal d'intégration de Fault Analyzer . . . . .	143
Journal d'intégration de File Manager . . . . .	143
Consignation de SCLM Developer Toolkit . . . . .	144
Journalisation pour le gestionnaire CARMA . . . . .	144
Journalisation de la transaction APPC (service Commandes TSO) . . . . .	144
Consignation des tests du programme de vérification de l'installation (IVP) fekfivpi . . . . .	145
Consignation des tests du programme de vérification de l'installation (IVP) fekfivps . . . . .	145
Fichiers de vidage . . . . .	145
Fichiers de vidage MVS . . . . .	145
Fichiers de vidage Java . . . . .	146
Emplacements des fichiers de vidage z/OS UNIX . . . . .	147
Fonction de trace . . . . .	148
Fonction de trace du moniteur de travaux JES . . . . .	148
Fonction de trace RSE . . . . .	148
Traçage du démon lock . . . . .	149
Fonction de trace CARMA . . . . .	149
Traçage de suivi des erreurs . . . . .	149
Données de droits z/OS UNIX . . . . .	150
Attribut du système de fichiers SETUID . . . . .	150
Autorisation de contrôle de programmes . . . . .	151
Autorisation APF . . . . .	152
Données de rappel . . . . .	153
Ports TCP/IP réservés . . . . .	153
Taille d'espace adresse . . . . .	155
Exigences liées au langage JCL de démarrage . . . . .	155
Limitations définies dans SYS1.PARMLIB(BPXPRMxx) . . . . .	155
Limitations stockées dans le profil de sécurité . . . . .	155
Limitations forcées par les sorties du système . . . . .	155
Limitations pour adressage 64 bits . . . . .	156
Transaction APPC et service Commandes TSO . . . . .	156
Informations diverses . . . . .	158
Limites du système . . . . .	158
Incidents connus relatifs aux éléments requis . . . . .	158
Emulateur de connexion à l'hôte . . . . .	159

## Chapitre 10. Remarques relatives à la sécurité . . . . . 161

Méthodes d'authentification . . . . .	161
ID utilisateur et mot de passe . . . . .	162
ID utilisateur et mot de passe utilisable une seule fois . . . . .	162
Certificat X.509 . . . . .	162
Authentification du moniteur de travaux JES . . . . .	162
Sécurité des connexions . . . . .	163
Limite des communications externes à des ports spécifiques . . . . .	163
Chiffrement des communications à l'aide de SSL . . . . .	163
Vérification du port d'entrée . . . . .	164
Ports TCP/IP . . . . .	164
Communications externes . . . . .	164
Communication interne . . . . .	165
CARMA et ports TCP/IP . . . . .	166
Utilisation de PassTickets . . . . .	166
Consignation dans le journal d'audit . . . . .	167
Contrôle d'audit . . . . .	167
Données d'audit . . . . .	168
Sécurité JES . . . . .	168
Actions sur les travaux - Limitations sur les cibles . . . . .	168
Actions sur les travaux - Limitations liées à l'exécution . . . . .	170
Accès aux fichiers spoule . . . . .	171
Communication chiffrée à l'aide du protocole SSL . . . . .	172
Authentification du client à l'aide de certificats X.509 . . . . .	173
Validation de l'autorité de certification (CA) . . . . .	173
(Facultatif) Interrogation d'une liste de révocation de certificat (CRL) . . . . .	174
Authentification par votre logiciel de sécurité . . . . .	174
Authentification effectuée par le démon RSE . . . . .	175
Vérification du port d'entrée (POE) . . . . .	176
Sécurité CICSTS . . . . .	177
Référentiel de CRD . . . . .	177
Transactions CICS . . . . .	177
Communication chiffrée à l'aide du protocole SSL . . . . .	177
Sécurité SCLM . . . . .	177
Fichiers de configuration Developer for System z . . . . .	178
Moniteur de travaux JES - FEJJCNFG . . . . .	178
RSE - rsed.envvars . . . . .	178
RSE - ssl.properties . . . . .	179
Définitions de sécurité . . . . .	179
Configuration requise et liste de contrôle . . . . .	180
Activation des paramètres et des classes de sécurité . . . . .	181
Définition d'un segment OMVS pour les utilisateurs de Developer for System z . . . . .	182
Définition des profils de fichier . . . . .	182
Définition des tâches démarrées Developer for System z . . . . .	185
Définition de la sécurité des commandes JES . . . . .	186
Définition de RSE comme serveur z/OS UNIX sécurisé . . . . .	188
Définition de bibliothèques contrôlées par programme MVS pour RSE . . . . .	188
Définition de la protection d'application pour RSE . . . . .	189



Définition du support de mots de passe	
PassTicket pour RSE	189
Définition de fichiers contrôlés par programme	
z/OS UNIX pour RSE	190
Vérification des paramètres de sécurité	191

## Chapitre 11. Compréhension de Developer for System z . . . . . 193

Présentation du composant	193
RSE comme application Java	195
Propriétaires de tâches	196
Flux de connexion	198
Démon lock	200
Libération d'un verrou	201
Structure de répertoire z/OS UNIX	202
Droits de mise à jour des administrateurs non système	203

## Chapitre 12. Remarques à propos de WLM . . . . . 205

Classification des charges de travail	205
Règles de classification	206
Définition des objectifs	207
Remarques relatives à la sélection des objectifs	208
STC	209
OMVS	210
JES	211
ASCH	212
CICS	212

## Chapitre 13. Remarques relatives à l'optimisation . . . . . 215

Utilisation des ressources	215
Présentation	216
Nombre d'espaces adresses	217
Nombre de processus	220
Nombre d'unités d'exécution	223
Utilisation de l'espace de stockage	226
Limite de la taille de pile Java	227
Limite de la taille d'espace adresse	227
Instructions relatives à l'évaluation de la taille	228
Exemple d'analyse de l'utilisation de l'espace de stockage	228
Utilisation de l'espace du système de fichiers z/OS UNIX	232
Définitions de ressources essentielles	235
/etc/rdz/rsed.envvars	235
SYS1.PARMLIB(BPXPRMxx)	236
Définitions de ressource différentes	239
Carte EXEC dans le JCL de serveur	239
FEK.#CUST.PARMLIB(FEJJCNF)	239
SYS1.PARMLIB(IEASYSxx)	239
SYS1.PARMLIB(IVTPRMxx)	239
SYS1.PARMLIB(ASCHPMxx)	240
Contrôle	240
Contrôle de RSE	240
Contrôle de z/OS UNIX	241
Contrôle du réseau	243
Contrôle des systèmes de fichiers z/OS UNIX	243
Exemple de configuration	244

Nombre de pools d'unités d'exécution	244
Détermination des limites minimales	244
Définition des limites	245
Utilisation des ressources du moniteur	246

## Chapitre 14. Remarques relatives aux performances . . . . . 249

Utilisation du système de fichiers zFS	249
Eviter l'emploi de STEPLIB	249
Amélioration de l'accès aux bibliothèques du système	250
Bibliothèque d'exécution Language Environment (LE)	250
Développement d'applications	250
Amélioration des performances du contrôle d'autorisations d'accès	251
Gestion de la charge de travail	251
Taille de pile Java fixe	251
Option Java -Xquickstart	252
Partage de classes entre JVM	252
Activation du partage de classe	253
Limites de la taille de la mémoire cache	253
Sécurité de la mémoire cache	253
SYS1.PARMLIB(BPXPRMxx)	254
Espace disque	254
Utilitaires de gestion de la mémoire cache	254

## Chapitre 15. Remarques à propos de CICSTS . . . . . 257

RESTful par opposition à Web Service	258
Régions de connexion primaires versus régions de connexion non primaires	258
Consignation des messages d'installation des ressources CICS	259
gestionnaire de déploiement d'application, sécurité	259
Sécurité du référentiel CRD	259
pipeline,sécurité	259
Sécurité de transaction	259
Communication chiffrée à l'aide du protocole SSL	260
Protection des ressources	261
Utilitaire d'administration	261
Notes de migration de l'utilitaire d'administration	265
Messages de l'utilitaire d'administration	266

## Chapitre 16. Personnalisation de l'environnement TSO . . . . . 269

Service Commandes TSO	269
Méthodes d'accès	269
Utilisation de la méthode d'accès par passerelle client TSO/ISPF	270
Personnalisation de base – ISPF.conf	270
Avancé – Utilisation des profils ISPF existants	270
Avancé – Utilisation d'une commande exec d'allocation	271
Avancé – Utilisation de plusieurs commandes exec d'allocation	271
Avancé – Fichiers ISPF.conf multiples avec configurations Developer for System z multiples	272



Utilisation de la méthode d'accès APPC . . . . .	272
Personnalisation de base – JCL de transaction APPC . . . . .	272
Avancé – Utilisation des profils ISPF existants	273
Avancé – Utilisation d'une commande exec d'allocation . . . . .	273
Avancé – Transactions APPC multiples avec configurations Developer for System z multiples	274

## Chapitre 17. Exécution de plusieurs instances . . . . . 277

Configuration identique par sysplex . . . . .	277
Niveaux de logiciels identiques, fichiers de configuration différents . . . . .	278
Dans tous les autres cas . . . . .	279

## Chapitre 18. Guide de migration . . . . . 283

Remarques relatives à la migration . . . . .	283
Sauvegarde des fichiers déjà configurés. . . . .	283
Notes de migration relatives à la Version 7.6.1 . . . . .	285
Migration de la version 7.5 vers la version 7.6 . . . . .	286
IBM Rational Developer for System z, FMID HHOP760 . . . . .	286
Fichiers configurables . . . . .	288
Migration de la version 7.1 vers la version 7.5 . . . . .	292
IBM Rational Developer for System z, FMID HHOP750 . . . . .	292
Fichiers configurables . . . . .	294
Migration de la version 7.0 à la version 7.1 . . . . .	296
IBM Rational Developer for System z, FMID HHOP710 . . . . .	296
IBM Common Access Repository Manager (CARMA), FMID HCMA710 . . . . .	296
Fichiers configurables . . . . .	297

## Annexe A. Configuration de l'authentification SSL et X.509 . . . . . 299

Choix de l'emplacement de stockage des clés privées et des certificats . . . . .	300
Création d'un fichier de clés avec RACF . . . . .	301
(Facultatif) Utilisation d'un certificat signé. . . . .	302
Clonage de la configuration RSE existante. . . . .	303
Mise à jour du fichier rsed.envvars pour assurer la coexistence . . . . .	303
Mise à jour du fichier ssl.properties pour activer SSL. . . . .	303
Activation de SSL en créant un démon RSE . . . . .	304
Test de la connexion . . . . .	305
(Facultatif) Ajout du support d'authentification du client via des certificats X.509 . . . . .	308
(Facultatif) Création d'une base de données de clés avec gskkyman. . . . .	308
(Facultatif) Création d'un magasin de clés avec keytool . . . . .	311

## Annexe B. Configuration de TCP/IP 315

Dépendance au nom d'hôte. . . . .	315
Présentation des programmes de résolution . . . . .	316
Présentation des ordres de recherche d'informations de configuration . . . . .	316

Ordres de recherche dans l'environnement z/OS UNIX . . . . .	317
Fichiers de configuration du programme de résolution de base. . . . .	317
Tables de conversion . . . . .	318
Tables de système hôte local . . . . .	318
Application de ces informations de configuration à Developer for System z . . . . .	319
Résolution erronée de l'adresse hôte. . . . .	321

## Annexe C. Configuration d'INETD. . . . . 323

inetd.conf . . . . .	323
ETC.SERVICES . . . . .	324
Ordre de recherche utilisé dans l'environnement z/OS UNIX . . . . .	325
Ordre de recherche dans l'environnement MVS natif . . . . .	325
Définitions de ports PROFILE.TCPIP . . . . .	326
/etc/inetd.pid . . . . .	327
Démarrage . . . . .	327
/etc/rc . . . . .	327
/etc/inittab . . . . .	327
BPXBATCH . . . . .	328
Session de Shell . . . . .	328
Sécurité . . . . .	329
Exigences de Developer for System z . . . . .	330
INETD . . . . .	330
REXEC (ou SSH) . . . . .	330

## Annexe D. Configuration APPC. . . . . 331

Méthode d'accès VSAM . . . . .	331
VTAM. . . . .	332
SYS1.PARMLIB(APPCPMxx) . . . . .	333
SYS1.PARMLIB(ASCHPMxx) . . . . .	334
Activation des modifications d'APPC . . . . .	335
Définition de la transaction du service Commandes TSO . . . . .	335
(Facultatif) Options de configuration alternatives	336
Nom de transaction alternatif . . . . .	336
Unités logiques multiples . . . . .	336
Sécurité des unités logiques . . . . .	336

## Annexe E. Conditions requises. . . . . 339

Éléments prérequis de l'hôte z/OS . . . . .	339
z/OS . . . . .	339
SMP/E . . . . .	341
SDK for z/OS Java 2 Technology Edition . . . . .	341
Corequis pour l'hôte z/OS . . . . .	341
z/OS . . . . .	342
Compilateur COBOL . . . . .	344
Compilateur PL/I . . . . .	344
Debug Tool for z/OS. . . . .	345
CICS Transaction Server. . . . .	345
IMS . . . . .	346
DB2 for z/OS . . . . .	347
Rational Team Concert for System z. . . . .	347
Gestionnaire de fichiers . . . . .	348
Fault Analyzer . . . . .	348
REXX . . . . .	348
Ported Tools. . . . .	349

Ant. . . . .	349
Endevor . . . . .	349
<b>Bibliographie . . . . .</b>	<b>351</b>
Publications référencées . . . . .	351
Publications d'information . . . . .	353
<b>Glossaire . . . . .</b>	<b>355</b>

<b>Remarques relatives à la documentation pour IBM Rational Developer for System z . . . . .</b>	<b>359</b>
Licence de copyright . . . . .	360
Marques . . . . .	361
<b>Index . . . . .</b>	<b>363</b>

## Figures

1.	JMON - Tâche démarrée du moniteur de travaux JES . . . . .	22	31.	uchars.settings - fichier de configuration des caractères non éditables . . . . .	104
2.	RSED - Tâche démarrée du serveur RSE . . . . .	23	32.	Panneaux REXX pour APPC ISPF. . . . .	107
3.	LOCKD - Tâche démarrée du démon lock . . . . .	23	33.	Commande de l'opérateur START JMON . . . . .	127
4.	RSED - Démarrage alternatif du démon RSE . . . . .	24	34.	Commande de l'opérateur START RSED . . . . .	128
5.	rsed.stdin.sh - Démarrage alternatif du démon RSE . . . . .	24	35.	Commande de l'opérateur START LOCKD . . . . .	128
6.	FEJCNFG, fichier de configuration du moniteur de travaux JES . . . . .	28	36.	Commande de l'opérateur MODIFY JMON . . . . .	129
7.	rsed.envvars - Fichier de configuration RSE . . . . .	34	37.	Commande de l'opérateur MODIFY RSED . . . . .	130
8.	(continued). . . . .	35	38.	Commande de l'opérateur MODIFY LOCKD . . . . .	133
9.	ISPF.conf - Fichier de configuration ISPF . . . . .	48	39.	Commande de l'opérateur STOP . . . . .	133
10.	CRASRV.properties - Fichier de configuration de CARMA . . . . .	54	40.	Ports TCP/IP . . . . .	164
11.	CRASRV.properties - Démarrage de CARMA à l'aide de la soumission par lots . . . . .	56	41.	Présentation du composant . . . . .	193
12.	CRASUBMT - CARMA startup using batch submit . . . . .	57	42.	RSE comme application Java . . . . .	195
13.	CRASRV.properties - *CRASTART alternative CARMA startup . . . . .	58	43.	Propriétaires de tâches . . . . .	197
14.	crastart.conf - *CRASTART alternative CARMA startup . . . . .	59	44.	flux de connexion . . . . .	198
15.	CRASRV.properties - *ISPF alternative CARMA startup . . . . .	61	45.	Flux du démon lock . . . . .	200
16.	ISPF.conf - *ISPF alternative CARMA startup . . . . .	62	46.	structure de répertoire z/OS UNIX . . . . .	202
17.	Figure x1. CRASRV.properties - Démarrage de CA Endevor SCM RAM a l'aide de la soumission par lots . . . . .	66	47.	Classification WLM . . . . .	205
18.	Figure x2. CRASUBMT - Démarrage de CA Endevor SCM RAM a l'aide de la soumission par lots . . . . .	67	48.	Nombre maximal d'espaces adresses . . . . .	219
19.	Figure x3. CRASRV.properties - Démarrage de CA Endevor SCM RAM à l'aide de CRASTART . . . . .	68	49.	Nombre d'espaces adresses par client . . . . .	220
20.	crastart.conf - Démarrage de CA Endevor SCM RAM à l'aide de CRASTART . . . . .	69	50.	Nombre maximal de processus. . . . .	222
21.	Mises à jour d'ISPF.conf pour SCLMDT . . . . .	85	51.	Nombre de processus par client . . . . .	222
22.	Mises à jour de rsed.envvars pour SCLMDT . . . . .	86	52.	Nombre maximal d'unités d'exécution du pool d'unités d'exécution RSE . . . . .	225
23.	FLM02LST - JCL de configuration de la conversion de nom long/abrégé . . . . .	87	53.	Nombre maximal d'unités d'exécution du moniteur de travaux JES . . . . .	225
24.	ELAXMSAM - Tâche de procédure mémorisée DB2 . . . . .	92	54.	Utilisation des ressources avec 5 connexions . . . . .	229
25.	ELAXMJCL - Définition d'une procédure mémorisée DB2 . . . . .	93	55.	Utilisation des ressources avec 5 connexions (suite) . . . . .	230
26.	ssl.properties - Fichier de configuration RSE . . . . .	97	56.	Utilisation des ressources lors de l'édition d'un membre PDS . . . . .	231
27.	rsecomm.properties - Fichier de configuration de consignment . . . . .	99	57.	Utilisation de l'espace du système de fichiers z/OS UNIX . . . . .	233
28.	propertiescfg.properties - Fichier de configuration des groupes de propriétés basés sur l'hôte . . . . .	100	58.	Utilisation des ressources de la configuration modèle. . . . .	247
29.	projectcfg.properties - Fichier de configuration de projets basé sur l'hôte . . . . .	102	59.	Utilitaire d'administration ADNJSPAU - CICSTS . . . . .	263
30.	FMIEXT.properties - Fichier de configuration de File Manager . . . . .	103	60.	FEKAPPCC - Création d'une seconde transaction APPC . . . . .	275
			61.	RSEDSSL - Travail de l'utilisateur du serveur RSE pour SSL . . . . .	304
			62.	Boîte de dialogue Importation du certificat hôte. . . . .	305
			63.	Boîte de dialogue Préférences - SSL . . . . .	307
			64.	Langage JCL de démarrage d'INETD . . . . .	328
			65.	Langage JCL pour la création des méthodes VSAM d'APPC . . . . .	332
			66.	SYS1.SAMPLIB(ATBAPPL) . . . . .	333
			67.	SYS1.PARMLIB(APPCPMxx) . . . . .	333
			68.	SYS1.PARMLIB(ASCHPMxx) . . . . .	335



## Tableaux

1. Ressources requises . . . . .	6	26. Variables de configuration de la sécurité	180
2. Ressources optionnelles . . . . .	6	27. Commandes opérateur du moniteur de travaux JES2 . . . . .	187
3. Administrateurs requis pour les tâches requises	7	28. Commandes opérateur du moniteur de travaux JES3 . . . . .	187
4. Administrateurs nécessaires pour les tâches optionnelles . . . . .	8	29. Sous-système du point d'entrée WLM	206
5. Liste de contrôle du client - Composants obligatoires . . . . .	13	30. Qualificateurs de travaux WLM . . . . .	207
6. Liste de contrôle du client - Composants facultatifs . . . . .	13	31. Charges de travail WLM . . . . .	208
7. Modèles de procédure ELAXF* . . . . .	25	32. Charges de travail et STC WLM . . . . .	209
8. Liste de contrôle des qualificatifs de haut niveau ELAXF* . . . . .	26	33. Charges de travail - OMVS WLM . . . . .	210
9. Matrice des droits d'accès des commandes LIMIT_COMMANDS . . . . .	30	34. Charge de travail - JES WLM . . . . .	211
10. Variables du fichier crastart.conf . . . . .	59	35. Charges de travail - ASCH WLM . . . . .	212
11. ID transaction du serveur CRD par défaut	78	36. Charges de travail - CICS WLM . . . . .	212
12. ID transaction du serveur CRD par défaut	79	37. Utilisation des ressources communes	216
13. Liste de contrôle de l'administrateur SCLM	90	38. Utilisation des ressources requises propres à l'utilisateur . . . . .	216
14. Mécanismes de stockage des certificats SSL	96	39. Utilisation des ressources propres à l'utilisateur . . . . .	217
15. Types de fichier de clés valides . . . . .	98	40. Nombre d'espaces adresses . . . . .	217
16. Liste de contrôle de transaction APPC	107	41. Limites d'espace adresse . . . . .	220
17. Programmes de vérification de l'installation pour les services . . . . .	114	42. Nombre de processus . . . . .	220
18. État d'erreur du pool d'unités d'exécution	131	43. Limites de processus . . . . .	223
19. Messages de console RSE . . . . .	134	44. Nombre d'unités d'exécution . . . . .	224
20. Variables JAVA_DUMP_TDUMP_PATTERN	146	45. Limites d'unités d'exécution. . . . .	226
21. Commandes de la console du moniteur de travaux JES . . . . .	169	46. Répertoires de sortie de journal . . . . .	234
22. Matrice des droits d'accès des commandes LIMIT_COMMANDS . . . . .	169	47. Personnalisations de la version 7.6 . . . . .	288
23. Profils JESSPOOL étendus . . . . .	169	48. Personnalisations de la version 7.5 . . . . .	294
24. Matrice des droits d'accès de consultation LIMIT_VIEW . . . . .	171	49. Personnalisations de la version 7.1 . . . . .	297
25. Mécanismes de stockage des certificats SSL	172	50. Mécanismes de stockage des certificats SSL	300
		51. Définitions locales disponibles pour le programme de résolution . . . . .	321
		52. Publications référencées . . . . .	351
		53. Sites Web référencés . . . . .	353
		54. Publications d'information . . . . .	353



---

## A propos de ce manuel

Le présent manuel décrit les fonctions de configuration d'IBM Rational Developer for System z. Il fournit les instructions permettant de configurer IBM Rational Developer for System z Version 7.6.1 sur votre système hôte z/OS.

A partir de maintenant, les noms suivants sont utilisés dans le présent ouvrage :

- *IBM Rational Developer for System z* est appelé *Developer for System z*.
- L'abréviation utilisée pour *Common Access Repository Manager* est *CARMA*.
- *Software Configuration and Library Manager Developer Toolkit* est appelé *SCLM Developer Toolkit* et parfois abrégé en *SCLMDT*.
- *z/OS UNIX® System Services* est appelé *z/OS UNIX*.
- *Customer Information Control System Transaction Server* est appelé *CICSTS*, abrégé en *CICS*.

Pour les éditions antérieures, telles qu'IBM WebSphere Developer for System z, IBM WebSphere Developer for zSeries et IBM® WebSphere Studio Enterprise Developer, utilisez les informations du guide de configuration de l'hôte et des répertoires de programme de ces éditions.

---

## A qui s'adresse ce guide

Le présent document s'adresse aux programmeurs système qui souhaitent installer et configurer IBM Rational Developer for System z Version 7.6.1, FMID HHOP760, sur leur système hôte z/OS.

Il fournit les procédures détaillées de configuration complète du produit, y compris certains scénarios autres que ceux fournis par défaut. Avant d'utiliser le présent manuel, vous devez maîtriser les systèmes hôtes z/OS UNIX System Services et MVS.

---

## Récapitulatif des changements

Cette section récapitule les changements apportés au document *Rational Developer for System z Version 7.6.1 - Guide de configuration de l'hôte*, SC11-6285-03 (mise à jour mai 2010).

Les changements et ajouts techniques au texte et illustrations sont indiqués par un trait vertical situé à gauche du changement.

Ce document contient des informations précédemment présentées dans le document *Rational Developer for System z Version 7.6 - Guide de configuration de l'hôte*, SC11-6285-02.

Nouvelles informations :

- Les corrections et informations supplémentaires présentées dans *Rational Developer for System z v7.6 Host Configuration Release Notes* sont incluses.
- Notes de migrations spécifiques de la Version 7.6.1. Voir «Notes de migration relatives à la Version 7.6.1», à la page 285.
- Ajout de la présentation du document. Voir «Description du contenu du document», à la page xiv.



- Prise en charge de Java™ 64-bit. Voir Annexe E, «Conditions requises», à la page 339.
- Configuration du produit via des panneaux ISPF. Voir «Préparation de la configuration», à la page 9.
- Nouvelles directives dans rsed.envvars. Voir «rsed.envvars - Fichier de configuration RSE», à la page 32.
- Nouveaux membres de bibliothèque de procédure. Voir «Procédures de construction à distance ELAXF\*», à la page 25.
- Nouvel agencement VSAM CARMA. Voir «Notes de migration VSAM CARMA», à la page 53.
- Prise en charge de plusieurs mémoires RAM CARMA. Voir «(Facultatif) Prise en charge de plusieurs RAM», à la page 70.
- Nouvelles options du gestionnaire de déploiement d'application. Voir «Utilitaire d'administration», à la page 261.
- Nouvel agencement VSAM du gestionnaire de déploiement d'application. Voir «Notes de migration de l'utilitaire d'administration», à la page 265.
- Nouvelles commandes de l'opérateur. Voir Chapitre 8, «Commandes de l'opérateur», à la page 127.
- Nouveaux messages de console. Voir «Messages de la console», à la page 134.
- Nouvelles informations de gestion de charge de travail. Voir Chapitre 12, «Remarques à propos de WLM», à la page 205.

---

## Description du contenu du document

Cette section récapitule les informations présentées dans ce document.

### Planification

Utilisez les informations du présent chapitre afin de planifier l'installation et le déploiement de Developer for System z.

### Personnalisation de base

La procédure de personnalisation ci-dessous s'applique à la configuration du produit Developer for System z de base :

- configuration personnalisée
- modifications de PARMLIB
- modifications de PROCLIB
- définitions de sécurité
- FEJCNFG, fichier de configuration du moniteur de travaux JES
- rsed.envvars, fichier de configuration RSE
- ISPF.conf, fichier de configuration de la passerelle client TSO/ISPF d'ISPF

### (Facultatif) Common Access Repository Manager (CARMA)

Common Access Repository Manager (CARMA) représente un gain de productivité pour les développeurs qui créent des gestionnaires d'accès au référentiel (RAM). Un RAM constitue une interface de programmation d'application pour les gestionnaires de configuration logicielle (SCM) basés sur z/OS.

Inversement, les applications créées par des utilisateurs peuvent démarrer un serveur CARMA qui charge les RAM et fournit une interface standard pour accéder au SCM.

L'interface d'IBM Rational Developer for System z pour le gestionnaire de configuration logicielle de CA Endevor donne aux clients Developer for System z un accès direct au SCM de CA Endevor SCM.

## **(Facultatif) Gestionnaire de déploiement d'application**

Developer for System z utilise certaines fonctions du Gestionnaire de déploiement d'application comme approche commune de déploiement pour plusieurs composants. La personnalisation facultative active davantage de fonctions d'Application Deployment Manager et peut ajouter les services suivants à Developer for System z :

- IBM CICS Explorer apporte une infrastructure Eclipse qui permet de visualiser et gérer les ressources CICS et qui assure une plus grande intégration entre les outils CICS.
- Le client et le serveur CICS Resource Definition (CRD) offrent les fonctions suivantes :
  - Editeur de définition de ressource CICS
  - Permet aux développeurs d'applications de définir des ressources CICS de manière limitée, contrôlée et sécurisée.
  - Empêche l'accès au développement CICS aux fichiers VSAM non autorisés ou incorrects en offrant à l'administrateur CICS le contrôle de l'attribut de nom de fichier physique dans les définitions de fichier.
  - Aides diverses au développement CICS
  - Aides diverses au développement de services Web CICS

## **(Facultatif) SCLM Developer Toolkit**

SCLM Developer Toolkit fournit les outils nécessaires à l'extension des fonctions de SCLM sur le client. SCLM est lui-même un gestionnaire de code source hôte livré comme partie intégrante d'ISPF.

SCLM Developer Toolkit intègre un module d'extension reposant sur Eclipse qui sert d'interface avec SCLM et fournit l'accès à tous les processus SCLM de développement du code existant. Il assure également la prise en charge du développement intégral de Java et J2EE sur le poste de travail avec la synchronisation de SCLM sur le grand système, y compris la construction, l'assemblage et le déploiement du code J2EE à partir du grand système.

## **(Facultatif) Tâches de personnalisation alternatives**

La présente section regroupe diverses tâches de personnalisation facultatives. Suivez les instructions de la section adéquate pour configurer le service souhaité.

- procédure mémorisée DB2
- Support EST (Enterprise Service Tools)
- Prise en charge de la langue bidirectionnelle CICS
- Messages d'erreur de diagnostic IRZ
- Chiffrement RSE SSL
- RSE, traçage
- Groupes de propriétés basés sur l'hôte
- Projets basés sur l'hôte
- Intégration du gestionnaire de fichiers
- Caractères non modifiables
- Utilisation de REXEC (ou SSH)

- Transaction APPC pour le service Commandes TSO
- Nettoyage de WORKAREA

## Vérification de l'installation

Une fois que vous avez terminé la personnalisation du produit, vous pouvez utiliser les programmes de vérification de l'installation décrits dans ce chapitre pour vérifier que l'installation des principaux composants du produit a abouti.

## Commandes de l'opérateur

Le présent chapitre répertorie les commandes de l'opérateur (ou de la console) disponibles pour Developer for System z.

## Identification des incidents liés à la configuration

Le présent chapitre vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration de Developer for System z. Il se compose des sections suivantes :

- Journal et analyse de configuration à l'aide de FEKLOGS
- Fichiers journaux
- Fichiers de vidage
- Traçage
- UNIX z/OS, données de droits
- Ports TCP/IP réservés
- Taille d'espace adresse
- Transaction APPC et service Commandes TSO
- Informations diverses

## Remarques relatives à la sécurité

Developer for System z fournit un accès grand système aux utilisateurs d'un poste de travail standard. La validation des demandes de connexion, l'établissement de communications sécurisées entre l'hôte et le poste de travail, l'autorisation et l'activité d'audit sont donc des aspects fondamentaux de la configuration d'un produit.

## Compréhension de Developer for System z

Le système hôte Developer for System z est composé de plusieurs composants qui interagissent pour permettre au client d'accéder aux services et données de l'hôte. En comprenant bien la conception de ces composants, vous pouvez prendre les bonnes décisions de configuration.

## Remarques à propos de WLM

Contrairement à des applications z/OS traditionnelles, Developer for System z n'est pas une application monolithique qui peut être identifiée facilement au niveau du Workload Manager (WLM). Les différents composants de Developer for System z interagissent pour offrir au client un accès à des services et des données d'hôte. Certains de ces services sont actifs dans différents espaces adresse, ce qui se traduit par différentes classifications WLM.

## Remarques relatives à l'optimisation

RSE (Remote Systems Explorer) est le cœur de Developer for System z. Pour gérer les connexions et charges de travail provenant des clients, RSE est composé d'un

espace adresse de démon, qui permet de contrôler les espaces adresse du groupe d'unités d'exécution. Le démon agit comme un point focal pour la connexion et la gestion, alors que les pools d'unités d'exécution traitent les charges de travail du client.

RSE devient une cible privilégiée d'optimisation de la configuration de Developer for System z. Toutefois, la gestion de centaines d'utilisateurs, chacun utilisant au moins 16 unités d'exécution, d'une certaine quantité de mémoire et éventuellement d'un ou de plusieurs espaces adresses implique de configurer correctement Developer for System z et z/OS.

## **Remarques relatives aux performances**

z/OS est un système d'exploitation hautement personnalisable, et des modifications (parfois mineures) du système peuvent présenter un impact très important sur les performances globales. Le présent chapitre met en évidence certaines modifications qui peuvent être apportées afin d'améliorer les performances de Developer for System z.

## **Remarques à propos de CICSTS**

Ce chapitre contient des informations utiles pour un administrateur CICS Transaction Server.

## **Personnalisation de l'environnement TSO**

Ce chapitre vous aide à simuler une procédure d'ouverture de session TSO en ajoutant des instructions de définition de données et des fichiers à l'environnement TSO dans Developer for System z.

## **Exécution de plusieurs instances**

Parfois, vous pouvez avoir besoin de plusieurs instances de Developer for System z actives sur un même système, lors du test d'une mise à niveau, par exemple. Cependant, certaines ressources (les ports TCP/IP, par exemple) ne peuvent pas être partagées. Les paramètres par défaut ne sont donc pas toujours applicables. Consultez les informations de ce chapitre afin de programmer la coexistence des différentes instances de Developer for System z, pour pouvoir ensuite les personnaliser à l'aide de ce guide de configuration.

## **Guide de migration**

La présente section met en évidence les modifications de l'installation et de la configuration par rapport aux précédentes éditions du produit. Elle fournit également des instructions générales pour la migration de cette édition. Pour de plus amples informations, consultez les sections associées dans ce manuel.

## **Configuration de l'authentification SSL et X.509**

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration du protocole SSL (Secure Socket Layer), ou au cours de la vérification ou de la modification d'une configuration existante. Elle contient un exemple de configuration pour prendre en charge les utilisateurs qui s'authentifient à l'aide d'un certificat X.509.

## **Configuration de TCP/IP**

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration du TCP/IP, ou au cours de la vérification ou de la modification d'une configuration existante.

## **Configuration d'INETD**

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration d'INETD ou au cours de la vérification ou de la modification d'une configuration existante. INETD est utilisé par Developer for System z pour la fonction REXEC/SSH.

## **Configuration APPC**

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration d'APPC (Advanced Program-to-Program Communication) ou au cours de la vérification ou de la modification d'une configuration existante.

## **Conditions requises**

La présente annexe répertorie les conditions requises et les corequis liés à l'hôte correspondant à cette version de Developer for System z.

---

## **Partie 1. Personnalisation de Developer for System z**





---

## Chapitre 1. Planification

Utilisez les informations du présent chapitre conjointement avec les informations contenues de l'Annexe E, «Conditions requises», à la page 339 afin de planifier l'installation et le déploiement de Developer for System z. Les sujets suivants sont abordés :

- «Remarques relatives à la migration»
- «Remarques relatives à la planification»
- «Remarques relatives à la pré-installation», à la page 4
- «Préparation de la configuration», à la page 9
- «Remarques relatives au prédéploiement», à la page 12
- «Liste de contrôle du client», à la page 13

---

### Remarques relatives à la migration

Le Chapitre 18, «Guide de migration», à la page 283 décrit les modifications de l'installation et de la configuration par rapport aux précédentes éditions du produit. Utilisez ces informations pour planifier votre migration vers la version en cours de Developer for System z.

#### Remarques :

1. Si vous disposez de la version précédente d'IBM Rational Developer for System z, d'IBM WebSphere Developer for System z, d'IBM WebSphere Developer for zSeries ou d'IBM WebSphere Studio Enterprise Developer, il est recommandé de sauvegarder les fichiers personnalisés associés AVANT d'installer IBM Rational Developer for System z Version 7.6.1. Pour connaître les fichiers qui nécessitent une personnalisation, consultez la section Chapitre 18, «Guide de migration», à la page 283.
2. Si vous prévoyez d'exécuter plusieurs instances de Developer for System z, reportez-vous à la section Chapitre 17, «Exécution de plusieurs instances», à la page 277.

---

### Remarques relatives à la planification

#### Présentation du produit

Developer for System z est composé d'un client, installé sur l'ordinateur personnel de l'utilisateur, et d'un serveur, installé sur un ou plusieurs hôtes. La présente documentation va porter sur l'hôte, à savoir un système z/OS. Toutefois, d'autres systèmes d'exploitation comme AIX et Linux® on System z, sont également pris en charge.

Le client offre aux développeurs un environnement de développement reposant sur Eclipse- et assurant à l'hôte l'uniformité de l'interface graphique, ce qui permet, entre autres, de décharger les travaux de l'hôte sur le client, en économisant les ressources de l'hôte.

La partie de l'hôte est composée de plusieurs tâches actives en permanence et de tâches démarrées ad-hoc. Ces tâches permettent au client de gérer les différents

composants de l'hôte z/OS (les ensembles de données MVS, les commandes TSO, les fichiers et commandes z/OS UNIX, les soumissions de travaux et les sorties de travaux, par exemple).

Developer for System z peut également interagir avec les sous-systèmes et les autres logiciels d'application installés sur l'hôte (CICS, Debug Tool et les gestionnaires de configuration de logiciel (SCM), par exemple) si Developer for system z est configuré pour le faire, et si ces produits (configuration annexe requise) sont disponibles.

Voir le Chapitre 11, «Compréhension de Developer for System z», à la page 193 pour obtenir des informations de base sur la conception de Developer for System z.

Consultez le site Web de Developer for System z (<http://www-01.ibm.com/software/awdtools/rdz/>) ou votre représentant IBM local pour obtenir des informations complémentaires sur les fonctionnalités proposées par Developer for System z.

## Compétences requises

Des compétences SMP/E minimales sont nécessaires pour installer l'hôte Developer for System z.

La configuration de Developer for System z nécessite plus de droits et de connaissances que ceux dont dispose habituellement le programmeur système. Une assistance peut donc s'avérer nécessaire. Le tableau 3, à la page 7 et le tableau 4, à la page 8 répertorient les administrateurs nécessaires pour effectuer les tâches de personnalisation obligatoires et facultatives.

## Temps nécessaire

Le temps nécessaire à l'installation et à la configuration des composants de l'hôte System z varie en fonction des facteurs suivants :

- la configuration z/OS UNIX et TCP/IP en cours
- la disponibilité du logiciel prérequis et de la maintenance
- la définition des segments OMVS des utilisateurs de Developer for System z
- la disponibilité d'un utilisateur ayant réussi l'installation du client pour tester l'installation et signaler les éventuels incidents

L'expérience a montré qu'un et quatre jours sont nécessaires pour mener à bien le processus d'installation et de configuration de System z. Il s'agit du temps nécessaire à une installation impeccable réalisée par un programmeur système expérimenté. En cas d'incidents ou d'indisponibilité des compétences requises, l'installation peut prendre plus de temps.

---

## Remarques relatives à la pré-installation

Voir le document *Program Directory for IBM Rational Developer for System z* (GI11-7314) pour obtenir des instructions détaillées sur l'installation SMP/E du produit.

**Remarque :** Le système de fichiers (HFS ou zFS) dans lequel Developer for System z est installé doit être monté avec le contrôle des données de droits SETUID activé (il s'agit de la valeur par défaut du système). Le montage du système de fichiers avec le paramètre NOSETUID empêche

Developer for System z de créer l'environnement de sécurité utilisateur et l'exécution de la requête de connexion du client.

Voir Chapitre 17, «Exécution de plusieurs instances», à la page 277 si vous prévoyez d'exécuter plusieurs instances de Developer for System z.

## Choix de configuration

Developer for System z vous permet de choisir le mode d'accès au service Commandes TSO. Le choix effectué ici a un impact sur la configuration des prérequis nécessaire. Vous devez sélectionner et configurer l'une des méthodes suivantes :

- Service de passerelle client TSO/ISPF d'ISPF, qui nécessite un niveau service ISPF minimal. Il s'agit de la méthode par défaut utilisée dans les exemples fournis.
- Transaction APPC (comme dans les éditions précédant la version 7.1).

**Remarque :** La passerelle client TSO/ISPF d'ISPF est également utilisée par SCLM Developer Toolkit et éventuellement par une méthode de démarrage alternative de Common Access Repository Manager (CARMA).

## Éléments prérequis pour les produits

L'Annexe E, «Conditions requises», à la page 339 contient une liste de logiciels prérequis qui doivent être installés et opérationnels pour que Developer for System z fonctionne. Il y a également une liste de logiciels corequis pour la prise en charge de fonctions spécifiques de Developer for System z. Ces éléments requis doivent être installés et opérationnels au moment de l'exécution pour que les fonctions correspondantes opèrent selon leur conception.

Voir *Rational Developer for System z Prerequisites* (SC23-7659) dans la bibliothèque en ligne de Developer for System z (<http://www-01.ibm.com/software/awdtools/rdz/library/>) pour obtenir une liste à jour des produits requis et corequis pour votre version de Developer for System z. Planifiez à l'avance l'obtention de ces produits requis car la procédure peut prendre du temps, en fonction des règles en vigueur sur votre site. Les principales conditions requises d'une configuration de base sont les suivantes :

- z/OS 1.8 ou version supérieure
- ISPF APAR OA29489 (passerelle client TSO/ISPF)
- Java 5.0 ou supérieur

**Remarque :** Le PTF pour Developer for System z APAR PM07305 doit être appliqué si vous utilisez une version 64 bits de Java. Les PTF sont accessibles via la page de service recommandée Developer for System z, <http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335>.

## Ressources requises

Developer for System z nécessite d'allouer les ressources système répertoriées dans le tableau 1, à la page 6. Les ressources répertoriées dans tableau 2, à la page 6 sont requises pour des services en option. Anticipez l'obtention de ces ressources disponibles, car leur obtention peut prendre du temps, suivant les règles en vigueur sur votre site.

**Remarque :** Developer for System z est composé de plusieurs tâches communiquant les unes avec les autres, et avec le client. Ces tâches utilisent différents temporisateurs pour détecter les pertes de communication avec leurs partenaires. Cela implique que des problèmes de dépassement du délai d'attente peuvent se produire (en raison du manque de temps UC dans la fenêtre de dépassement du délai d'attente) sur les systèmes dont la charge de l'UC est élevée ou dont les paramètres WLM (WorkLoad Management) sont incorrects pour Developer for system z.

*Tableau 1. Ressources requises*

Ressource	Valeur par défaut	Information
fichier APF autorisé	FEK.SFEKAUTH	«Autorisations APF dans PROGxx», à la page 18
tâche démarrée	JMON, RSED et LOCKD	«Remarques relatives au serveur», à la page 10
port pour une utilisation limitée au système hôte	6715	«FEJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27
port pour une utilisation limitée au système hôte	4036	«rsed.envvars - Fichier de configuration RSE», à la page 32
port pour les communications client-hôte	4035	«Modifications de PROCLIB», à la page 21
plage de ports pour les communications client-hôte	tout port disponible est utilisé	«Définition de PORTRANGE disponible pour un serveur RSE», à la page 41
Définition de sécurité d'application	Accès universel READ (lecture) pour FEKAPPL	«Définition de la protection d'application pour RSE», à la page 189
Définition de sécurité PassTicket (mot de passe associé)	aucun port par défaut	«Définition du support de mots de passe PassTicket pour RSE», à la page 189

*Tableau 2. Ressources optionnelles*

Ressource	Valeur par défaut	Information
fichier LINKLIST	FEK.SFEKAUTH et FEK.SFEKLOAD	Chapitre 5, «(Facultatif) SCLM Developer Toolkit», à la page 83
Ensemble de données LPA	FEK.SFEKLPA	Chapitre 3, «(Facultatif) Common Access Repository Manager (CARMA)», à la page 51
plage de ports pour une utilisation limitée au système hôte	5227-5326 (100 ports)	Chapitre 3, «(Facultatif) Common Access Repository Manager (CARMA)», à la page 51
ports pour une utilisation limitée au système hôte	tout port disponible est utilisé	«(Facultatif) Transaction APPC pour le service Commandes TSO», à la page 106

Tableau 2. Ressources optionnelles (suite)

Ressource	Valeur par défaut	Information
port pour les communications client-hôte	aucun port par défaut	Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75
mise à jour CSD CICS	valeurs multiples	Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75
mise à jour du JCL CICS	FEK.SFEKLOAD	<ul style="list-style-type: none"> <li>Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75</li> <li>«(Facultatif) Support de langue bidirectionnelle CICS», à la page 94</li> </ul>

La configuration de Developer for System z nécessite plus de droits et de connaissances que ceux dont dispose habituellement le programmeur système. Une assistance minimale peut donc s'avérer nécessaire. Le tableau 3 et le tableau 4, à la page 8 répertorient les administrateurs nécessaires pour effectuer les tâches de personnalisation obligatoires et facultatives.

Tableau 3. Administrateurs requis pour les tâches requises

Administrateur	Tâche	Information
Système	Des actions typiques du programmeur système sont requises pour les tâches de personnalisation	Sans objet
Sécurité	<ul style="list-style-type: none"> <li>Définition d'un segment OMVS pour les utilisateurs de Developer for System z</li> <li>Définition des profils de fichier</li> <li>Définition des tâches démarrées</li> <li>Définition de la sécurité de la commande de l'opérateur</li> <li>Définition des profils de serveur z/OS UNIX</li> <li>Définition de la sécurité d'application</li> <li>Définition de la prise en charge de PassTicket</li> <li>Définition de fichiers contrôlés par un programme</li> <li>Définition de fichiers z/OS UNIX contrôlés par programme</li> </ul>	Chapitre 10, «Remarques relatives à la sécurité», à la page 161
TCP/IP	Définition de nouveaux ports TCP/IP	«Ports TCP/IP», à la page 164

Tableau 3. Administrateurs requis pour les tâches requises (suite)

Administrateur	Tâche	Information
WLM	Attribution des objectifs de la tâche démarrée aux serveurs et à leurs processus enfants	Chapitre 12, «Remarques à propos de WLM», à la page 205

Tableau 4. Administrateurs nécessaires pour les tâches optionnelles

Administrateur	Tâche	Information
Système	Des actions typiques du programmeur système sont requises pour les tâches de personnalisation	Sans objet
Sécurité	<ul style="list-style-type: none"> <li>• Définition des profils de fichier</li> <li>• Définition de fichiers contrôlés par un programme</li> <li>• Définition de droits pour la soumission de travaux xxx*</li> <li>• Définition de la sécurité des transactions CICS</li> <li>• Ajout de certificat pour SSL</li> <li>• Configuration du support de certificat client X.509</li> </ul>	<ul style="list-style-type: none"> <li>• Chapitre 10, «Remarques relatives à la sécurité», à la page 161</li> <li>• Chapitre 15, «Remarques à propos de CICSTS», à la page 257</li> <li>• Annexe A, «Configuration de l'authentification SSL et X.509», à la page 299</li> </ul>
TCP/IP	Définition de nouveaux ports TCP/IP	«Ports TCP/IP», à la page 164
SCLM	<ul style="list-style-type: none"> <li>• Définition de traducteurs de langage pour le support JAVA/J2EE</li> <li>• Définition de types SCLM pour le support JAVA/J2EE</li> </ul>	Chapitre 5, «(Facultatif) SCLM Developer Toolkit», à la page 83
CICS TS	<ul style="list-style-type: none"> <li>• Mise à jour du JCL de la région CICS</li> <li>• Mise à jour du CSD de la région CICS</li> <li>• Définition de groupe CICS</li> <li>• Définition de noms de transaction CICS</li> <li>• Définition d'un programme dans CICS</li> </ul>	<ul style="list-style-type: none"> <li>• Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75</li> <li>• «(Facultatif) Support de langue bidirectionnelle CICS», à la page 94</li> </ul>
DB2	Définition d'une procédure mémorisée DB2	«(Facultatif) Procédure mémorisée DB2», à la page 91

Tableau 4. Administrateurs nécessaires pour les tâches optionnelles (suite)

Administrateur	Tâche	Information
WLM	<ul style="list-style-type: none"> <li>Attribution d'objectifs à une procédure mémorisée DB2</li> <li>Attribution d'objectifs TSO à une transaction APPC</li> </ul>	<ul style="list-style-type: none"> <li>«(Facultatif) Procédure mémorisée DB2», à la page 91</li> <li>Chapitre 12, «Remarques à propos de WLM», à la page 205</li> </ul>
APPC	Définition d'une transaction APPC	«(Facultatif) Transaction APPC pour le service Commandes TSO», à la page 106

## Préparation de la configuration

### Gestion de la charge de travail

Contrairement aux application z/OS classiques, Developer for System z n'est pas une application monolithique qui peut être identifiée facilement au niveau du Workload Manager (WLM). Les différents composants qui constituent Developer for System z interagissent pour offrir au client un accès à des services et des données d'hôte. Voir Chapitre 12, «Remarques à propos de WLM», à la page 205 pour prévoir votre configuration WLM en conséquence.

### Utilisation des ressources et limites du système

Developer for System z utilise un nombre variable de ressources système (les espaces adresse et les processus et unités d'exécution z/OS UNIX, par exemple). La disponibilité de ces ressources est limitée par différentes définitions du système. Voir le Chapitre 13, «Remarques relatives à l'optimisation», à la page 215 pour estimer l'utilisation des ressources clé et planifier la configuration système en conséquence.

### Configuration nécessaire des produits requis

Vérifiez auprès de votre programmeur système MVS, de votre administrateur sécurité et de votre administrateur TCP/IP que les logiciels et produits requis sont installés, testés et qu'ils fonctionnent. Certaines tâches de personnalisation requises facilement omises sont répertoriées ci-après :

- Tous les utilisateurs de Developer for System z doivent disposer de droits READ et EXECUTE dans les répertoires Java.
- Tous les utilisateurs de Developer for System z doivent disposer des droits READ, WRITE et EXECUTE dans le répertoire /tmp/.
- Les actions à distance (basées sur l'hôte) pour les sous-projets z/OS UNIX requièrent l'activation de la version z/OS UNIX de REXEC ou de SSH sur l'hôte.

### Remarques relatives à l'ID utilisateur

L'ID d'un utilisateur de Developer for System z doit contenir (au moins) les attributs suivants :

- Un accès TSO (avec une taille de région normale).

**Remarque :** Une taille de région importante est nécessaire pour l'ID utilisateur qui exécute les programmes de vérification d'installation (IVP), les



fonctions demandant beaucoup de mémoire (Java, par exemple) étant exécutées. Il est recommandé d'attribuer au moins 131072 octets (128 mégaoctets) ou plus à la taille de région.

- Un segment OMVS défini sur le système de sécurité (RACF, par exemple), à la fois pour l'ID utilisateur et pour son groupe par défaut.
  - La zone HOME doit faire référence à un répertoire de base alloué pour l'utilisateur (avec des droits d'accès WRITE, READ et EXECUTE).
  - La zone PROGRAM du segment OMVS doit être /bin/sh ou un autre shell z/OS UNIX valide, comme /bin/tcsh.
  - Le champ ASSIZEMAX ne doit pas être rempli, de sorte que les valeurs par défaut du système soient utilisées.
  - L'ID utilisateur ne nécessite pas d'UID 0.

Exemple (commande **LISTUSER userid NORACF OMVS**) :

USER=userid

```
OMVS INFORMATION
-----
UID= 0000003200
HOME= /u/userid
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMA= NONE
PROCUSERMA= NONE
THREADSMA= NONE
MMAPAREAMA= NONE
```

- Le groupe par défaut de l'ID utilisateur nécessite un GID.

Exemple (commande **LISTGRP group NORACF OMVS**):

GROUP group

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

- Des droits d'accès READ et EXECUTE aux répertoires et fichiers d'installation et de configuration de Developer for System z, par défaut /usr/lpp/rdz/\*, /etc/rdz/\* et /var/rdz/\*.
- Des droits d'accès READ, WRITE et EXECUTE au répertoire WORKAREA de Developer for System z, par défaut /var/rdz/WORKAREA.
- Des droits d'accès READ aux fichiers d'installation de Developer for System z, par défaut FEK.SFEK\*.

## Remarques relatives au serveur

Developer for System z comprend trois serveurs actifs en permanence, qui peuvent être des tâches démarrées ou des travaux utilisateur. Ces serveurs fournissent les services demandés eux-mêmes ou démarrent d'autres serveurs (tels que des unités d'exécution ou travaux utilisateur z/OS UNIX ) pour fournir le service.

- Le moniteur de travaux JES (JMON) offre tous les services liés à JES.
- Le démon lock (LOCKD) fournit des services de suivi pour les verrous de fichiers.
- L'Explorateur de systèmes distants (RSE) fournit des services de base, comme la connexion du client à l'hôte et le démarrage d'autres serveurs pour des services spécifiques. RSE se compose de deux entités logiques :
  - Le démon RSE (RSED), qui gère la configuration de la connexion et qui est en charge de l'exécution en mode serveur unique.

- Le serveur RSE qui gère les demandes client individuelles.

Le moniteur de travaux JES (JMON) fournit tous les services relatifs à JES.

- Les mécanismes de sécurité utilisés par le gestionnaire doivent reposer sur des fichiers sécurisés. Cela signifie que seuls des administrateurs système dignes de confiance sont autorisés à mettre à jour les bibliothèques et les fichiers de configuration.

L'Explorateur de systèmes distants (RSE) est le composant Developer for System z qui fournit les services de base comme la connexion du client à l'hôte.

- Depuis la version 7.5, le démon RSE n'est plus un processus géré par INETD mais une tâche démarrée.
- Depuis la version 7.5, le serveur RSE utilise un modèle de serveur unique, alors que dans les versions précédentes, chaque connexion client-hôte avait un serveur RSE privé.
- Différents niveaux de sécurité des communications sont pris en charge par RSE :
  - Les communications (client-hôte) externes peuvent être limitées à des ports spécifiques. Cette fonction est désactivée par défaut.
  - Les communications (client-hôte) externes peuvent être chiffrées à l'aide de la couche SSL. Cette fonction est désactivée par défaut.
  - La vérification du port d'entrée peut être utilisée afin d'autoriser l'accès uniquement aux adresses TCP/IP sécurisées. Cette fonction est désactivée par défaut.
- RSE prend également en charge plusieurs méthodes d'authentification client :
  - Un ID utilisateur et un mot de passe
  - Un ID utilisateur et un mot de passe utilisable une seule fois
  - Un certificat X.509
- Les mécanismes de sécurité utilisés par RSE doivent reposer sur un système de fichiers sécurisé. Cela signifie que seuls des administrateurs système dignes de confiance sont autorisés à mettre à jour les bibliothèques et les fichiers de configuration.

Comme expliqué à la section «Ports TCP/IP», à la page 164, certains services hôte (et donc leurs ports) doivent être disponibles pour que le client puisse s'y connecter et doivent être définis sur le pare-feu qui protège l'hôte. Tous les autres ports utilisés par Developer for System z reçoivent un trafic réservé à l'hôte. Les ports répertoriés ci-après sont les ports requis pour une configuration de base du produit Developer for System z.

- Démon RSE pour la configuration des communications client-hôte (utilisation du protocole TCP, port par défaut 4035).
- Serveur RSE pour les communications hôte-client (à l'aide du protocole TCP). Par défaut, tout port disponible est utilisé mais vous pouvez limiter les ports à une plage définie.

**Remarque :** Les anciens clients (version 7.0 et plus ancienne) communiquent directement avec le moniteur de travaux JES (via le protocole TCP). Le port utilisé par défaut est le port 6715.

## Méthode de configuration

A partir de la version 7.6.1, Developer for System z offre une méthode alternative, via une application de panneaux ISPF, pour la configuration du côté hôte du produit. Ceci vous donne un choix parmi les méthodes suivantes :

- Utilisation de l'application de panneaux ISPF. Ce guide vous conduira tout au long des étapes de personnalisation obligatoires et de celles facultatives que vous aurez sélectionnées. Pour plus d'informations, voir le livre blanc *Host Configuration Utility*, que vous trouverez dans la bibliothèque Internet de Developer for System z, <http://www-306.ibm.com/software/awdtools/rdz/library/>.
- Utilisation du *Guide de démarrage rapide de configuration de l'hôte*. Ce guide vous conduira tout au long des étapes de personnalisation obligatoire. Le périmètre de ce guide se limite à une configuration de base.
- Utilisation du *Guide de configuration de l'hôte*. Ce guide vous conduira tout au long des étapes de personnalisation obligatoires et de toutes celles facultatives. Toutes les options configurables sont couvertes dans ce guide, y compris certains scénarios autres que par défaut

---

## Remarques relatives au prédéploiement

Developer for System z prend en charge le clonage d'une installation sur un système différent, ce qui évite d'avoir à installer SMP/E sur chaque système.

Les ensembles de fichiers, répertoires et fichiers suivants sont obligatoires pour le déploiement sur d'autres systèmes. Si vous avez copié un fichier à un emplacement différent, ce fichier doit remplacer son équivalent dans la liste ci-après.

**Remarque :** La liste ci-après ne couvre pas les besoins liés au déploiement des logiciels prérequis et corequis.

- FEK.SFEKAUTH(\*)
- FEK.SFEKLOAD(\*)
- FEK.SFEKPROC(\*)
- FEK.#CUST.PARMLIB(\*)
- FEK.#CUST.PROCLIB(\*)
- /usr/lpp/rdz/\*
- /etc/rdz/\*
- /var/rdz/\* (arborescence des répertoires uniquement)
- Composants facultatifs :
  - FEK.SFEKLPA(\*)
  - FEK.#CUST.CNTL(\*)
  - définitions, fichiers et répertoires issus des travaux de personnalisation dans FEK.#CUST.JCL

### Remarques :

1. FEK et /usr/lpp/rdz correspondent au qualificatif de haut niveau et au chemin d'accès utilisés lors de l'installation du produit. FEK.#CUST, /etc/rdz et /var/rdz désignent les emplacements par défaut utilisés au cours de la personnalisation du produit (voir «Configuration personnalisée», à la page 15 pour de plus amples informations).
2. Il est recommandé d'installer Developer for System z dans un système de fichiers privé (HFS ou zFS) pour faciliter le déploiement des composants z/OS UNIX du produit.
3. Si vous ne pouvez pas utiliser un système de fichiers privé, il est recommandé d'utiliser un outil d'archivage (la commande z/OS UNIX tar, par exemple)

pour transférer les répertoires z/OS UNIX d'un système à un autre. Il s'agit de préserver les attributs (tels que le contrôle par programme) des fichiers et répertoires Developer for System z.

Pour plus d'informations sur les exemples de commande ci-dessous qui permettent d'archiver et de restaurer le répertoire d'installation de Developer for System z, voir le document *UNIX System Services Command Reference* (SA22-7802)

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

## Liste de contrôle du client

Les utilisateur du client Developer for System z doivent connaître les résultats de certaines personnalisations de l'hôte, telles que les numéros de port TCP/IP, pour que le client fonctionne correctement. Utilisez les listes de contrôle ci-après pour rassembler les informations nécessaires.

La liste de contrôle dans le tableau 5 répertorie les résultats requis pour les étapes de personnalisation obligatoires. Le tableau 6 répertorie les résultats requis pour les étapes de personnalisation facultatives.

*Tableau 5. Liste de contrôle du client - Composants obligatoires*

Personnalisation	Valeur
Numéro de port de serveur du moniteur de travaux JES (par défaut 6715) :  Voir SERV_PORT dans «FEJJCNF, fichier de configuration du moniteur de travaux JES», à la page 27.	
Numéro de port TCP/IP du démon RSE (4035 par défaut) :  Voir «Démon RSE», à la page 22.	

*Tableau 6. Liste de contrôle du client - Composants facultatifs*

Personnalisation	Valeur
Emplacement des procédures ELAXF* si elles ne se trouvent pas dans une bibliothèque de procédure système.  Voir les annotations sur JCLLIB dans «Procédures de construction à distance ELAXF*», à la page 25.	
Procédure ou noms des étapes des procédures ELAXF* si elles ont été modifiées.  Voir les annotations sur leur modification dans «Procédures de construction à distance ELAXF*», à la page 25.	
Nom de la procédure mémorisée DB2 (ELAXMSAM par défaut) :  Voir les informations sur les procédures mémorisées DB2 dans le Chapitre 17, «Exécution de plusieurs instances», à la page 277.	
Emplacement de la procédure mémorisée DB2 si elle ne se trouve pas dans une bibliothèque de procédure système :  Voir «(Facultatif) Procédure mémorisée DB2», à la page 91.	

Tableau 6. Liste de contrôle du client - Composants facultatifs (suite)

Personnalisation	Valeur
(corequis) Numéro de port TN3270 de Host Connect Emulator (23, par défaut).  Voir le Chapitre 10, «Remarques relatives à la sécurité», à la page 161	
(corequis) Numéro de port REXEC ou SSH (par défaut, 512 ou 22, respectivement) :  Voir «(Facultatif) Utilisation de REXEC (ou SSH)», à la page 104.	
Emplacement du fichier server.zseries si la méthode de connexion REXEC/SSH est utilisée (par défaut /etc/rdz).  Voir «(Facultatif) Utilisation de REXEC (ou SSH)», à la page 104.	
Emplacement du JCL CRA#ASLM pour les allocations de fichiers RAM SCLM CARMA (par défaut FEK.#CUST.JCL):  Voir les annotations concernant CRA#ASLM dans «Activation du RAM SCLM», à la page 63.	

---

## Chapitre 2. Personnalisation de base

La procédure de personnalisation ci-dessous s'applique à la configuration du produit Developer for System z de base. Reportez-vous aux chapitres relatifs aux composants facultatifs pour connaître leurs exigences de personnalisation.

---

### Configuration requise et liste de contrôle

Vous avez besoin de l'aide d'un administrateur de sécurité et d'un administrateur TCP/IP pour effectuer cette tâche de personnalisation, qui requiert les tâches de personnalisation spéciales et les ressources suivantes :

- fichier APF autorisé
- différentes mises à jour de PARMLIB
- différentes mises à jour du logiciel de sécurité
- différents ports TCP/IP pour les communications internes et client-hôte

Pour vérifier l'installation et commencer à utiliser Developer for System z sur votre site, vous devez effectuer les tâches ci-après. Sauf indication contraire, toutes les tâches sont obligatoires.

1. Créez des copies personnalisables des exemples et créez l'environnement de travail pour Developer for system z. Pour plus de détails, voir «Configuration personnalisée».
2. Mettez à niveau les limites système z/OS UNIX, lancez des tâches démarrées, définissez des fichiers autorisés APF et LINKLIST et éventuellement des fichiers LPA. Pour plus d'informations, voir «Modifications de PARMLIB», à la page 16.
3. Créez des procédures de tâche démarrée et de compilation/liaison. Pour plus d'informations, voir «Modifications de PROCLIB», à la page 21.
4. Mettez à jour les définitions de sécurité. Pour plus d'informations, voir «Définitions de sécurité», à la page 26. De même, n'oubliez pas que des PassTickets sont utilisés pour établir la sécurité des unités d'exécution sur le serveur. Pour plus d'informations, voir «Utilisation de PassTickets», à la page 166.
5. Personnalisez les fichiers de configuration de Developer for System z. Pour plus d'informations, voir :
  - «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27
  - «rsed.envvars - Fichier de configuration RSE», à la page 32
  - «ISPF.conf, fichier de configuration de la passerelle client TSO/ISPF d'ISPF», à la page 47

---

### Configuration personnalisée

Developer for System z est fourni avec plusieurs exemples de fichiers de configuration et de langage JCL. Pour que vos personnalisations ne soient pas remplacées lors de l'application de la maintenance, il est recommandé de copier tous ces membres et les fichiers z/OS UNIX à un emplacement différent, puis de personnaliser la copie.

Certaines fonctions de Developer for System z requièrent également l'existence de certains répertoires dans z/OS UNIX, qui doivent être créés pendant la

personnalisation du produit. Pour faciliter la procédure d'installation, un exemple de travail, FEKSETUP, est fourni pour créer les copies et les répertoires requis.

Personnalisez et soumettez l'exemple de membre FEKSETUP dans le fichier FEK.SFEKSAMP pour créer des copies personnalisables des fichiers de configuration et du JCL de configuration, et les répertoires z/OS UNIX requis. La procédure de personnalisation requise est décrite dans ce membre.

Dans ce travail vous devez effectuer les tâches suivantes :

- Créez FEK.#CUST.PARMLIB et remplissez-le avec des exemples de fichiers de configuration.
- Créez FEK.#CUST.PROCLIB et remplissez-le avec des exemples de membres SYS1.PROCLIB.
- Créez FEK.#CUST.JCL et remplissez-le avec un exemple de JCL de configuration.
- Créez FEK.#CUST.CNTL et remplissez-le avec des exemples de scripts de démarrage du serveur.
- Créez FEK.#CUST.ASM et remplissez-le avec un exemple de code source en assembleur.
- Créez FEK.#CUST.COBOL et remplissez-le avec un exemple de code source COBOL.
- Créez /etc/rdz/\* et remplissez-le avec des exemples de fichiers de configuration.
- Créez /var/rdz/\* comme répertoires de travail pour plusieurs fonctions Developer for System z.

#### Remarques :

1. Les procédures de configuration de ce manuel utilisent les emplacements membre/fichier créés par le travail FEKSETUP, sauf indication contraire. Les exemples d'origine, qui ne peuvent pas être mis à jour, se trouvent dans FEK.SFEKSAMP et /usr/lpp/rdz/samples/.
2. Pour conserver tous les fichiers Developer for System z z/OS UNIX dans le même système de fichiers (HFS ou zFS), mais également placer les fichiers de configuration dans /etc/rdz, vous pouvez utiliser les liens symboliques. Les exemples de commande z/OS UNIX suivants permettent de créer un répertoire dans le système de fichiers existant (/usr/lpp/rdz/cust) et de définir un lien symbolique (/etc/rdz) vers ce dernier :

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

---

## Modifications de PARMLIB

Pour plus d'informations sur les définitions de PARMLIB répertoriées ci-dessous, reportez-vous au document *MVS Initialization and Tuning Reference* (SA22-7592). Voir le document *MVS System Commands* (SA22-7627) pour de plus amples informations sur les exemples de commandes de la console.

### Définition des limites z/OS UNIX dans BPXPRMxx

L'Explorateur de systèmes distants (RSE), qui fournit des services de base (la connexion du client à l'hôte et le démarrage d'autres serveurs, par exemple) est un processus reposant sur z/OS UNIX. Il est donc indispensable de définir des valeurs appropriées pour les limites système z/OS UNIX dans BPXPRMxx, en fonction du nombre d'utilisateurs Developer for System z actifs simultanément et de leur charge de travail moyenne.



Voir le Chapitre 13, «Remarques relatives à l'optimisation», à la page 215 pour plus d'informations sur les différentes limites définies dans BPXPRMxx et leur incidence sur Developer for System z.

MAXASSIZE définit la taille de la région de l'espace adresse maximal (processus). Définissez MAXASSIZE dans SYS1.PARMLIB(BPXPRMxx) sur 2G. Il s'agit de la valeur maximale autorisée. Il s'agit d'une limite à l'échelle du système. Elle est donc active pour tous les espaces adresses z/OS UNIX. Si elle ne répond pas à vos attentes, vous pouvez la définir uniquement pour Developer for System z dans votre logiciel de sécurité (voir «Définition des tâches démarrées Developer for System z», à la page 185).

MAXTHREADS indique le nombre maximal d'unités d'exécution actives pour un même processus. Associez MAXTHREADS dans SYS1.PARMLIB(BPXPRMxx) à 1500 ou à une valeur supérieure. Il s'agit d'une limite à l'échelle du système. Elle est donc active pour tous les espaces adresses z/OS UNIX. Si elle ne répond pas à vos attentes, vous pouvez la définir uniquement pour Developer for System z dans votre logiciel de sécurité (voir «Définition des tâches démarrées Developer for System z», à la page 185).

MAXTHREADTASKS indique le nombre maximal de tâches MVS actives pour un même processus. Attribuez la valeur 1500 ou une valeur supérieure à MAXTHREADTASKS dans SYS1.PARMLIB(BPXPRMxx). Il s'agit d'une limite à l'échelle du système. Elle est donc active pour tous les espaces adresses z/OS UNIX. Si elle ne répond pas à vos attentes, vous pouvez la définir uniquement pour Developer for System z dans votre logiciel de sécurité (voir «Définition des tâches démarrées Developer for System z», à la page 185).

MAXPROCUSER définit le nombre maximal de processus qui peuvent être actifs simultanément pour un même ID utilisateur z/OS UNIX. Définissez MAXPROCUSER dans SYS1.PARMLIB(BPXPRMxx) sur 50 ou une valeur supérieure. Ce paramètre est censé être une limite à l'échelle du système, étant donné qu'il doit être actif pour chaque client utilisant Developer for System z.

Ces valeurs peuvent être vérifiées et définies dynamiquement (jusqu'à la procédure de chargement initial suivante) à l'aide des commandes de la console ci-dessous :

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

**Remarques :**

1. Vous trouverez de plus amples informations sur les autres emplacements au niveau desquels les tailles d'espace adresse peuvent être définies ou limitées dans la section «Taille d'espace adresse», à la page 155.
2. La valeur MAXPROCUSER utilisée ci-dessus est destinée aux utilisateurs ayant un ID utilisateur z/OS UNIX (UID) unique. Augmentez cette valeur si vos utilisateurs partagent le même numéro d'utilisateur (UID).
3. Assurez-vous que les autres valeurs BPXPRMxx (celles définies pour MAXPROCSYS et MAXUIDS, par exemple) permettent de gérer le nombre prévu d'utilisateurs simultanés de Developer for System z. Pour plus d'informations, voir Chapitre 13, «Remarques relatives à l'optimisation», à la page 215.

## Ajout de tâches démarrées à COMMNDxx

Ajoutez des commandes de démarrage à SYS1.PARMLIB(COMMANDxx) pour les serveurs RSED, LOCKD et JMON Developer for System z afin qu'ils soient démarrés automatiquement lors de la prochaine procédure IPL.

Une fois définis et configurés, les serveurs peuvent être démarrés de manière dynamique (jusqu'au prochain démarrage système) à l'aide des commandes de la console suivantes :

- S RSED
- S LOCKD
- S JMON

**Remarque :** Le démon lock doit être démarré avant que les utilisateurs Developer for System z ne se connectent au démon RSE. Cette procédure permet au démon lock de suivre les demandes de verrouillage de fichiers émises par ces utilisateurs. Vous devez donc démarrer le démon lock au démarrage du système.

## Définitions LPA dans LPALSTxx

Le service facultatif CARMA (Common Access Repository Manager) prend en charge des méthodes de démarrage du serveur alternatives qui ne requièrent pas l'utilisation d'un initiateur JES. La plus flexible de ces alternatives requiert que le module CRASTART dans la bibliothèque de chargement FEK.SFEKLPA se trouve dans la zone permanente de programme (LPA).

Les fichiers LPA sont définis dans SYS1.PARMLIB(LPALSTxx).

Les définitions LPA peuvent être définies dynamiquement (jusqu'au prochain démarrage système) à l'aide des commandes de la console suivantes :

- SETPROG LPA,ADD,DSN=FEK.SFEKLPA

## Autorisations APF dans PROGxx

Pour que le moniteur de travaux JES puisse accéder aux fichiers spoule JES, le module FEJMON de la bibliothèque de chargement FEK.SFEKAUTH et des bibliothèques d'exécution Language Environment (LE) (CEE.SCEERUN\*) doit être autorisé APF.

Pour que le service SCLM Developer Toolkit (facultatif) fonctionne, le module BWBTSOW de la bibliothèque de chargement FEK.SFEKAUTH et de la bibliothèque d'exécution REXX (REXX.\*.SEAGLPA) doit être autorisé APF.

Pour qu'ISPF crée la passerelle client TSO/ISPF, le module ISPZTS0 dans SYS1.LINKLIB doit être autorisé APF. La passerelle client TSO/ISPF est utilisée par le service Commandes de Developer for System z', SCLM Developer Toolkit et éventuellement CARMA.

Les autorisations de l'APF sont définies dans SYS1.PARMLIB(PROGxx), si votre site se conforme aux recommandations IBM.

Les autorisations APF peuvent être définies de manière dynamique (jusqu'au prochain démarrage système) à l'aide des commandes de la console suivantes où volser correspond au volume sur lequel le fichier se trouve s'il n'est pas géré par SMS :

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

#### Remarques :

1. Lorsque vous utilisez le module Alternate Library for REXX, le nom de la bibliothèque d'exécution REXX par défaut est REXX.\*.SEAGALT au lieu de REXX.\*.SEAGLPA, comme utilisé dans l'exemple précédent.
2. Les bibliothèques LPA, telles que REXX.\*.SEAGLPA, sont automatiquement autorisées APF lorsqu'elles se trouvent dans la zone LPA et elles ne requièrent donc pas de définitions explicites.
3. Certains des produits corequis (IBM Debug Tool, par exemple) requièrent également l'autorisation d'APF. Pour plus d'informations, reportez-vous aux guides de personnalisation de produit associés.

## Définitions LINKLIST dans PROGxx

Les définitions LINKLIST de Developer for System z peuvent être regroupées en 3 catégories :

- Les bibliothèques de chargement Developer for System z dont ont besoin les fonctions de Developer for System z. Ces définitions sont décrites dans la présente section.
- Les bibliothèques de chargement requises pour les fonctions de Developer for System z. Ces définitions sont présentées dans «Définitions LINKLIST et LPA prérequis», à la page 20.
- Les bibliothèques de chargement Developer for System z dont ont besoin les autres produits. Ces définitions sont présentées dans «Définitions LINKLIST pour les autres produits», à la page 21.

Pour que le service SCLM Developer Toolkit (facultatif) fonctionne, tous les modules BWB\* des bibliothèques de chargement FEK.SFEKAUTH et FEK.SFEKLOAD doivent être disponibles via STEPLIB ou LINKLIST.

Si vous optez pour l'utilisation de STEPLIB, vous devez définir les bibliothèques non disponibles via LINKLIST dans la directive STEPLIB du fichier de configuration RSE rsed.envvars. Gardez toutefois les remarques suivantes à l'esprit :

- L'utilisation de STEPLIB dans z/OS UNIX a un impact négatif sur les performances.
- Si une bibliothèque STEPLIB est autorisée par APF, il doit en être de même pour toutes les bibliothèques. Les bibliothèques perdent leur autorisation APF lorsqu'elles sont mélangées avec des bibliothèques non autorisées dans STEPLIB.

Les fichiers LINKLIST sont définis dans SYS1.PARMLIB(PROGxx), si votre site se conforme aux recommandations IBM.

Les définitions requises se présentent comme suit, où nomListe correspond au nom du fichier LINKLIST qui va être activé et volser au volume dans lequel se trouve le fichier s'il n'est pas placé dans le catalogue maître :

- LNKLIST ADD NAME(nomListe) DSNAME(FEK.SFEKAUTH) VOLUME(volser)
- LNKLIST ADD NAME(nomListe) DSNAME(FEK.SFEKLOAD)

Les définitions LINKLIST peuvent être créées de manière dynamique (jusqu'au prochain démarrage du système) à l'aide du groupe de commandes de console suivant, listname étant le nom du fichier LINKLIST en cours et volser étant le volume dans lequel réside le fichier s'il n'est pas placé dans la catalogue maître :

1. LNKLIST DEFINE,NAME=LLTMP,COPYFROM=CURRENT
2. LNKLIST ADD NAME=LLTMP,DSN=FEK.SFEKAUTH,VOL=volser
3. LNKLIST ADD NAME=LLTMP,DSN=FEK.SFEKLOAD
4. LNKLIST ACTIVATE,NAME=LLTMP
5. LNKLIST UNDEFINE,NAME=listname
6. LNKLIST UPDATE,JOB=\*

## Définitions LINKLIST et LPA prérequis

L'Explorateur de systèmes distants (RSE) est un processus z/OS UNIX qui requiert l'accès aux bibliothèques de chargement MVS. Les bibliothèques (prérequis) suivantes doivent être disponibles, via STEPLIB ou LINKLIST/LPALIB :

- Bibliothèque de chargement système
  - SYS1.LINKLIB
- Environnement d'exécution Language Environment
  - CEE.SCEERUN
  - CEE.SCEERUN2
- Bibliothèque de classes DLL C++
  - CBC.SCLBDLL
- Passerelle client TSO/ISPF d'ISPF
  - ISP.SISPLoad
  - ISP.SISPLPA

Les bibliothèques supplémentaires suivantes doivent être disponibles via STEPLIB ou LINKLIST/LPALIB pour la prise en charge des services facultatifs. Cette liste n'inclut pas les fichiers spécifiques d'un produit avec lequel interagit Developer for System z (IBM Debug Tool, par exemple) :

- Bibliothèque d'exécution REXX (pour SCLM Developer Toolkit)
  - REXX.\*.SEAGLPA
- Bibliothèque de chargement système (pour le chiffrement SSL)
  - SYS1.SIEALNKE
- Bibliothèque de charge TCP/IP (lorsque APPC est utilisé pour le service Commandes TSO)
  - TCPIP.SEZALOAD

### Remarques :

1. Lorsque vous utilisez le module Alternate Library for REXX, le nom de la bibliothèque d'exécution REXX par défaut est REXX.\*.SEAGALT au lieu de REXX.\*.SEAGLPA, comme utilisé dans l'exemple précédent.
2. Les bibliothèques sont conçues pour le positionnement LSA, telles que REXX.\*.SEAGLPA, peuvent requérir des autorisations de contrôle par programme et/ou APF supplémentaires si l'utilisateur y accède via LINKLIST ou STEPLIB.
3. Certains des produits corequis (IBM Debug Tool, par exemple) requièrent également des définitions STEPLIB ou LINKLIST/LPALIB. Pour plus d'informations, reportez-vous aux guides de personnalisation de produit associés.

4. Si CEE.SCEELKED se trouve dans LINKLIST ou STEPLIB, TCP/IP.SEZALOAD doit être placé avant CEE.SCEELKED. Tout manquement à cette règle entraîne une fin anormale du système 0C1 pour les appels de prise TCP/IP REXX.

Les fichiers LINKLIST sont définis dans SYS1.PARMLIB(PROGxx), si votre site se conforme aux recommandations IBM. Les fichiers LPA sont définis dans SYS1.PARMLIB(LPALSTxx).

Si vous optez pour l'utilisation de STEPLIB, vous devez définir les bibliothèques non disponibles via LINKLIST/LPALIB dans la directive STEPLIB du fichier de configuration RSE rsed.envvars. Gardez toutefois les remarques suivantes à l'esprit :

- L'utilisation de STEPLIB dans z/OS UNIX a un impact négatif sur les performances.
- Si une bibliothèque STEPLIB est autorisée par APF, il doit en être de même pour toutes les bibliothèques. Les bibliothèques perdent leur autorisation APF lorsqu'elles sont mélangées avec des bibliothèques non autorisées dans STEPLIB.
- Les bibliothèques ajoutées à STEPLIB DD en langage JCL ne sont pas transmises aux processus z/OS UNIX démarrés par JCL.

## Définitions LINKLIST pour les autres produits

Le client Developer for System z contient un composant de génération de code appelé Enterprise Service Tools (EST). Pour que le code généré émette des messages d'erreur de diagnostic, tous les modules IRZ\* et IIRZ\* de la bibliothèque de charge FEK.SFEKLOAD doivent être disponibles par l'intermédiaire de STEPLIB ou de LINKLIST.

Les fichiers LINKLIST sont définis dans SYS1.PARMLIB(PROGxx), si votre site se conforme aux recommandations IBM.

Si vous optez pour l'utilisation de STEPLIB, vous devez définir les bibliothèques non disponibles via LINKLIST dans la directive STEPLIB de la tâche qui exécute le code (IMS ou le travail par lots). Toutefois, n'oubliez pas que :

- Si une bibliothèque STEPLIB est autorisée par APF, il doit en être de même pour toutes les bibliothèques. Les bibliothèques perdent leur autorisation APF lorsqu'elles sont mélangées avec des bibliothèques non autorisées dans STEPLIB.

---

## Modifications de PROCLIB

Les procédures de tâche démarrée et de génération à distance répertoriées ci-dessous doivent résider dans une bibliothèque de procédures système définie pour votre sous-système JES. Dans les instructions ci-dessous, la bibliothèque de procédures par défaut IBM, SYS1.PROCLIB, est utilisée.

## Moniteur de travaux JES

Personnalisez l'exemple de membre de tâche démarrée, FEK.#CUST.PROCLIB(JMON), comme décrit dans le membre, et copiez-le dans SYS1.PROCLIB. Comme indiqué dans l'exemple de code suivant, vous devez spécifier les éléments suivants :

- Le qualificatif de haut niveau de la bibliothèque de chargement (autorisée), par défaut FEK
- Le fichier de configuration du moniteur de travaux JES, par défaut FEK.#CUST.PARMLIB(FEJJCNFG)

```

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

```

Figure 1. JMON - Tâche démarrée du moniteur de travaux JES

#### Remarques :

1. Pour plus d'informations sur les paramètres de démarrage, voir Chapitre 8, «Commandes de l'opérateur», à la page 127.
2. Cet exemple de fichier JCL est initialement fourni sous le nom FEK.SFEKSAMP(FEJJJCL) et est renommé FEK.#CUST.PROCLIB(JMON) lors de l'étape «Configuration personnalisée», à la page 15.
3. Le traçage peut également être contrôlé à l'aide des commandes de la console, comme décrit à la section Chapitre 8, «Commandes de l'opérateur», à la page 127.
4. Cette tâche doit être attribuée à SYSSTC ou à un objectif équivalent dans WLM.
5. La variable d'environnement LE\_CEE\_ENVFILE\_S nécessite z/OS 1.8 ou supérieur. Il est possible de substituer la variable par \_CEE\_ENVFILE sur des niveaux plus anciens de z/OS, sauf qu'en raison d'un bogue dans l'exécution C, la variable TZ du fichier de configuration du moniteur de travaux JES (FEJJCNFG) pourrait ne pas être interprétée correctement.

## Démon RSE

Personnalisez l'exemple de membre de tâche démarrée, FEK.#CUST.PROCLIB(RSED), comme décrit dans le membre, et copiez-le dans SYS1.PROCLIB. Comme indiqué dans l'exemple de code suivant, vous devez spécifier les éléments suivants :

- Le port du démon RSE, par défaut, 4035.
- Le répertoire de base où Developer for System z est installé, par défaut /usr/lpp/rdz.
- L'emplacement des fichiers de configuration, par défaut, /etc/rdz

```

/*
/* RSE DAEMON
/*
//RSED      PROC IVP='',                      * 'IVP' to do an IVP test
//          PORT=4035,
//          HOME='/usr/lpp/rdz',
//          CNFG='/etc/rdz'
/*
//RSE      EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
//          PARM='PGM &HOME/bin/rsed.sh &IVP &PORT &CNFG'
//STDERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//          PEND
/*

```

Figure 2. RSED - Tâche démarrée du serveur RSE

#### Remarque :

- Pour plus d'informations sur les paramètres de démarrage, voir Chapitre 8, «Commandes de l'opérateur», à la page 127.
- Cet exemple de fichier JCL est initialement fourni sous le nom FEK.SFEKSAMP(FEKRSED) et est renommé FEK.#CUST.PROCLIB(RSED) lors de l'étape «Configuration personnalisée», à la page 15.
- Limitez le nom du travail à 7 caractères maximum. Si un nom de 8 caractères est utilisé, les commandes opérateur **modify** et **stop** échouent en générant le message "IEE342I MODIFY REJECTED-TASK BUSY". Ce comportement est dû à la conception de z/OS UNIX pour les processus enfant.
- Cette tâche, et les processus enfant qu'elle crée, doivent être attribués à SYSSTC ou à un objectif équivalent dans WLM. Les processus enfant portent le même nom que la tâche parent (RSED) suivi d'un chiffre aléatoire (RSED8).

## Démon lock

Personnalisez l'exemple de membre de tâche démarrée, FEK.#CUST.PROCLIB(LOCKD), comme indiqué dans le membre, et copiez-le dans SYS1.PROCLIB. Comme indiqué dans l'exemple de code suivant, vous devez spécifier les éléments suivants :

- Le répertoire de base dans lequel Developer for System z est installé (/usr/lpp/rdz, par défaut)
- L'emplacement des fichiers de configuration, par défaut /etc/rdz
- Le niveau de consignation initial, par défaut 1.

```

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

```

Figure 3. LOCKD - Tâche démarrée du démon lock



#### Remarques :

1. Pour plus d'informations sur les paramètres de démarrage, voir Chapitre 8, «Commandes de l'opérateur», à la page 127.
2. Cet exemple de fichier JCL est initialement fourni sous le nom FEK.SFEKSAMP(FEKLCKD) et est renommé FEK.#CUST.PROCLIB(LOCKD) lors de l'étape «Configuration personnalisée», à la page 15.
3. Cette tâche doit être attribuée à SYSSTC ou à un objectif équivalent dans WLM.

## Limitations JCL pour la variable PARM

La longueur maximale de la variable PARM est de 100 caractères, longueur qui peut engendrer des problèmes si vous utilisez des noms de répertoire personnalisés. Pour contourner ce problème, vous pouvez :

- Utiliser des liens symboliques

Les liens symboliques peuvent être utilisés comme abréviation pour un nom de répertoire long. L'exemple de commande z/OS UNIX suivant définit un lien symbolique (/usr/lpp/rdz) vers un autre répertoire (/long/directory/name/usr/lpp/rdz).

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

- Utiliser STDIN

Lorsque la zone PARM est vide, **BPXBATSL** lance un shell z/OS UNIX et exécute le script de shell fourni par STDIN. Notez que STDIN doit être un fichier z/OS UNIX (alloué en tant que ORDONLY) et que son utilisation désactive les variables ROC pour le port etc. En outre, le shell doit exécuter les scripts de connexion du shell /etc/profile et \$HOME/.profile.

Pour utiliser cette méthode, vous devez d'abord mettre à jour le code JCL pour qu'il corresponde à l'exemple suivant :

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

Figure 4. RSED - Démarrage alternatif du démon RSE

Vous devez ensuite créer le script de shell (/etc/rdz/rsed.stdin.sh dans le présent exemple) qui démarrera le démon RSE. Le contenu de ce script ressemblera à l'exemple suivant :

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

Figure 5. rsed.stdin.sh - Démarrage alternatif du démon RSE

**Remarque :** Il est recommandé d'allouer rsed.envvars à STDENV dans le code JCL de démarrage du démon car il définit certaines directives z/OS UNIX qui permettent d'économiser les ressources système lorsque vous utilisez cette méthode de démarrage.



## Procédures de construction à distance ELAXF\*

Developer for System z fournit des exemples de procédures JCL qui peuvent être utilisés lors de la construction du code JCL, de la génération de projets distants et pour les fonctions de vérification syntaxique à distance des mappes BMS CICS, des écrans MFS IMS et des programmes COBOL, PL/I, Assembler et C/C++. Ces procédures permettent aux installations d'appliquer leurs propres normes et garantissent que les développeurs utilisent les mêmes procédures, options de compilation et niveaux de compilateur.

Les exemples de procédures et leurs fonctions sont répertoriées dans le tableau 7.

Tableau 7. Modèles de procédure ELAXF\*

Membre	Fonction
ELAXFADT	Modèle de procédure pour l'assemblage et le débogage des programmes Assembleur de haut niveau.
ELAXFASM	Modèle de procédure pour l'assemblage des programmes Assembleur de haut niveau.
ELAXFBMS	Modèle de procédure de création d'un objet BMS CICS BMS et de sa copie correspondante, dsect, ou du membre d'inclusion.
ELAXFCOC	Modèle de procédure pour l'exécution de compilations COBOL, de traductions CICS et DB2 intégrées.
ELAXFCOP	Modèle de procédure pour l'exécution du pré-processus DB2 des instructions SQL EXEC intégrées dans des programmes COBOL.
ELAXFCOT	Modèle de procédure pour l'exécution d'une traduction CICS des instructions CICS EXEC intégrées dans des programmes COBOL.
ELAXFCPC	Modèle de procédure pour l'exécution de compilations C.
ELAXFCPP	Modèle de procédure pour l'exécution de compilations C++.
ELAXFCP1	Exemple de procédure pour les compilations COBOL avec des instructions de préprocesseur SCM (-INC et ++INCLUDE).
ELAXFDCL	Exemple de procédure d'exécution d'un programme en mode TSO.
ELAXFGO	Modèle de procédure pour l'étape GO.
ELAXFLNK	Modèle de procédure pour la liaison des programmes C/C++, COBOL, PLI et Assembleur de haut niveau.
ELAXFMFS	Modèle de procédure pour la création d'écrans IMS MFS.
ELAXFPLP	Modèle de procédure pour l'exécution du pré-processus DB2 des instructions SQL EXEC intégrées dans des programmes PLI.
ELAXFPLT	Modèle de procédure pour l'exécution d'une conversion CICS des instructions CICS EXEC intégrées dans des programmes PLI.
ELAXFPL1	Modèle de procédure pour l'exécution de compilations PL/I, de traductions CICS et DB2 intégrées.
ELAXFPP1	Exemple de procédure pour les compilations PL/I avec des instructions de préprocesseur SCM (-INC and ++INCLUDE).
ELAXFTSO	Exemple de procédure pour exécuter/déboguer le code DB2généralisé en mode TSO.
ELAXFUOP	Modèle de procédure pour générer l'étape UOPT lors de la création de programmes de génération s'exécutant dans CICS ou des sous-systèmes IMS.

Les noms des procédures et les noms des étapes des procédures correspondent aux propriétés par défaut livrées avec le client Developer for System z. Si vous décidez

de modifier le nom d'une procédure ou le nom d'une étape de la procédure, le fichier des propriétés correspondantes des clients doit également être mis à jour. Nous recommandons de ne pas modifier les noms de procédure et d'étape.

Personnalisez les exemples de membres de procédure de génération, FEK.#CUST.PROCLIB(ELAXF\*), comme décrit dans les membres, et copiez-les dans SYS1.PROCLIB. Vous devez fournir les qualificatifs de haut niveau appropriés des différentes bibliothèques de produits, comme décrit dans le tableau 8.

*Tableau 8. Liste de contrôle des qualificatifs de haut niveau ELAXF\**

Produit	Valeur par défaut HLQ	Valeur
Developer for System z	FEK	
CICS	CICSTS32.CICS	
DB2	DSN910	
IMS	IMS	
COBOL	IGY.V4R1M0	
PL/I	IBMZ.V3R8M0	
C/C++	CBC	
LE	CEE	
système LINKLIB	SYS1	
système MACLIB	SYS1	

Si les procédures ELAXF\* ne peuvent pas être copiées dans une bibliothèque de procédures système, demandez aux utilisateurs de Developer for System z d'ajouter une carte JCLLIB (tout de suite après la carte JOB) aux propriétés du travail sur le client.

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

## Définitions de sécurité

Personnalisez et soumettez l'exemple de membre FEKRACF dans le fichier FEK.#CUST.JCL en vue de créer les définitions de sécurité pour Developer for System z. L'utilisateur qui soumet le travail doit disposer des privilèges d'administrateur de la sécurité (RACF SPECIAL, par exemple).

### Remarque :

- Pour les sites qui utilisent CA ACF2™ for z/OS, veuillez consulter le lien <https://support.ca.com/irj/portal/kbtech?ipLogNrow=0&docid=492389&searchID=TEC492389> pour obtenir des détails sur les commandes de sécurité nécessaires à la configuration correcte de Developer for System z.
- Pour les sites qui utilisent CA Top Secret for z/OS, veuillez consulter la page de votre produit du site de support CA (<https://support.ca.com>), puis le document Developer for System z connexe. Ce document présente les commandes de sécurité nécessaires à la configuration correcte de Developer for System z.

La liste des définitions obligatoires associées à la sécurité pour Developer for System z est décrite en détail dans le Chapitre 10, «Remarques relatives à la sécurité», à la page 161. Ce chapitre traite également de la sécurité générale

concernant Developer for System z, notamment des aspects relatifs à la sécurité des produits requis non couverts dans l'exemple de travail FEKRACF.

- Activation des paramètres et des classes de sécurité
- Définition d'un segment OMVS pour les utilisateurs Developer for System z
- Définition des profils de fichier
- Définition des tâches démarrées JMON, RSED et LOCKD
- Définition de la sécurité de commande JES
- Définition de RSE comme serveur z/OS UNIX sécurisé
- Définition des bibliothèques contrôlées par programme MVS pour RSE
- Définition de la sécurité d'application pour RSE
- Définition de la prise en charge de PassTicket pour RSE
- Définition des fichiers contrôlés par programme z/OS UNIX pour RSE

**Remarque :** L'exemple de travail FEKRACF ne contient pas seulement des commandes RACF. La dernière étape des définitions de sécurité consiste à créer un fichier z/OS UNIX contrôlé par programme. Suivant les règles en vigueur sur votre site, cette tâche relèvera du programmeur système et non de l'administrateur système.

<p><b>Avertissement :</b> La demande de connexion du client n'aboutit pas si la sécurité de l'application et les mots de passe PassTickets ne sont pas correctement configurés.</p>
---

---

## FEJJCNFG, fichier de configuration du moniteur de travaux JES

Le moniteur de travaux JES (JMON) offre tous les services liés à JES. Son comportement peut être contrôlé à l'aide des définitions indiquées dans FEJJCNFG.

FEJJCNFG se trouve dans FEK.#CUST.PARMLIB, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Personnalisez l'exemple de membre de configuration du moniteur de travaux JES FEJJCNFG, comme indiqué dans l'exemple suivant. Les lignes de commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne.

**Remarque :** Vous devez redémarrer la tâche démarrée JMON pour appliquer les modifications apportées.

```

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

```

Figure 6. FEJCNFG, fichier de configuration du moniteur de travaux JES

## SERV\_PORT

Le numéro de port du serveur hôte du moniteur de travaux JES. Le port par défaut est 6715. Vous pouvez le modifier mais le serveur et les clients Developer for System z doivent TOUS DEUX être configurés avec le même numéro de port. Si vous modifiez le numéro de port du serveur, tous les clients doivent modifier le port du moniteur de travaux JES de ce système dans la vue Systèmes distants.

### Remarque :

- Avant de sélectionner un port, vérifiez qu'il est disponible sur votre système à l'aide des commandes TSO **NETSTAT** et **NETSTAT PORTL**.
- Lors de l'utilisation d'un client dont la version est 7.1 ou supérieure, toutes les communications sur ce port ne concernent que votre machine hôte z/OS.

**TZ** Sélecteur de fuseau horaire. La valeur par défaut est EST5EDT. Le fuseau horaire par défaut est le temps universel coordonné + 5 heures (heure d'été de la côte Est). Modifiez cette valeur pour afficher votre fuseau horaire. Pour plus d'informations, reportez-vous au document *UNIX System Services Command Reference* (SA22-7802).

Les définitions suivantes sont facultatives. Si vous les omettez, les valeurs par défaut seront utilisées comme indiqué ci-dessous :

## \_BPXK\_SETIBMOPT\_TRANSPORT

Indique le nom de la pile TCPIP à utiliser. La valeur par défaut est TCPIP. Supprimez la mise en commentaire et remplacez par le nom de la pile TCPIP demandée, comme défini dans l'instruction TCPIPJOBNAME du fichier TCPIP.DATA associé.

### Remarque :

- Le codage d'une instruction SYSTCPD DD dans le langage de contrôle des travaux du serveur ne définit pas l'affinité de pile demandée.

- Si cette directive n'est pas active, le moniteur de travaux JES se relie à toutes les piles disponibles sur le système (BIND INADDRANY).

## APPLID

Indique l'identificateur de l'application utilisé pour identifier le moniteur de travaux JES auprès de votre logiciel de sécurité. La valeur par défaut est FEKAPPL. Supprimez la mise en commentaire et modifiez l'ID de l'application de votre choix.

**Remarque :** Cette valeur doit correspondre à l'ID de l'application défini pour RSE dans le fichier de configuration rsed.envvars. Si ces valeurs ne sont pas identiques, RSE ne peut pas connecter le client au moniteur de travaux JES.

## AUTHMETHOD

La valeur par défaut est SAF, ce qui signifie que l'interface de sécurité SAF (System Authorization Facility) est utilisée. Ne pas la modifier sauf recommandation explicite du point service IBM.

## CODEPAGE

Page de code du poste de travail. La valeur par défaut est UTF-8. La page de code du poste de travail est définie sur UTF-8 et ne doit généralement pas être modifiée. Vous devrez peut-être supprimer la mise en commentaire et modifier la page de codes UTF-8 pour qu'elle corresponde à la page de codes du poste de travail si des caractères NLS, tels que le symbole monétaire, posent problème.

## CONCHAR

Indique le caractère de commande de la console du moniteur de travaux JES. CONCHAR a pour valeur par défaut CONCHAR=\$ pour JES2, ou CONCHAR=\* pour JES3. Supprimez la mise en commentaire et modifiez par le caractère de commande demandé.

## CONSOLE\_NAME

Indique le nom de la console EMCS utilisée pour lancer des commandes sur des travaux (Mettre en attente, Publier, Annuler et Purger). La valeur par défaut est JMON. Supprimez la mise en commentaire et modifiez le nom de la console de votre choix en suivant les recommandations ci-après.

- CONSOLE\_NAME doit être un nom de console composé de 2 à 8 caractères alphanumériques ou '&SYSUID' (sans apostrophes).
- Si un nom de console est indiqué, une seule console portant ce nom est utilisée pour tous les utilisateurs. Si la console dont le nom est indiqué est en cours d'utilisation, la commande exécutée par le client échoue.
- Si &SYSUID est indiqué, l'ID utilisateur du client est utilisé comme nom de console. Une console différente est donc utilisée pour chaque utilisateur. Si la console dont le nom est indiqué est en cours d'utilisation (par exemple, un utilisateur utilise SDSF ULOG), la commande exécutée par le client risque d'échouer, en fonction du paramètre GEN\_CONSOLE\_NAME.

Quel que soit le nom de console utilisé, l'ID utilisateur du client qui demande la commande est utilisé en tant qu'unité logique de la console et est consigné dans les messages syslog IEA630I et IEA631I.

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

## GEN\_CONSOLE\_NAME

Active ou désactive la génération automatique de noms de console

alternatifs. La valeur par défaut est OFF. Supprimez la mise en commentaire et remplacez la valeur par ON pour activer les noms de console alternatifs.

Cette directive est utilisée uniquement lorsque `CONSOLE_NAME` égale `&SYSUID` et que l'ID utilisateur n'est pas disponible en tant que nom de console.

Si `GEN_CONSOLE_NAME=ON`, un nom de console alternatif est généré en ajoutant une seule valeur numérique à l'ID utilisateur. Des tentatives sont effectuées en utilisant des valeurs comprises 0 à 9. Si aucune console disponible n'est trouvée, la commande exécutée par le client échoue.

Si `GEN_CONSOLE_NAME=OFF`, la commande exécutée par le client échoue.

**Remarque :** Les seuls paramètres valides sont ON et OFF.

## HOST\_CODEPAGE

Page de code hôte. La valeur par défaut est IBM-1047. Supprimez la mise en commentaire et modifiez cette valeur en fonction de votre page de codes hôte.

A partir de la version 7.6.1 comprise, les clients Developer for System z ignorent la valeur `HOST_CODEPAGE` spécifiée ici et utilisent la page de codes spécifiée localement dans les propriétés du sous-système de "fichiers MVS".

**Remarque :** Même pour des clients récents, un moniteur de travaux JES utilisera la page de codes hôte spécifiée dans `HOST_CODEPAGE` au cours de la configuration de la communication client initiale.

## LIMIT\_COMMANDS

Définir les travaux auxquels l'utilisateur peut appliquer les commandes JES sélectionnées (Afficher JCL, Mettre en attente, Publier, Annuler et Purger). La valeur par défaut (`LIMIT_COMMANDS=USERID`) limite les commandes aux travaux dont l'utilisateur est le propriétaire. Supprimez la mise en commentaire de cette directive et spécifiez `LIMITED` ou `NOLIMIT` pour permettre à l'utilisateur d'émettre les commandes sur des fichiers spoule, si votre produit de sécurité l'autorise.

Tableau 9. Matrice des droits d'accès des commandes `LIMIT_COMMANDS`

LIMIT_COMMANDS	Propriétaire du travail	
	Utilisateur	Autre
USERID (valeur par défaut)	Autorisé	Non autorisé
LIMITED	Autorisé	Autorisé uniquement si permis de manière explicite par les profils de sécurité
NOLIMIT	Autorisé	Autorisé si les profils de sécurité l'acceptent ou lorsque la classe JESSPOOL n'est pas active

**Remarque :** Les seuls paramètres acceptés sont USERID, LIMITED et NOLIMIT.

## LIMIT\_VIEW

Définissez la sortie à visualiser par l'utilisateur. La valeur par défaut (`LIMIT_VIEW=NOLIMIT`) permet à l'utilisateur d'afficher tous les résultats JES,

si votre produit de sécurité l'autorise. Supprimez la mise en commentaire de cette directive et spécifiez USERID pour limiter l'affichage aux résultats dont l'utilisateur est le propriétaire.

**Remarque :** Les seuls paramètres valides sont USERID et NOLIMIT.

#### **LISTEN\_QUEUE\_LENGTH**

Longueur de file d'attente d'écoute TCP/IP. La valeur par défaut est 5. Ne pas la modifier sauf recommandation explicite du point service IBM.

#### **MAX\_DATASETS**

Nombre maximal de fichiers de sortie en spoule renvoyés au client par le moniteur de travaux JES (par exemple, SYSOUT, SYSPRINT, SYS00001, etc.). La valeur par défaut est 32. La valeur maximale est 2147483647.

#### **MAX\_THREADS**

Nombre maximal d'utilisateurs qui peuvent utiliser simultanément un moniteur de travaux JES. La valeur par défaut est 200. La valeur maximale est 2147483647. Si vous augmentez cette valeur, vous devez augmenter la taille de l'espace adresse du moniteur de travaux JES.

#### **TIMEOUT**

Durée, en secondes, avant l'arrêt d'une unité d'exécution, dû à l'absence d'interaction avec le client. La valeur par défaut est 3600 (1 heure). La valeur maximale est 2147483647. La valeur TIMEOUT=0 désactive la fonction.

#### **TIMEOUT\_INTERVAL**

Nombre de secondes entre les vérifications de dépassement de délai. La valeur par défaut est 1200. La valeur maximale est 2147483647.

#### **SUBMITMETHOD=TSO**

Soumission de travaux via TSO. La valeur par défaut (SUBMITMETHOD=JES) permet de soumettre des travaux directement au moniteur JES. Supprimez la mise en commentaire de cette directive et spécifiez TSO afin de soumettre le travail via la commande TSO **SUBMIT**. Cette méthode permet d'appeler des sorties TSO ; toutefois, elle présente des inconvénients en termes de performances et n'est pas recommandée pour cette raison.

**Remarque :**

- Les seuls paramètres valides sont TSO et JES.
- Si vous spécifiez SUBMITMETHOD=TSO, vous devez également définir TSO\_TEMPLATE.

#### **TSO\_TEMPLATE**

JCL de l'encapsuleur pour la soumission de travaux TSO. La valeur par défaut est FEK.#CUST.CNTL(FEJTSO). Cette instruction indique le nom de membre complet du JCL à utiliser comme encapsuleur pour la soumission TSO. Pour plus d'informations, voir l'instruction SUBMITMETHOD.

**Remarque :**

- Un exemple de travail d'encapsuleur est fourni dans FEK.#CUST.CNTL(FEJTSO). Consultez ce membre pour plus d'informations sur la personnalisation nécessaire.
- TSO\_TEMPLATE n'a aucun effet si SUBMITMETHOD=TSO n'est pas également spécifié.



---

## rsed.envvars - Fichier de configuration RSE

Les processus du démon lock RSE et duserveur RSE (démon RSE, pool d'unités d'exécution RSE et serveur RSE) utilisent les définitions figurant dans le fichier `rsed.envvars`. Le composant Developer for System z facultatif et les services tiers peuvent également recourir à ce fichier de configuration en vue de définir des variables à utiliser.

L'Explorateur de systèmes distants (RSE) fournit des services de base, comme la connexion du client à l'hôte et le démarrage d'autres serveurs pour des services spécifiques. Le démon lock fournit des services de suivi pour les verrous de fichiers.

Le fichier `rsed.envvars` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**.

Consultez l'exemple de fichier `rsed.envvars` suivant qui doit être personnalisé pour correspondre à votre environnement système. Les lignes de commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne. Les continuations de ligne et les espaces autour du signe égal (=) ne sont pas pris en charge.

**Remarque :** Vous devez redémarrer les tâches démarrées `RSED` et `LOCKD` pour appliquer les modifications apportées.





```

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

```

Figure 7. rsed.envvars - Fichier de configuration RSE

```

# (5) do not change unless directed by IBM support center
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_BPX_SHAREAS=YES
_BPX_SPAWN_SCRIPT=YES
JAVA_PROPAGATE=NO
RSE_LIB=$RSE_HOME/lib
PATH=.:$JAVA_HOME/bin:$RSE_HOME/bin:$CMDSESV_BASE_HOME/bin:$PATH
LIBPATH=$JAVA_HOME/bin:$JAVA_HOME/bin/classic:$RSE_LIB:$RSE_LIB/icuc
LIBPATH=.:usr/lib:$LIBPATH
CLASSPATH=$RSE_LIB:$RSE_LIB/dstore_core.jar:$RSE_LIB/clientserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_extra_server.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/zosserver.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/dstore_miners.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/universalminers.jar:$RSE_LIB/mvsminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/carma.jar:$RSE_LIB/luceneminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsluceneminer.jar:$RSE_LIB/cdzminer.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvscdzminer.jar:$RSE_LIB/jesminers.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/FAMiner.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/mvsutil.jar:$RSE_LIB/jesutils.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/lucene-core-2.3.2.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/cdtparser.jar
CLASSPATH=$CLASSPATH:$RSE_LIB/wdzbidi.jar:$RSE_LIB/fmiExtensions.jar
CLASSPATH=$CLASSPATH:$RSE_SAF_CLASS
CLASSPATH=.:$CLASSPATH
_RSE_CMDSESV_OPTS="&SESSION=SPAWN$ RSE_CMDSESV_OPTS"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DISPF_OPTS='$_RSE_CMDSESV_OPTS'"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DA_PLUGIN_PATH=$RSE_LIB"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xbootclasspath/p:$RSE_LIB/bidiTools.jar"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dfile.encoding=$_RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dconsole.encoding=$_RSE_HOST_CODEPAGE"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_SPIRIT_ON=true"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DSPIRIT_EXPIRY_TIME=6"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DSPIRIT_INTERVAL_TIME=6"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dcom.ibm.cacheLocalHost=true"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Duser.home=$HOME"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dclient.username=$RSE_USER_ID"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlow.heap.usage.ratio=15"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dmaximum.heap.usage.ratio=40"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_KEEPALIVE_ENABLED=true"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_KEEPALIVE_RESPONSE_TIMEOUT=30000"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_IO_SOCKET_READ_TIMEOUT=90000"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DRSECOMM_LOGFILE_MAX=0"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlock.daemon.port=$_RSE_LOCKD_PORT"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dlock.daemon.cleanup.interval=1440"
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -showversion"
_RSE_SERVER_CLASS=org.eclipse.dstore.core.server.Server
_RSE_DAEMON_CLASS=com.ibm.etools.zos.server.RseDaemon
_RSE_POOL_SERVER_CLASS=com.ibm.etools.zos.server.ThreadPoolProcess
_RSE_LOCKD_CLASS=com.ibm.ftt.rse.mvs.server.miners.MVSLockDaemon
_RSE_SERVER_TIMEOUT=120000
_SCLMDT_BASE_HOME=$RSE_HOME
_SCLMDT_WORK_HOME=$CMDSESV_WORK_HOME
CGI_DTWORK=$_SCLMDT_WORK_HOME
#=====
# (6) additional environment variables

```

Figure 8. (continued)

**Remarque :** Les liens symboliques sont autorisés pour l'indication des répertoires dans `rsed.envvars`.

Les définitions suivantes sont requises :

**JAVA\_HOME**

Répertoire de base Java. Le répertoire par défaut est `/usr/lpp/java/J5.0`. Modifiez en fonction de votre installation Java.

**RSE\_HOME**

Répertoire de base RSE. Le répertoire par défaut est `/usr/lpp/rdz`. Modifiez en fonction de votre installation de Developer for System z.

**\_RSE\_LOCKD\_PORT**

Numéro de port du démon lock RSE. La valeur par défaut est 4036. Peut être modifiée au besoin.

**Remarque :**

- Avant de sélectionner un port, vérifiez qu'il est disponible sur votre système à l'aide des commandes TSO **NETSTAT** et **NETSTAT PORTL**.
- Toutes les communications de ce port se limitent au système hôte z/OS.

**\_RSE\_HOST\_CODEPAGE**

Page de code hôte. La valeur par défaut est IBM-1047. Modifiez cette valeur en fonction de votre page de codes hôte.

**TZ**

Sélecteur de fuseau horaire. La valeur par défaut est EST5EDT. Le fuseau horaire par défaut est le temps universel coordonné + 5 heures (heure d'été de la côte Est). Modifiez cette valeur pour afficher votre fuseau horaire.

Pour plus d'informations, reportez-vous au document *UNIX System Services Command Reference* (SA22-7802).

**LANG**

Indique le nom des paramètres régionaux par défaut. Le nom par défaut est C. C indique les paramètres régionaux POSIX et (par exemple) Ja\_JP indique les paramètres régionaux japonais. Modifiez cette valeur pour afficher vos paramètres régionaux.

**PATH**

Chemin d'accès de la commande. La valeur par défaut est `/bin:/usr/sbin:..`. Peut être modifiée au besoin.

**\_CEE\_DMPTARG**

Emplacement d'exportation de Language Environment (LE) z/OS UNIX utilisé par la machine virtuelle Java (JVM). L'emplacement par défaut est `/tmp`.

**STEPLIB**

L'accès aux fichiers MVS ne figure pas dans LINKLIST/LPALIB. La valeur par défaut est NONE.

Vous pouvez supprimer le besoin de disposer de bibliothèques (prérequis) dans LINKLIST/LPALIB en supprimant la mise en commentaire et en personnalisant une ou plusieurs des directives STEPLIB suivantes. Voir «Modifications de PARMLIB», à la page 16 pour plus d'informations sur l'utilisation des bibliothèques répertoriées ci-après :

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

**Remarque :**

- L'utilisation de STEPLIB dans z/OS UNIX a un impact négatif sur les performances.
- Si une bibliothèque STEPLIB est autorisée par APF, il doit en être de même pour toutes les bibliothèques. Les bibliothèques perdent leur autorisation APF lorsqu'elles sont mélangées avec des bibliothèques non autorisées dans STEPLIB.
- Les bibliothèques conçues pour le placement LPA peuvent nécessiter un contrôle de programmes et des autorisations APF supplémentaires si leur accès est obtenu via LINKLIST ou STEPLIB.
- Le codage d'une instruction STEPLIB DD dans le langage de contrôle des travaux du serveur ne définit pas la concaténation STEPLIB demandée.

**RSE\_SAF\_CLASS**

Indique l'interface Java dans votre produit de sécurité. La valeur par défaut est `/usr/include/java_classes/IRRacf.jar`. Modifiez cette valeur pour qu'elle corresponde à votre configuration de logiciel de sécurité.

**Remarque :** Comme z/OS 1.10, `/usr/include/java_classes/IRRacf.jar` est un composant de SAF fourni avec le z/OS de base, il est également disponible pour les clients non RACF.

**RSE\_JAVAOPTS**

Options Java supplémentaires spécifiques de RSE. . Pour plus d'informations sur cette définition, voir «Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JAVAOPTS`», à la page 42.

Par défaut, Developer for System z utilise la passerelle client TSO/ISPF d'ISPF pour le service Commandes TSO. Une transaction APPC est utilisée à la place lorsque l'option `_RSE_JAVAOPTS` suivante est retirée de la mise en commentaire :

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

Les définitions suivantes sont nécessaires si la passerelle client TSO/ISPF d'ISPF est utilisée pour le service Commandes TSO, SCLM Developer Toolkit ou CARMA.

**\_CMDSERV\_BASE\_HOME**

Répertoire de base pour le code ISPF qui fournit le service de passerelle client TSO/ISPF. La valeur par défaut est `/usr/lpp/ispf`. Modifiez cette valeur en fonction de votre installation ISPF. Cette directive est requise uniquement lorsque la passerelle client TSO/ISPF d'ISPF est utilisée.

**\_CMDSERV\_CONF\_HOME**

Répertoire de configuration de base ISPF. La valeur par défaut est `/etc/rdz`. Modifiez cette valeur pour qu'elle corresponde à l'emplacement de `ISPF.conf`, le fichier de personnalisation de la passerelle client TSO/ISPF. Cette directive est requise uniquement lorsque la passerelle client TSO/ISPF d'ISPF est utilisée.

**\_CMDSERV\_WORK\_HOME**

Répertoire de travail de base ISPF. La valeur par défaut est `/var/rdz`. Modifiez cette valeur pour qu'elle corresponde à l'emplacement du répertoire WORKAREA utilisé par la passerelle client TSO/ISPF. Cette directive est requise uniquement lorsque la passerelle client TSO/ISPF d'ISPF est utilisée.

Remarques :

- La passerelle client TSO/ISPF va ajouter /WORKAREA au chemin spécifié dans \_CMDSEV\_WORK\_HOME. Ne l'ajoutez pas vous-même.
- Si vous n'avez pas utilisé le modèle de travail SFEKSAMP(FEKSETUP) pour générer l'environnement personnalisable, vérifiez que le répertoire WORKAREA existe dans le chemin spécifié dans \_CMDSEV\_WORK\_HOME. La valeur 777 doit être attribuée aux données de droit du répertoire.

#### **STEPLIB**

STEPLIB est décrit précédemment dans la section des définitions requises.

#### **RSE\_CMDSEV\_OPTS**

Options Java supplémentaires spécifiques de la passerelle client TSO/ISPF. La valeur par défaut est "". Pour plus d'informations sur cette définition, voir «Définition de paramètres de démarrage Java supplémentaires avec \_RSE\_CMDSEV\_OPTS», à la page 47. Cette directive est requise uniquement lorsque la passerelle client TSO/ISPF d'ISPF est utilisée.

Les définitions suivantes sont requises si SCLM Developer Toolkit est utilisé.

#### **SCLMDT\_CONF\_HOME**

Répertoire de configuration de base de SCLM Developer Toolkit. La valeur par défaut est /var/rdz/scldmt. Modifiez cette valeur pour qu'elle corresponde à l'emplacement du répertoire CONFIG utilisé par SCLMDT pour conserver les informations de projet SCLM. Cette directive est requise uniquement lorsque SCLMDT est utilisé.

**Remarque :** SCLMDT va ajouter /CONFIG et /CONFIG/PROJECT au chemin spécifié dans SCLMDT\_CONF\_HOME. Ne l'ajoutez pas vous-même.

#### **STEPLIB**

STEPLIB est décrit précédemment dans la section des définitions requises.

#### **\_SCLMDT\_TRANTABLE**

Nom de la méthode d'accès VSAM de conversion de noms courts/longs. Le nom par défaut est FEK.#CUST.LSTRANS.FILE. Supprimez la mise en commentaire et modifiez le nom pour qu'il corresponde au nom utilisé dans l'exemple de travail SCLM ISP.SISPSAMP(FLM02LST). Cette directive est requise uniquement si la conversion de noms longs/courts dans SCLM Developer Toolkit est utilisée.

#### **ANT\_HOME**

Répertoire de base de votre installation Ant. La valeur par défaut est /usr/lpp/apache/Ant/apache-ant-1.7.1. Modifiez le répertoire pour qu'il corresponde à celui de votre installation Ant. Cette directive est requise uniquement lorsque la prise en charge de la génération JAVA/J2EE est utilisée avec SCLM Developer Toolkit.

Les définitions suivantes sont facultatives. Si vous les omettez, les valeurs par défaut seront utilisées :

#### **\_RSE\_PORTRANGE**

Indique la plage de ports que le serveur RSE peut ouvrir pour communiquer avec un client. Chaque port peut être utilisé par défaut. Pour plus d'informations sur cette définition, voir «Définition de PORTRANGE disponible pour un serveur RSE», à la page 41. Il s'agit d'une directive facultative.

### **\_BPXK\_SETIBMOPT\_TRANSPORT**

Indique le nom de la pile TCP/IP à utiliser. La valeur par défaut est TCPIP. Supprimez la mise en commentaire et remplacez le nom par celui de la pile TCP/IP demandée, comme défini dans l'instruction TCPIPJOBNAME du fichier TCPIP.DATA associé. Il s'agit d'une directive facultative.

#### **Remarque :**

- Le codage d'une instruction SYSTCPD DD dans le langage de contrôle des travaux du serveur ne définit pas l'affinité de pile demandée.
- Si cette directive n'est pas active, RSE se relie à toutes les piles disponibles sur le système (BIND INADDRANY).

### **\_FEKFSCMD\_TP\_NAME\_**

Nom de programme transactionnel APPC. La valeur par défaut est FEKFRSRV. Supprimez la mise en commentaire et modifiez cette définition si vous n'avez pas utilisé le nom de programme transactionnel par défaut lors de la définition de la transaction APPC. Il s'agit d'une directive facultative.

### **\_FEKFSCMD\_PARTNER\_LU\_**

Force le serveur à utiliser cette unité logique partenaire APPC. La valeur par défaut est l'unité logique de base indiquée pendant la configuration d'APPC. Il s'agit d'une directive facultative.

### **GSK\_CRL\_SECURITY\_LEVEL**

Indique le niveau de sécurité que les applications SSL doivent utiliser lors de l'établissement d'un contact avec les serveurs LDAP pour détecter les certificats révoqués dans les listes CRL pendant la validation des certificats. La valeur par défaut est MEDIUM. Supprimez la mise en commentaire et modifiez le paramètre pour appliquer la valeur indiquée. Il s'agit d'une directive facultative. Les valeurs suivantes sont admises :

- LOW : La validation du certificat n'échoue pas si le serveur LDAP ne peut pas être contacté.
- MEDIUM : La validation du certificat requiert l'établissement d'un contact avec le serveur LDAP mais pas la définition d'une liste CRL. Il s'agit de la valeur par défaut.
- HIGH : La validation du certificat requiert l'établissement d'un contact avec le serveur LDAP et la définition d'une liste CRL.

**Remarque :** Cette directive requiert z/OS 1.9 ou version suivante.

### **GSK\_LDAP\_SERVER**

Indique un ou plusieurs noms d'hôte du serveur LDAP séparés par des espaces. Supprimez la mise en commentaire et modifiez la définition pour permettre aux serveurs LDAP indiqués d'obtenir leur liste de révocation de certificat. Il s'agit d'une directive facultative.

Le nom d'hôte peut être une adresse TCP/IP ou une adresse URL. Chaque nom d'hôte peut inclure un numéro de port facultatif séparé du nom d'hôte par le signe deux points (:).

### **GSK\_LDAP\_PORT**

Indique le port du serveur LDAP. La valeur par défaut est 389. Supprimez la mise en commentaire et modifiez la valeur pour appliquer la valeur indiquée. Il s'agit d'une directive facultative.

**GSK\_LDAP\_USER**

Indique le nom distinctif à utiliser lors de la connexion au serveur LDAP. Supprimez la mise en commentaire et modifiez la valeur pour appliquer la valeur indiquée. Il s'agit d'une directive facultative.

**GSK\_LDAP\_PASSWORD**

Indique le mot de passe à utiliser lors de la connexion au serveur LDAP. Supprimez la mise en commentaire et modifiez la valeur pour appliquer la valeur indiquée. Il s'agit d'une directive facultative.

Les définitions suivantes sont nécessaires, et ne doivent pas être modifiées sans instruction du point service IBM :

**\_CEE\_RUNOPTS**

Options d'exécution de Language Environment (LE). La valeur par défaut est "ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)". Ne pas modifier.

**\_BPX\_SHAREAS**

Exécute les processus d'avant-plan dans le même espace adresse que le shell. La valeur par défaut est YES. Ne pas modifier.

**\_BPX\_SPAWN\_SCRIPT**

Exécute des scripts de shell directement à partir de la fonction spawn(). La valeur par défaut est YES. Ne pas modifier.

**JAVA\_PROPAGATE**

Propage le contexte de sécurité et de charge de travail pendant la création d'unité d'exécution (Java version 1.4 et supérieures uniquement). La valeur par défaut est NO. Ne pas modifier.

**RSE\_LIB**

Chemin d'accès à la bibliothèque de l'Explorateur de systèmes éloignés RSE. La valeur par défaut est \$RSE\_HOME/lib. Ne pas modifier.

**PATH** Chemin d'accès de la commande. La valeur par défaut est `.:$JAVA_HOME/bin:$RSE_HOME/bin:$CMDSESV_BASE_HOME/bin:$PATH`. Ne pas modifier.

**LIBPATH**

Chemin d'accès à la bibliothèque. La valeur par défaut est trop longue pour être reprise. Ne pas modifier.

**CLASSPATH**

Chemin de classes. La valeur par défaut est trop longue pour être reprise. Ne pas modifier.

**\_RSE\_CMDSESV\_OPTS**

Options Java spécifiques au service Commandes TSO. La valeur par défaut est "&SESSION=SPAWN\$\_RSE\_CMDSESV\_OPTS". Ne pas modifier.

**\_RSE\_JAVAOPTS**

Options Java supplémentaires spécifiques de RSE. La valeur par défaut est trop longue pour être reprise. Ne pas modifier.

**\_RSE\_SERVER\_CLASS**

Classe Java pour le serveur RSE. La valeur par défaut est `org.eclipse.dstore.core.server.Server`. Ne pas modifier.

**\_RSE\_DAEMON\_CLASS**

Classe Java pour le démon RSE. La valeur par défaut est `com.ibm.etools.zos.server.RseDaemon`. Ne pas modifier.



#### **\_RSE\_POOL\_SERVER\_CLASS**

La classe Java pour le pool d'unités d'exécution RSE. La valeur par défaut est `com.ibm.etools.zos.server.ThreadPoolProcess`. Ne pas modifier.

#### **\_RSE\_LOCKD\_CLASS**

Classe Java du démon lock RSE. La valeur par défaut est `com.ibm.ftt.rse.mvs.server.miners.MVSLockDaemon`. Ne pas modifier.

#### **\_RSE\_SERVER\_TIMEOUT**

Valeur de dépassement de délai pour le serveur RSE (attente du client) en millisecondes. La valeur par défaut est 120000 (2 minutes). Ne pas modifier.

#### **SCLMDT\_BASE\_HOME**

Répertoire de base du code SCLM Developer Toolkit. Le répertoire par défaut est `$RSE_HOME`. Ne pas modifier.

#### **SCLMDT\_WORK\_HOME**

Répertoire de travail de base de SCLM Developer Toolkit. Le répertoire par défaut est `$_CMD SERV_WORK_HOME`. Ne pas modifier.

#### **CGI\_DTWORK**

Support SCLM Developer Toolkit pour les clients plus anciens. La valeur par défaut est `$_SCLMDT_WORK_HOME`. Ne pas modifier.

## **Définition de PORTRANGE disponible pour un serveur RSE**

Cette partie de la personnalisation de `rsed.envvars` spécifie les ports par lesquels le serveur RSE peut communiquer avec le client. Cette plage de ports n'a pas de connexion avec le port du démon RSE.

Afin de mieux comprendre l'utilisation des ports, une brève description du processus de connexion RSE est incluse ci-dessous :

1. Le client se connecte au port hôte 4035 du démon RSE.
2. Le démon RSE crée une unité d'exécution de serveur RSE.
3. Le serveur RSE ouvre un port hôte pour que le client se connecte. Le choix de ce port peut être configuré par l'utilisateur, soit au niveau du client dans l'onglet des propriétés du sous-système (méthode non recommandée) soit par l'intermédiaire de la définition `_RSE_PORTRANGE` du fichier `rsed.envvars`.
4. Le démon RSE renvoie le numéro de port au client.
5. Le client se connecte au port hôte.

#### **Remarque :**

- Le processus est identique pour la méthode de connexion alternative (facultative) à l'aide de REXEC/SSH.
- Pour plus d'informations, voir Chapitre 11, «Compréhension de Developer for System z», à la page 193.

Pour spécifier la gamme de ports permettant au client de communiquer avec z/OS, supprimez la mise en commentaire et personnalisez la ligne suivante dans `rsed.envvars`:

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

**Remarque :** Avant de sélectionner une plage de ports, vérifiez qu'elle est disponible sur votre système à l'aide des commandes **NETSTAT** et **NETSTAT PORTL**.

Le format du paramètre PORTRANGE est le suivant : `_RSE_PORTRANGE=min-max` (max est non inclusif ; par exemple `_RSE_PORTRANGE=8108-8118` indique que la plage des ports 8108 à 8117 peut être utilisée). Le numéro de port utilisé par le serveur RSE est déterminé en fonction des priorités suivantes :

1. Si un numéro de port différent de zéro est indiqué dans les propriétés de sous-système du client, ce numéro de port est utilisé. Si le port n'est pas disponible, la connexion échoue. Cette configuration n'est pas recommandée.

**Remarque :** L'hôte peut refuser ce type de demande de connexion en spécifiant la directive `deny.nonzero.port=true` dans `rsed.envvars`. Pour plus d'informations sur cette directive, voir «Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JAVAOPTS`».

2. Si le numéro de port dans les propriétés de sous-système a pour valeur 0 et que `_RSE_PORTRANGE` est spécifié dans `rsed.envvars`, la gamme de ports spécifiée par `_RSE_PORTRANGE` est utilisée. Si aucun port de la plage n'est disponible, la connexion échoue.
3. Si le numéro de port dans les propriétés de sous-système est 0 et que `_RSE_PORTRANGE` n'est pas spécifié dans `rsed.envvars`, tout port disponible est utilisé.

**Remarque :** Lorsqu'un serveur ouvre un port et passe en mode écoute, le numéro de port ne peut pas être utilisé par un autre serveur, mais lorsque la connexion est établie, ce numéro de port est de nouveau utilisable. Ainsi, le nombre de ports de la plage ne limite en rien le nombre d'utilisateurs susceptibles de se connecter simultanément.

## Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JAVAOPTS`

Avec les différentes directives `_RSE_*OPTS`, `rsed.envvars` permet de transmettre des paramètres supplémentaires à Java lors du démarrage des processus RSE. Les exemples d'options inclus dans `rsed.envvars` peuvent être activés en supprimant la mise en commentaire.

`_RSE_JAVAOPTS` définit les options Java standard et spécifiques de RSE.

`_RSE_JAVAOPTS=""`

Initialisation variable. Ne pas modifier.

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

Définit la taille de pile initiale (Xms) et maximale (Xmx). Les valeurs par défaut sont 1M et 256M, respectivement. Modifiez la valeur pour appliquer la taille de pile de votre choix. Si cette directive est mise en commentaire, les valeurs par défaut Java sont alors utilisées, 4M et 512M respectivement (1M et 64M pour Java 5.0).

**Remarque :** Pour déterminer les valeurs optimales de cette directive, voir «Définitions de ressources essentielles», à la page 235.

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

Répertoire contenant les données de consignation du serveur et du démon RSE, ainsi que les données d'audit RSE. La valeur par défaut est `/var/rdz/logs`. Modifiez la valeur pour appliquer l'emplacement de votre choix. Si cette directive est mise en commentaire, le répertoire de base de l'ID utilisateur affecté au démon RSE est utilisé. Le répertoire de base est défini dans le segment de sécurité OMVS de l'ID utilisateur.

**Remarque :** Si cette directive (ou son équivalent, le répertoire de base) n'est pas indiquée dans le chemin d'accès absolu (le chemin ne débute pas par une barre oblique (/)), l'emplacement réel du journal est relatif au répertoire de configuration (par défaut /etc/rdz).

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

Répertoire contenant les journaux propres à l'utilisateur. La valeur par défaut est /var/rdz/logs. Modifiez la valeur pour appliquer l'emplacement de votre choix. Si cette directive est mise en commentaire, le répertoire de base de l'ID utilisateur client est utilisé. Le répertoire de base est défini dans le segment de sécurité OMVS de l'ID utilisateur.

**Remarque :**

- Si cette directive (ou son équivalent, le répertoire de base) n'est pas indiquée dans le chemin d'accès absolu (le chemin ne débute pas par une barre oblique (/)), l'emplacement réel du journal est relatif au répertoire de configuration (par défaut /etc/rdz).
- Le chemin complet aux journaux d'utilisateur est userlog/dstorelog/\$LOGNAME/, où userlog est la valeur de la directive user.log, dstorelog celle de la directive DSTORE\_LOG\_DIRECTORY et \$LOGNAME l'ID utilisateur des clients en majuscules.
- Vérifiez que les données de droit de userlog/dstorelog sont définies de sorte que chaque client puisse créer \$LOGNAME.

**`_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_LOG_DIRECTORY="`**

Ce répertoire est ajouté au chemin spécifié dans la directive user.log. Ils permettent de créer le chemin contenant les journaux propres à l'utilisateur. La valeur par défaut est une chaîne nulle. Modifiez-la pour appliquer l'utilisation du répertoire indiqué. Si cette directive est mise en commentaire, .eclipse/RSE/ est utilisé.

**Remarque :**

- Le chemin complet aux journaux d'utilisateur est userlog/dstorelog/\$LOGNAME/, où userlog est la valeur de la directive user.log, dstorelog celle de la directive DSTORE\_LOG\_DIRECTORY et \$LOGNAME l'ID utilisateur des clients en majuscules.
- Le répertoire spécifié ici est relatif au répertoire indiqué dans user.log. Par conséquent, il ne peut pas démarrer avec une barre oblique (/).
- Vérifiez que les données de droit de userlog/dstorelog sont définies de sorte que chaque client puisse créer \$LOGNAME.

Les directives suivantes sont mises en commentaire par défaut.

**`#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Dmaximum.clients=60"`**

Nombre maximal de clients pris en charge par un même pool d'unités d'exécution. La valeur par défaut est 60. Supprimez la mise en commentaire et personnalisez l'option pour limiter le nombre de clients par pool d'unités d'exécution. Notez que d'autres limites risquent d'empêcher RSE d'atteindre cette limite.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.threads=1000"**

Nombre maximum d'unités d'exécution actives d'un pool d'unités d'exécution pour autoriser de nouveaux clients. La valeur par défaut est 1000. Supprimez la mise en commentaire et personnalisez pour limiter le nombre de clients par pool d'unités d'exécution en fonction du nombre d'unités d'exécution utilisées. Notez que chaque connexion client utilise plusieurs unités d'exécution (au moins 16) et que d'autres limites risquent d'empêcher RSE d'atteindre cette valeur maximale.

**Remarque :** Cette valeur doit être inférieure à celle de MAXTHREADS et MAXTHREADTASKS dans SYS1.PARMLIB(BPXPRMxx).

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dminimum.threadpool.process=1"**

Nombre minimal de pools d'unités d'exécution actifs. La valeur par défaut est 1. Supprimez la mise en commentaire de cette ligne et personnalisez-la pour lancer au moins le nombre de processus de pool d'unités d'exécution répertoriés. Les processus de pool d'unité d'exécution sont utilisés pour l'équilibrage de charge des unités d'exécution du serveur RSE. Des processus supplémentaires sont démarrés, si nécessaire. Le démarrage immédiat de nouveaux processus permet d'éviter les délais de connexion mais utilise davantage de ressources pendant les phases d'inactivité.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.threadpool.process=100"**

Nombre maximal de pools d'unités d'exécution actifs. La valeur par défaut est 100. Supprimez la mise en commentaire et personnalisez pour limiter le nombre de processus de pool d'unité d'exécution. Les processus de pool d'unités d'exécution sont utilisés pour l'équilibrage de charge des unités d'exécution du serveur RSE ; si vous les limitez, ils limiteront donc la quantité de connexions client actives.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dipv6=true"**

Version TCP/IP. La valeur par défaut est false, ce qui signifie qu'une interface IPv4 est utilisée. Supprimez la mise en commentaire et indiquez true pour utiliser une interface IPv6.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dkeep.last.log=true"**

Conservez une copie des fichiers journaux de l'hôte appartenant à la session précédente. La valeur par défaut est false. Supprimez la mise en commentaire et indiquez true afin de renommer \*.last les précédents fichiers journaux lors du démarrage du serveur et de la connexion du client.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.standard.log=true"**

Ecrivez les flux stdout et stderr des pools d'unités d'exécution dans un fichier journal. La valeur par défaut est false. Supprimez la mise en commentaire et indiquez true pour sauvegarder les flux stdout et stderr. Les fichiers journaux ainsi obtenus se trouvent dans le répertoire référencé par la directive daemon.log.

**Remarque :**

- La commande de l'opérateur **MODIFY RSESTANDARDLOG** peut être utilisée pour arrêter ou démarrer de manière dynamique la mise à jour des fichiers journaux de flux.
- Lorsque la directive enable.standard.log est active, il n'existe aucun fichier journal stdout.log et stderr.log

propre à l'utilisateur. Les données propres à l'utilisateur sont à présent écrites dans le flux du pool d'unités d'exécution RSE.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.port.of.entry=true"**

Option de vérification du port d'entrée (POE). La valeur par défaut est false. Supprimez la mise en commentaire et indiquez true pour appliquer la vérification du port d'entrée sur les connexions client. Pendant la vérification du port d'entrée, l'adresse IP du client est mappée sur une zone de sécurité d'accès réseau par votre logiciel de sécurité. L'ID utilisateur du client doit disposer des droits d'utilisation du profil qui définit la zone de sécurité.

**Remarque :**

- La vérification du port d'entrée doit également être activée dans votre produit de sécurité.
- Lorsque le port d'entrée est activé, il est également activé pour d'autres services z/OS UNIX (INETD, par exemple).

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.certificate.mapping=false"**

Utilisez le logiciel de sécurité pour authentifier une connexion avec un certificat X.509. La valeur par défaut est true. Supprimez la mise en commentaire et indiquez false pour demander au démon RSE d'effectuer l'authentification sans utiliser le support X.509 du logiciel de sécurité.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.automount=true"**

Prise en charge des répertoires de base créés par montage automatique z/OS UNIX. La valeur par défaut est false. Supprimez la mise en commentaire et indiquez true pour s'assurer que le montage automatique z/OS UNIX utilise bien l'ID utilisateur du client comme propriétaire du répertoire.

**Remarque :** Le montage automatique z/OS UNIX utilise l'ID utilisateur du processus qui appelle le service lors de la création d'un système de fichiers. Si cette option est désactivée, ce processus est le serveur de pools d'unités d'exécution RSE (ID utilisateur STCRSE). Si cette option est activée, un processus temporaire est créé à l'aide de l'ID utilisateur du client avant d'appeler le service.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Denable.audit.log=true"**

Option d'audit. La valeur par défaut est false. Supprimez la mise en commentaire et indiquez true pour appliquer la consignation des actions effectuées par les clients par la fonction d'audit. Les journaux d'audit sont générés à l'emplacement du journal du démon RSE. Voir l'option daemon.log de la variable \_RSE\_JAVAOPTS pour connaître son emplacement.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Daudit.cycle=30"**

Nombre de jours conservés dans un journal d'audit. Le nombre par défaut est 30. Supprimez la mise en commentaire et personnalisez le nombre pour contrôler la quantité de données d'audit consignées dans un journal d'audit. Le nombre maximal est de 365.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Daudit.retention.period=0"**

Nombre de jours pendant lesquels les journaux d'audit sont conservés. Le nombre par défaut est 0 (pas de limite). Supprimez la mise en commentaire et personnalisez le nombre afin de supprimer les journaux d'audit après un nombre de jours donné. Le nombre maximal est de 365.

| **#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Ddeny.nonzero.port=true"**

| Interdire au client le choix du numéro de port de communication. La  
| valeur par défaut est false. Supprimez la mise en commentaire et indiquez  
| true pour refuser des connexions là où le client indique quel port le  
| serveur RSE doit utiliser pour la connexion. Pour plus d'informations, voir  
| «Définition de PORTRANGE disponible pour un serveur RSE», à la page  
| 41.

| **#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dsingle.logon=false"**

| Interdire à un ID utilisateur de se connecter plusieurs fois. La valeur par  
| défaut est true. Supprimez la mise en commentaire et indiquez false pour  
| permettre à un ID utilisateur de se connecter plusieurs fois à un démon  
| RSE.

| **Remarque :** Une deuxième tentative de connexion provoquera l'annulation  
| de la première par l'hôte si cette directive est désactivée ou  
| définie à false. Cette annulation est accompagnée par un  
| message FEK210I à la console.

| **RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dprocess.cleanup.interval=0"**

| Suppression automatique des pools d'unités d'exécution RSE qui sont dans  
| un état d'erreur irrémédiable. Par défaut, les pools d'unités d'exécution  
| RSE en erreur ne sont pas supprimés automatiquement. Supprimez la mise  
| en commentaire et personnalisez pour que des serveurs de pools d'unités  
| d'exécution en erreur soient supprimés automatiquement à chaque  
| intervalle (l'unité d'intervalle est exprimée en secondes). L'indication 0  
| désactive la fonction.

| **#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DAPPLID=FEKAPPL"**

| ID d'application du serveur RSE. La valeur par défaut est FEKAPPL.  
| Supprimez la mise en commentaire et personnalisez cette option pour  
| utiliser l'ID d'application.

| **Remarque :**

- L'ID application doit être défini dans votre logiciel de sécurité. Sinon, le client ne peut pas se connecter.
- Voir «Utilisation de PassTickets», à la page 166 pour les implications en termes de sécurité lors du changement de cette valeur.
- L'ID de l'application doit correspondre à l'ID d'application utilisé par le moniteur de travaux JES. Pour savoir comment définir l'ID d'application pour le moniteur de travaux JES, voir «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27.

| **#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -DDENY\_PASSWORD\_SAVE=true"**

| Option d'enregistrement du mot de passe. La valeur par défaut est false.  
| Supprimez la mise en commentaire et indiquez true pour empêcher les  
| utilisateurs de sauvegarder leur mot de passe hôte sur le client. Les mots  
| de passe enregistrés auparavant seront supprimés. Cette option ne  
| fonctionne qu'avec des clients des versions 7.1 et ultérieures.

| **#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dhide\_zos\_unix=true"**

| Masquez l'option z/OS UNIX. La valeur par défaut est false. Supprimez  
| la mise en commentaire et indiquez true pour empêcher les utilisateurs de  
| voir les éléments z/OS UNIX (structure de répertoires et ligne de  
| commande) sur le client. Cette option fonctionne uniquement avec la  
| version 7.6 et les versions ultérieures.



```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS
-DDSTORE_IDLE_SHUTDOWN_TIMEOUT=3600000"
    Déconnecte les clients inactifs. Par défaut, les clients inactifs ne sont pas
    déconnectés. Supprimez la mise en commentaire de la ligne et
    personnalisez-la pour déconnecter les clients inactifs depuis une durée
    supérieure à celle indiquée en millisecondes (3600000 est égal à 1 heure).
```

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_TRACING_ON=true"
    Démarre la fonction de trace dstore. A utiliser uniquement sur instruction
    du point service IBM. Notez que le fichier journal .dstoreTrace obtenu est
    créé en Unicode (ASCII), pas en EBCDIC.
```

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DDSTORE_MEMLOGGING_ON=true"
    Démarre la fonction de trace de mémoire dstore. A utiliser uniquement sur
    instruction du point service IBM. Notez que le fichier journal
    .dstoreMemLogging obtenu est créé en Unicode (ASCII), pas en EBCDIC.
```

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"
    Utilise une transaction APPC pour le service Commandes TSO. Par défaut,
    la passerelle client TSO/ISPF d'ISPF est utilisée. Supprimez la mise en
    commentaire pour utiliser une transaction APPC à la place. Ne modifiez
    pas la valeur affectée.
```

## Définition de paramètres de démarrage Java supplémentaires avec `_RSE_CMDSERV_OPTS`

Avec les différentes directives `_RSE_*OPTS`, `rsed.envvars` permet de transmettre des paramètres supplémentaires à Java lors du démarrage des processus RSE. Les exemples d'options inclus dans `rsed.envvars` peuvent être activés en supprimant la mise en commentaire.

Les directives `_RSE_CMDSERV_OPTS` sont des options Java spécifiques de RSE et prennent effet uniquement lorsque Developer for System z utilise la passerelle client TSO/ISPF d'ISPF (valeur par défaut).

```
_RSE_CMDSERV_OPTS=""
```

Initialisation variable. Ne pas modifier.

```
_RSE_CMDSERV_OPTS="$_RSE_CMDSERV_OPTS &ISPROF=
&SYSUID..ISPROF="
```

Utilisez un profil ISPF existant pour l'initialisation ISPF. Supprimez la mise en commentaire et modifiez le nom du fichier pour utiliser le profil ISPF indiqué.

Les variables suivantes peuvent être utilisées dans le nom du fichier :

- `&SYSUID`, en remplacement de l'ID utilisateur du développeur
- `&SYSPREF`, en remplacement du préfixe TSO du développeur

---

## ISPF.conf, fichier de configuration de la passerelle client TSO/ISPF d'ISPF

Le service de passerelle client TSO/ISPF d'ISPF utilise les définitions figurant dans `ISPF.conf` pour créer un environnement valide en vue d'exécuter les commandes de traitement par lots TSO et ISPF. Developer for System z utilise cet environnement pour exécuter certains services MVS. Citons parmi ces services, le service Commandes TSO, le service SCLM Developer Toolkit et une méthode de démarrage CARMA alternative.

Le fichier `ISPF.conf` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**.

Les lignes mises en commentaire commencent par un astérisque (\*) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne. La continuation de ligne n'est pas prise en charge. Lorsque vous concaténez les noms de fichiers, ajoutez-les sur la même ligne et séparez les noms avec une virgule (,).

Outre les noms des fichiers ISPF appropriés, vous devez ajouter le nom du fichier du service Commandes TSO, `FEK.SFEKPROC`, à l'instruction `SYSPROC` ou `SYSEXEC`, comme indiqué dans l'exemple suivant.

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

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

*Figure 9. ISPF.conf - Fichier de configuration ISPF*

**Remarque :**

- Vous pouvez ajouter vos propres instructions de type définition de données et concaténations de fichiers pour personnaliser l'environnement TSO, ce qui permet de simuler une procédure d'ouverture de session TSO. Pour plus d'informations, voir le Chapitre 16, «Personnalisation de l'environnement TSO», à la page 269.
- La passerelle client TSO/ISPF risque ne pas fonctionner correctement si vous utilisez un produit (tiers) qui intercepte les commandes ISPF, telles que **ISPSTART**. Vérifiez la documentation du produit pour savoir comment la désactiver pour Developer for System z. Si le produit requiert l'attribution d'une instruction de définition de données spécifique à DUMMY, vous pouvez simuler cette opération dans `ISPF.conf` en attribuant cette instruction de définition de données à `nullfile`.

Par exemple :

```
ISPTRACE=nullfile
```

- Lorsque vous utilisez la directive `allocjob`, veillez à ne pas annuler les définitions DD effectuées précédemment dans `ISPF.conf`.
- Un arrêt système 522 pour le module `ISPZTS0` peut se produire si la valeur du paramètre `JWT` du membre `parmlib SMFPRMxx` est inférieure à la valeur `ISPF_timeout` définie dans `ISPF.conf`. Cela n'a pas d'incidence sur les opérations de Developer for System z car la passerelle client TSO/ISPF est redémarrée automatiquement, si nécessaire.



- Les modifications sont appliquées pour toutes les invocations. Il est inutile de redémarrer le serveur.

---

## Composants facultatifs

La procédure de personnalisation ci-dessus s'applique à la configuration du produit Developer for System z de base. Reportez-vous aux chapitres relatifs aux composants facultatifs pour connaître leurs exigences de personnalisation :

- Chapitre 3, «(Facultatif) Common Access Repository Manager (CARMA)», à la page 51
- Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75
- Chapitre 5, «(Facultatif) SCLM Developer Toolkit», à la page 83
- «(Facultatif) Procédure mémorisée DB2», à la page 91
- «(Facultatif) Support de langue bidirectionnelle CICS», à la page 94
- «(Facultatif) Chiffrement SSL RSE», à la page 95
- «(Facultatif) Fonction de trace RSE», à la page 98
- «(Facultatif) Groupes de propriétés basés sur l'hôte», à la page 100
- «(Facultatif) Projets basés sur l'hôte», à la page 101
- «(Facultatif) Intégration de File Manager», à la page 102
- «(Facultatif) Caractères non éditables», à la page 103
- «(Facultatif) Utilisation de REXEC (ou SSH)», à la page 104
- «(Facultatif) Transaction APPC pour le service Commandes TSO», à la page 106
- «(Facultatif) Nettoyage du répertoire WORKAREA», à la page 110

---

## Vérification de l'installation

Vous trouverez la description des différents programmes de vérification d'installation (IVP pour Installation Verification Program) dans Chapitre 7, «Vérification de l'installation», à la page 111, car certains des IVP sont destinés à des composants facultatifs.



---

## Chapitre 3. (Facultatif) Common Access Repository Manager (CARMA)

Common Access Repository Manager (CARMA) représente un gain de productivité pour les développeurs qui créent des gestionnaires d'accès au référentiel (RAM). Un RAM constitue une interface de programmation d'application pour les gestionnaires de configuration logicielle (SCM) basés sur z/OS basé sur .

Inversement, les applications créées par des utilisateurs peuvent démarrer un serveur CARMA qui charge les RAM et fournit une interface standard pour accéder au SCM.

Developer for System z prend en charge plusieurs méthodes de lancement d'un serveur CARMA, chacune possédant ses avantages et inconvénients.

- La méthode de "soumission par lots" démarre le serveur CARMA en soumettant un travail. Il s'agit de la méthode par défaut utilisée dans les fichiers de configuration fournis. L'avantage de cette méthode est que les journaux CARMA sont facilement accessibles dans la sortie de travaux. Elle permet également d'utiliser un JCL de serveur personnalisé pour chaque développeur qui sera géré par le développeur lui-même. Toutefois, cette méthode utilise un initiateur JES pour chaque développeur qui démarre un serveur CARMA.
- La méthode "CRASTART" démarre le serveur CARMA en tant que sous-tâche au sein de RSE. Elle offre une configuration très flexible grâce à l'utilisation d'un fichier de configuration distinct qui définit les attributions de fichiers et les appels de programme nécessaires pour démarrer un serveur CARMA. Cette méthode offre les meilleures performances et utilise le moins de ressources mais requiert cependant que le module CRASTART se trouve dans LPA.
- La méthode "Passerelle client TSO/ISPF" utilise la passerelle client TSO/ISPF d'ISPF pour créer un environnement TSO ou ISPF dans lequel le serveur CARMA est démarré. Il permet d'attribuer des fichiers de manière flexible grâce aux fonctionnalités de ISPF.conf. Cependant, cette méthode ne convient pas pour accéder aux SCM qui interfèrent avec les opérations TSO ou ISPF normales.

---

### Configuration requise et liste de contrôle

Vous avez besoin de l'aide d'un administrateur de sécurité et d'un administrateur TCP/IP pour effectuer cette tâche de personnalisation, qui requiert les ressources ou tâches de personnalisation spéciales ci-après :

- Plage de ports TCP/IP pour les communications internes
- Règle de sécurité permettant aux développeurs de mettre à jour les méthodes d'accès VSAM CARMA
- (Facultatif) Règle de sécurité pour autoriser les utilisateurs à soumettre des travaux CRA\*
- (Facultatif) Mise à jour LPA

Pour commencer à utiliser CARMA sur votre site, vous devez effectuer les tâches ci-après. Sauf indication contraire, toutes les tâches sont obligatoires.

1. Créez des composants CARMA requis. Pour plus d'informations, voir «Composants CARMA», à la page 52.

2. Effectuez la personnalisation initiale des fichiers RSE pour communiquer avec CARMA. La personnalisation complète dépend de la méthode choisie pour démarrer CARMA. Pour plus d'informations, voir «interface RSE avec CARMA», à la page 53.
3. Sélectionnez une méthode pour démarrer CARMA et apportez les personnalisations requises aux fichiers de configuration associés. Pour plus d'informations, voir :
  - «Démarrage du serveur CARMA à l'aide de la soumission par lots», à la page 56
  - «(Facultatif) Démarrage du serveur CARMA alternatif à l'aide de CRASTART», à la page 57
  - «(Facultatif) Démarrage du serveur CARMA alternatif à l'aide de la passerelle client TSO/ISPF», à la page 60
4. Activez les modèles Repository Access Manager (RAM), si vous le souhaitez. Pour plus d'informations, voir «(Facultatif) Activation des modèles de RAM (Repository Access Manager)», à la page 62.
5. Eventuellement, activez CA Endevor RAM. Pour plus d'informations, voir «(Facultatif) Activation du RAM de CA Endevor à l'aide de SCM», à la page 64.
6. Créez CRAXJCL en remplacement de IRXJCL, si vous le souhaitez. Pour plus d'informations, voir «(Facultatif) IRXJCL versus CRAXJCL», à la page 72.

**Remarque :** Les exemples de membres référencés dans le présent chapitre se trouvent dans FEK.#CUST.\* et /etc/rdz, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

---

## Composants CARMA

Les composants CARMA suivants doivent être personnalisés, indépendamment de la méthode de démarrage choisie. Les exemples de membre indiqués ci-après se trouvent dans FEK.#CUST.JCL, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

1. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA\$VDEF). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRA\$VDEF. CRA\$VDEF crée et amorce le fichier VSAM de configuration CARMA, CRADEF.
2. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA\$VMSG). Pour obtenir les instructions de personnalisation, reportez-vous à la documentation qui se trouve dans CRA\$VMSG. CRA\$VMSG crée et amorce le fichier de méthode d'accès VSAM des messages CARMA, CRAMSG.
3. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA\$VSTR). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRA\$VSTR. CRA\$VSTR crée et amorce le fichier VSAM d'informations personnalisées CARMA, CRASTRS.

**Remarque :**

- Les méthodes d'accès CARMA VSAM créées avec ces travaux définissent les modèles de RAM. Voir «(Facultatif) Activation du RAM de CA Endevor à l'aide de SCM», à la page 64 pour définir CA Endevor RAM.

- Voir le modèle de travail FEK.#CUST.JCL(CRA#UADD) si vous souhaitez fusionner les définitions d'un RAM (personnalisé) dans une configuration VSAM existante. Ce travail doit être personnalisé et soumis pour chaque VSAM CARMA modifié. Voir *Rational Developer for System z Common Access Repository Manager Developer's Guide* (SC23-7660) pour plus d'informations la structure d'enregistrement utilisée par les différents VSAM CARMA.
- Utilisez le modèle de travail FEK.#CUST.JCL(CRA#UQRY) pour extraire les définitions actives d'un VSAM dans un fichier séquentiel.

## Notes de migration VSAM CARMA

Developer for System z version 7.6.1 prend en charge un nouvel agencement de structure de données pour les fichiers VSAM d'informations personnalisées CARMA, CRASTRS, pour supprimer des limitations de longueur de messages.

Dans les versions de Developer for System z antérieures à la version 7.6.1, les chaînes définies dans le fichier VSAM d'informations personnalisées CARMA sont limités aux longueurs prédéfinies. Cette limitation oblige les développeurs RAM à réduire les chaînes de description ou à utiliser des plug-ins côté client pour afficher des chaînes sur toute leur longueur.

Developer for System z version 7.6.1 prend en charge un nouvel agencement de structure de données, de longueur variable, pour des fichiers VSAM d'informations personnalisées CARMA, CRASTRS, dans lequel des chaînes sont séparées par un caractère délimiteur, au lieu d'avoir une longueur fixe.

Personnalisez et soumettez le JCL FEK.SFEKSAMP(CRA#VS2) pour convertir votre fichier VSAM d'informations personnalisées CARMA de longueur fixe, CRASTRS, au nouveau format de longueur variable.

### Remarque :

- A partir de la version 7.6.1, l'exemple de fichier VSAM d'informations personnalisées CARMA est fourni est format longueur variable.
- A partir de la version 7.6.1, le module de chargement CARMA, CRASERV, prend en charge le format à longueur fixe et le format à longueur variable pour le fichier VSAM d'informations personnalisées CARMA.
- Les versions plus anciennes du module de chargement CARMA ne prennent pas en charge le format à longueur variable et produiront des chaînes confuses s'ils sont utilisés avec un fichier VSAM d'informations personnalisées CARMA à longueur variable.

## interface RSE avec CARMA

Le serveur CARMA fournit une interface API standard pour d'autres produits hôte permettant d'accéder à un ou plusieurs gestionnaires de configuration de logiciel (SCM). Cependant, il ne fournit pas de méthodes permettant la communication directe avec un PC client. Pour cela, il s'appuie sur d'autres produits tels que le serveur RSE. Le serveur RSE utilise les paramètres de CRASRV.properties pour démarrer et se connecter au serveur CARMA.

Le fichier CRASRV.properties se trouve dans /etc/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail

FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO OEDIT.

**Remarque :** Vous devez redémarrer la tâche démarrée RSED pour appliquer les modifications apportées.

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

Figure 10. CRASRV.properties – Fichier de configuration de CARMA

#### **port.start**

Premier port utilisé pour la communication entre CARMA et le serveur RSE. Le port par défaut est 5227. La communication sur ce port ne concerne que la machine hôte.

**Remarque :** Avant de sélectionner un port, vérifiez qu'il est disponible sur votre système à l'aide des commandes **NETSTAT** et **NETSTAT PORTL**. Pour plus d'informations, voir «Ports TCP/IP réservés», à la page 153.

#### **port.range**

Plage de ports, commençant par port.start, qui sera utilisée pour la communication CARMA. La valeur par défaut est 100. Par exemple, quand vous utilisez les valeurs par défaut, les ports 5227 à 5326 (inclus) peuvent être utilisés par le gestionnaire CARMA.

#### **startup.script.name**

Définit le chemin d'accès absolu du script de démarrage CARMA. L'emplacement par défaut est /usr/lpp/rdz/bin/carma.startup.rex. Cette commande EXEC REXX va déclencher le démarrage d'un serveur CARMA.

#### **clist.dsname**

Définit la méthode de démarrage du serveur CARMA.

- \*CRASTART indique que le serveur CARMA doit être démarré en tant que sous-tâche au sein de RSE à l'aide de CRASTART. Pour plus d'informations, voir «(Facultatif) Démarrage du serveur CARMA alternatif à l'aide de CRASTART», à la page 57. Si vous spécifiez \*CRASTART, vous devez également spécifier les directives crastart.\*.
- \*ISPF indique que le serveur CARMA doit être démarré à l'aide de la passerelle client de TSO/ISPF d'ISPF. Pour plus d'informations, voir «(Facultatif) Démarrage du serveur CARMA alternatif à l'aide de la passerelle client TSO/ISPF», à la page 60.
- Toute autre valeur définit l'emplacement du CLIST CRASUBMT en utilisant les conventions de dénomination de style TSO. Lorsque les apostrophes (') sont indiquées, le nom de fichier constitue une référence absolue. Lorsque les apostrophes ne sont pas indiquées ('), le nom de fichier est précédé d'un préfixe composé de l'sID utilisateur des client, et non du

préfixe TSO. Le second nom de fichier requiert que les utilisateurs CARMA gèrent leur propre CLIST CRASUBMT.

La valeur par défaut est 'FEK.#CUST.CNTL(CRASUBMT)'. Ce CLIST démarrera un serveur CARMA lors de l'ouverture d'une connexion à l'aide de la méthode de soumission par lots.

#### **crastart.stub**

Module de remplacement z/OS UNIX pour l'appel de CRASTART. La valeur par défaut est /usr/lpp/rdz/bin/CRASTART. Ce module de remplacement rend le module de chargement CRASTART basé sur MVS disponible pour les processus z/OS UNIX. Cette directive est uniquement utilisée si la directive `clist.dsname` possède la valeur \*CRASTART.

#### **crastart.configuration.file**

Indique le nom du fichier de configuration CRASTART. Le nom par défaut est /etc/rdz/crastart.conf. Ce fichier indique les attributions de fichier et les appels de programme nécessaires pour démarrer un serveur CARMA. Cette directive est uniquement utilisée si la directive `clist.dsname` possède la valeur \*CRASTART.

#### **crastart.syslog**

Indique la quantité d'informations consignées dans le journal système lorsque CRASTART démarre un serveur CARMA. La valeur par défaut est Partial. Les valeurs acceptées sont les suivantes :

A (All)	Toutes les informations de traçage sont imprimées dans le journal système
P (Partial)	Seules les informations de connexion, de déconnexion et d'erreur sont imprimées dans le journal système
anything else	Seules les conditions d'erreur sont imprimées dans le journal système

Cette directive est uniquement utilisée si la directive `clist.dsname` possède la valeur \*CRASTART.

#### **crastart.timeout**

Durée, en secondes, avant que le serveur CARMA ne s'arrête en raison d'un manque d'activité. La valeur par défaut est 420 (7 minutes). Cette directive est uniquement utilisée si la directive `clist.dsname` possède la valeur \*CRASTART.

#### **crastart.steplib**

L'emplacement du module CRASTART lors d'un accès par l'intermédiaire de la directive STEPLIB de `rsed.envvars`. La valeur par défaut est FEK.SFEKLP. Supprimez la mise en commentaire et personnalisez cette directive si le module CRASTART ne peut pas faire partie de LPA ou LINKLIST. Notez que des incidents de contrôle de programme et d'APF surviennent si le module CRASTART ne se trouve pas dans la LPA. Cette directive est uniquement utilisée si la directive `clist.dsname` possède la valeur \*CRASTART.

#### **crastart.tasklib**

Autre nom pour le nom DD TASKLIB DD dans `crastart.conf`. La valeur par défaut est TASKLIB. Supprimez la mise en commentaire et personnalisez cette directive si le nom DD TASKLIB possède une signification spéciale pour votre SCM ou RAM et ne peut être utilisé en remplacement de

STEPLIB. Cette directive est uniquement utilisée si la directive `clist.dsname` possède la valeur `*CRASTART`.

---

## Démarrage du serveur CARMA à l'aide de la soumission par lots

Les informations de la présente section décrivent comment configurer la méthode par défaut pour Developer for System z afin de démarrer le serveur CARMA. Cette étape de personnalisation peut être ignorée si vous utilisez une autre méthode de démarrage.

Developer for System z utilise par défaut la méthode de démarrage du serveur CARMA via la soumission par lots, qui n'oblige pas à placer le module CRASTART dans la zone LPA et qui ne dépend pas de la passerelle client TSO/ISPF. La méthode soumet le serveur CARMA sous la forme d'un travail par lot à exécution longue dans JES.

### Ajustement de CRASRV.properties

Le serveur RSE utilise les paramètres de `/etc/rdz/CRASRV.properties` pour démarrer et se connecter à un serveur CARMA, tel qu'indiqué dans «interface RSE avec CARMA», à la page 53. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Notez que RSE doit être redémarré pour que les modifications prennent effet.

Modifiez la valeur de la directive `clist.dsname` en indiquant le nom du fichier ou du membre de la liste CLIST de démarrage du serveur CARMA `CRASUBMT`, comme indiqué dans l'exemple ci-dessous. Pour plus d'informations sur les différentes directives, voir «interface RSE avec CARMA», à la page 53.

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

*Figure 11. CRASRV.properties - Démarrage de CARMA à l'aide de la soumission par lots*

### Ajustement de CRASUBMT

Personnalisez la liste CLIST `CRASUBMT`, comme indiqué dans l'exemple de code ci-dessous. Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans `CRASUBMT`. La liste de commandes `CRASUBMT` soumet un serveur CARMA.

`CRASUBMT` se trouve dans `FEK.#CUST.CNTL`, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15.



```

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

```

Figure 12. CRASUBMT - CARMA startup using batch submit

#### Remarque :

- Vous pouvez ajouter vos propres instructions de définition de données et concaténations de fichiers pour personnaliser l'environnement TSO CARMA et imiter ainsi une procédure de connexion TSO.
- Vous pouvez éventuellement modifier la valeur du délai d'attente CARMA en modifiant la ligne PROC 1 PORT TIMEOUT(420) dans le CLIST FEK.#CUST.CNTL(CRASUBMT). Cette valeur correspond au nombre de secondes que CARMA devra attendre avant de recevoir la commande suivante du client. Définir une valeur sur 0 entraîne l'application de la valeur du délai d'attente par défaut, actuellement 420 secondes (7 minutes).
- Les détails du processus de démarrage CARMA se trouvent dans rsecomm.log. Voir «(Facultatif) Fonction de trace RSE», à la page 98 pour de plus amples informations sur la définition du niveau de détail de rsecomm.log.
- Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

---

## (Facultatif) Démarrage du serveur CARMA alternatif à l'aide de CRASTART

Les informations de la présente section décrivent comment configurer une méthode alternative pour Developer for System z afin de démarrer un serveur CARMA. Cette étape de personnalisation peut être ignorée si vous utilisez une autre méthode de démarrage.

Developer for System z prend en charge une méthode de démarrage d'un serveur CARMA alternatif qui ne dépend pas de la passerelle client TSO/ISPF et qui ne

soumet pas un travail de serveur à l'aide d'un initiateur JES. La méthode utilise CRASTART pour démarrer le serveur CARMA en tant que sous-tâche au sein de RSE et ressemble au service de passerelle client TSO/ISPF.

**Remarque :** Les détails du processus de démarrage CARMA se trouvent dans rsecomm.log. Voir «(Facultatif) Fonction de trace RSE», à la page 98 pour de plus amples informations sur la définition du niveau de détail de rsecomm.log.

## Ajustement de CRASRV.properties

Le serveur RSE utilise les paramètres de /etc/rdz/CRASRV.properties pour démarrer et se connecter à un serveur CARMA, tel qu'indiqué dans «interface RSE avec CARMA», à la page 53. Vous pouvez modifier le fichier à l'aide de la commande TSO OEDIT. Notez que RSE doit être redémarré pour que les modifications prennent effet.

Modifiez la valeur de la directive clist.dsname par \*CRASTART et entrez les bonnes valeurs pour les directives crastart.\* comme illustré dans l'exemple suivant. Pour plus d'informations sur les différentes directives, voir «interface RSE avec CARMA», à la page 53.

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

Figure 13. CRASRV.properties - \*CRASTART alternative CARMA startup

**Remarque :** Un arrêt système 522 pour le module CRASERV se produira si le paramètre JWT du membre SMFPRMxx est défini à une valeur inférieure à la valeur d'expiration définie dans CRASRV.properties. Cette configuration n'a pas d'incidence sur les opérations CARMA car le serveur est redémarré automatiquement, si nécessaire.

## Ajustement de crastart.conf

Conservez une sortie imprimée de l'élément CRASUBMT (voir «Démarrage du serveur CARMA à l'aide de la soumission par lots», à la page 56) à portée de main pour pouvoir vous y référer facilement pendant cette étape de personnalisation. La sortie imprimée sera utile même si vous n'avez pas personnalisé le membre.

CRASTART utilise les définitions de crastart.conf pour créer un environnement valide afin d'exécuter les commandes TSO et ISPF par lots. Developer for System z utilise cet environnement pour exécuter le serveur CARMA appelé CRASERV.

Le fichier crastart.conf se trouve dans /etc/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO OEDIT.

**Remarque :** Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

Les étapes de personnalisation ci-dessous sont requises pour ajuster le fichier de configuration illustré dans l'exemple de code suivant.

- Ajoutez les fichiers attribués à la concaténation STEPLIB de la procédure CRASUBMT à l'instruction TASKLIB dans `crastart.conf`.
- Créez les entrées pour les DD, CRADEF, CRAMSG et CRSTRS de méthode d'accès VSAM CARMA obligatoires. Utilisez les noms de fichier fournis dans la procédure (personnalisée) CRASUBMT.
- Ajoutez toute instruction de définition de données personnalisée et la concaténation de fichiers de données associée, disponible dans la procédure CRASUBMT (personnalisée). Par exemple, ajoutez le nom de l'instruction de définition de données CRARAM1 et le nom du fichier si vous utilisez l'exemple de RAM PDS. Notez que vous pouvez utiliser les noms de fichiers (attribués à l'aide de DISP=SHR) des constructions SYSOUT et DUMMY.
- Vous pouvez éventuellement ajouter toute commande BPXWDYN à l'aide de l'instruction -COMMAND. Plusieurs instructions -COMMAND peuvent être présentes. BPXWDYN vous permet de procéder à des attributions plus complexes, notamment en créant des fichiers temporaires, des dispositions autres que SHR, des attributions à d'autres sous-systèmes, etc. Pour de plus amples informations sur BPXWDYN, voir le document *Using REXX and z/OS UNIX System Services* (SA22-7806).
- Sélectionnez la méthode d'appel de programme souhaité à l'aide de l'instruction PROGRAM. La méthode recommandée est "PROGRAM=IKJEFT01 CRASERV &CRAPRM1. &CRAPRM2.", car elle vous fournit un environnement TSO qui peut gérer un mélange de fichiers APF et non APF. Voir l'exemple `crastart.conf` pour connaître d'autres méthodes.

**Remarque :** Les définitions `crastart.conf` ne peuvent être réparties sur plusieurs lignes.

\* `crastart.conf` - CARMA allocation options

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

Figure 14. `crastart.conf` - \*CRASTART alternative CARMA startup

Les variables suivantes peuvent être utilisées dans le fichier de configuration :

Tableau 10. Variables du fichier `crastart.conf`

&CRAUSER.	ID utilisateur de connexion du client.
&CRADATE.	Date actuelle au format Dyyyymmdd (7 caractères juliens).
&CRATIME.	Heure actuelle au format Thhmmss (heure minute seconde).

Tableau 10. Variables du fichier crastart.conf (suite)

&CRAPRM3. à &CRAPRM9.	Variables supplémentaires avec des valeurs attribuées à l'utilisateur. L'utilisation de ces variables requiert la personnalisation du REXX de démarrage CARMA référencé par startup.script.name dans CRASRV.properties.  Lorsque vous utilisez ces variables, vous devez personnaliser une copie du REXX de démarrage par défaut, /usr/lpp/rdz/bin/carma.startup.rex, et faire pointer startup.script.name sur cette copie. Ceci pour éviter de perdre votre travail lors des mises à jour de maintenance du REXX par défaut.
symbole système	Tout symbole système défini dans SYS1.PARMLIB(IEASYMxx)
-<DD>	Un trait d'union (-) suivi d'un nom de définition de données défini précédemment agit comme une référence amont *.ddname dans JCL. La définition de données d'origine doit être attribuée à l'aide de l'instruction -COMMAND.

**Remarque :** Il n'existe pas de variable pour le préfixe TSO, car TSO n'est pas actif lorsque le fichier de configuration est interprété.

## (Facultatif) Démarrage du serveur CARMA alternatif à l'aide de la passerelle client TSO/ISPF

Les informations de la présente section décrivent comment configurer une méthode alternative pour Developer for System z afin de démarrer un serveur CARMA. Cette étape de personnalisation peut être ignorée si vous utilisez une autre méthode de démarrage.

Developer for System z prend en charge une méthode de démarrage d'un serveur CARMA alternatif qui ne requiert pas que le module CRASTART se trouve dans la LPA et qui ne soumet pas un travail de serveur à l'aide d'un initiateur JES. Cette méthode utilise la passerelle client TSO/ISPF d'ISPF et est similaire au mode d'accès par défaut au service Commandes TSO.

**Remarque :** Les détails du processus de démarrage CARMA se trouvent dans rsecomm.log. Voir «(Facultatif) Fonction de trace RSE», à la page 98 pour de plus amples informations sur la définition du niveau de détail de rsecomm.log.

## Ajustement de CRASRV.properties

Le serveur RSE utilise les paramètres de /etc/rdz/CRASRV.properties pour démarrer et se connecter à un serveur CARMA, tel qu'indiqué dans «interface RSE avec CARMA», à la page 53. Vous pouvez modifier le fichier à l'aide de la commande TSO OEDIT. Notez que RSE doit être redémarré pour que les modifications prennent effet.

Modifiez la valeur de la directive `clist.dsname` par `*ISPF`, comme illustré dans l'exemple suivant. Pour plus d'informations sur les différentes directives, voir «interface RSE avec CARMA», à la page 53.

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

Figure 15. *CRASRV.properties* - *\*ISPF alternative CARMA startup*

## Ajustement d'ISPF.conf

Conservez une sortie imprimée de l'élément CRASUBMT (voir «Démarrage du serveur CARMA à l'aide de la soumission par lots», à la page 56) à portée de main pour pouvoir vous y référer facilement pendant cette étape de personnalisation. La sortie imprimée sera utile même si vous n'avez pas personnalisé le membre.

Le service de passerelle client TSO/ISPF d'ISPF utilise les définitions figurant dans `ISPF.conf` pour créer un environnement valide en vue d'exécuter les commandes de traitement par lots TSO et ISPF. Developer for System z utilise cet environnement pour exécuter le serveur CARMA.

Le fichier `ISPF.conf` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**.

**Remarque :** Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

Les étapes de personnalisation ci-dessous sont requises pour ajuster le fichier de configuration illustré dans l'exemple de code suivant.

- Indiquez les noms appropriés pour les fichiers ISPF obligatoires, sauf `ISPPROF` qui est alloué de manière dynamique.
- Ajoutez la bibliothèque de procédures de Developer for System z, `FEK.SFEKPROC`, à l'instruction `SYSPROC` ou `SYSEXEC` afin que la commande `exec CRASRVI` puisse être trouvée par le système. Cette commande `exec` démarre le serveur CARMA (puis remplace les définitions de données `SYSTSIN` de `CRASUBMT`).
- Ajoutez la concaténation `STEPLIB` de définition de données de la procédure `CRASUBMT` à l'instruction `ispllib`.
- Créez les entrées pour les `DD`, `CRADEF`, `CRAMSG`, et `CRSTRS` de méthode d'accès `VSAM` CARMA obligatoires. Utilisez les noms de fichier fournis dans la procédure (personnalisée) `CRASUBMT`.
- Ajoutez toute instruction de définition de données personnalisée et la concaténation de fichiers de données associée, disponible dans la procédure `CRASUBMT` (personnalisée). Par exemple, ajoutez le nom de l'instruction de définition de données `CRARAM1` et le nom du fichier si vous utilisez l'exemple de `RAM PDS`. Notez que vous pouvez uniquement utiliser les noms de fichiers (attribués à l'aide de `DISP=SHR`).
- Vous pouvez éventuellement supprimer la mise en commentaire et personnaliser la directive `allocexec` pour effectuer des attributions supplémentaires à l'aide d'une commande `exec`.

**Remarque :** N'incluez pas les définitions de données SYSTSIN, SYSTSOUT ou CARMALOG ni aucune instruction de définition de données qui utilise des constructions JES (les données de flots de travaux SYSOUT=, par exemple). Ces entrées doivent être converties pour utiliser les fichiers.

```
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
isp1lib=FEK.SFEKLOAD
ispmlib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
CRADEF =FEK.#CUST.CRADEF
CRAMSG =FEK.#CUST.CRAMSG
CRASTRS=FEK.#CUST.CRASTRS
*CRARAM1=FEK.#CUST.CRARAM1
allocjob=FEK.#CUST.CNTL(CRAISPRX)
```

*Figure 16. ISPF.conf - \*ISPF alternative CARMA startup*

Les définitions de données CARMALOG font référence par défaut à SYSOUT=\* qui ne peut pas être mappé dans ISPF.conf. Vous ne pouvez pas mapper les définitions de données directement sur un fichier, car tous les utilisateurs de Developer for System z utiliseront le même fichier ISPF.conf et, par conséquent, les mêmes fichiers.

En revanche, comme indiqué au Chapitre 16, «Personnalisation de l'environnement TSO», à la page 269, section «Avancé – Utilisation d'une commande exec d'allocation», à la page 271, vous pouvez utiliser une instruction exec d'allocation et allouer un fichier en fonction de l'ID utilisateur actif. Consultez l'exemple de membre CRAISPRX dans le fichier FEK.#CUST.CNTL en tant qu'exemple qui attribue les définitions de données CARMALOG au nom de fichier TSOPREFIX'. 'USERID'.CRA.'TIMESTAMP'.CARMALOG'.

**Remarque :**

- Lorsque vous utilisez la directive allocjob, veillez à ne pas annuler les définitions DD effectuées précédemment dans ISPF.conf.
- Un arrêt système 522 pour le module CRASERV peut se produire si la valeur du paramètre JWT du membre parmlib SMFPRMxx est inférieure à la valeur ISPF\_timeout définie dans ISPF.conf. Cette configuration n'a pas d'incidence sur les opérations CARMA car le serveur est redémarré automatiquement, si nécessaire.

---

## (Facultatif) Activation des modèles de RAM (Repository Access Manager)

Les gestionnaires d'accès au référentiel (RAM) sont des API écrites par l'utilisateur permettant d'interfacer avec les gestionnaires de configuration de logiciel z/OS. Pour activer les modèles de RAM voulus, suivez les instructions des sections suivantes.

**Remarque :** Les modèles de RAM permettent de tester la configuration de votre environnement CARMA et servent d'exemples pour le développement de votre propre RAM. N'UTILISEZ PAS les modèles de RAM fournis dans un environnement de production.



Voir le document *Rational Developer for System z Common Access Repository Manager Developer's Guide* (SC23-7660) pour plus d'informations sur les exemples de RAM et de code source fournis.

Les exemples de membre indiqués ci-après se trouvent dans FEK.#CUST.JCL, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

## Activation du RAM PDS

Le RAM PDS donne une liste de fichiers analogues aux fichiers **MVS** -> **Mes fichiers** de la vue Systèmes distants. Le RAM PDS utilise l'ID RAM 0 par défaut.

**Remarque :** Le RAM PDS implique de démarrer ce gestionnaire de référentiels d'accès commun dans l'ISPF (à l'aide d'ISPSTART).

1. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA#VPDS). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRA#VPDS. CRA#VPDS crée et amorce le fichier de la méthode d'accès VSAM aux messages du RAM PDS.
2. Ajoutez l'instruction de définition de données CRARAM1 à la méthode de démarrage CARMA sélectionnée et indiquez le nom du fichier de la méthode d'accès VSAM aux messages du gestionnaire RAM PDS.

## Activation du RAM SCLM

Le RAM SCLM offre une entrée de base à SCLM, le gestionnaire de configuration de logiciel d'ISPF. Le RAM SCLM utilise l'ID RAM 1 par défaut.

**Remarque :** Le RAM SCLM implique de démarrer ce gestionnaire de référentiels d'accès commun dans l'ISPF (à l'aide d'ISPSTART).

1. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA#VSLM). Pour connaître les instructions de personnalisation, reportez-vous à la documentation qui se trouve dans CRA#VSLM. CRA#VSLM crée et amorce le fichier de la méthode d'accès VSAM aux messages du RAM SCLM.
2. Ajoutez l'instruction de définition de données CRARAM2 à la méthode de démarrage CARMA sélectionnée et indiquez le nom du fichier de la méthode d'accès VSAM aux messages du RAM SCLM.
3. Personnalisez le langage JCL FEK.#CUST.JCL(CRA#ASLM). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRA#ASLM. CRA#ASLM attribue les fichiers nécessaires aux clients de la RAM SCLM.

**Remarque :** Chaque utilisateur doit soumettre FEK.#CUST.JCL(CRA#ASLM) une fois avant d'utiliser le gestionnaire CARMA avec le RAM SCLM. Sinon, une erreur d'attribution est engendrée.

## Activation du gestionnaire RAM du squelette

Le RAM du squelette offre un cadre qui peut être utilisé pour développer vos RAM personnelles. Le RAM du squelette utilise l'ID RAM 3 par défaut.

1. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA#CRAM). Pour connaître les instructions de personnalisation, reportez-vous à la documentation qui se trouve dans CRA#CRAM. CRA#CRAM compile le gestionnaire RAM du squelette.

2. Ajoutez la bibliothèque de chargement qui contient le module compilé du RAM du squelette, CRARAMSA, à la définition de données STEPLIB de la méthode de démarrage CARMA sélectionnée (définition de données TASKLIB pour la méthode CRASTART).

---

## (Facultatif) Activation du RAM de CA Endevor à l'aide de SCM

L'interface d'IBM Rational Developer for System z pour le gestionnaire de configuration logicielle de CA Endevor donne aux clients Developer for System z un accès direct au SCM de CA Endevor SCM. A partir de maintenant, l'interface IBM Rational Developer for System z pour le SCM de CA Endevor est abrégé en CA Endevor SCM RAM (Repository Access Manager).

Contrairement aux modèles de RAM présentés dans cette publication, CA Endevor SCM RAM est un RAM de type production. Il est recommandé de ne pas activer les deux types de RAM dans la même configuration.

Attention : les travaux de configuration proposés pour CA Endevor SCM RAM remplacent la configuration CARMA active par une autre ne contenant que CA Endevor SCM RAM.

**Remarque :** La méthode de démarrage de la passerelle client TSO/ISPF ne peut pas être utilisée conjointement avec CA Endevor SCM RAM.

## Configuration requise et liste de contrôle

Vous avez besoin de l'aide d'un administrateur de sécurité et d'un administrateur TCP/IP pour effectuer cette tâche de personnalisation, qui requiert les ressources ou tâches de personnalisation spéciales ci-après :

- Plage de ports TCP/IP pour les communications internes
- (Facultatif) Règle de sécurité pour autoriser les utilisateurs à soumettre des travaux CRA\*
- (Facultatif) Mise à jour LPA

Pour commencer à utiliser CA Endevor SCM RAM sur votre site, vous devez effectuer les tâches ci-après. Sauf indication contraire, toutes les tâches sont obligatoires.

1. Allouer et amorcer des fichiers VSAM qui définissent CA Endevor SCM RAM pour CARMA. Pour plus d'informations, voir «Définir CA Endevor SCM RAM», à la page 65.
2. Choisir votre méthode de démarrage préférée, soumission par lots ou CRASTART, et effectuer les opérations de personnalisation requises pour les fichiers de configuration apparentés. Pour plus d'informations, voir :
  - «Démarrage de CA Endevor SCM RAM à l'aide de la soumission par lots», à la page 65
  - «Démarrage de CA Endevor SCM RAM à l'aide de CRASTART», à la page 68
3. Vous pouvez aussi personnaliser la commande exec d'allocation utilisée pour une allocation dynamique de fichiers spécifiques de l'utilisateur. Pour plus d'informations, voir «(Facultatif) Personnaliser CRANDVRA», à la page 69.
4. Vous pouvez aussi personnaliser des fichiers de configuration spécifiques de CA Endevor SCM RAM. Pour plus d'informations, voir «(Facultatif) Personnaliser CA Endevor SCM RAM», à la page 70.



## Définir CA Endevor SCM RAM

Les composants CARMA suivants doivent être personnalisés, indépendamment de la méthode de démarrage choisie. Les exemples de membre indiqués ci-après se trouvent dans FEK.#CUST.JCL, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

1. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA#VCAD). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRA\$VDEF. CRA#VCAD crée et amorce le fichier VSAM de configuration CARMA CRADEF.
2. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA\$VMSG). Pour obtenir les instructions de personnalisation, reportez-vous à la documentation qui se trouve dans CRA\$VMSG. CRA\$VMSG crée et amorce le fichier de méthode d'accès VSAM des messages CARMA, CRAMSG.

**Remarque :** Il s'agit du même travail que pour les modèles de RAM.

3. Personnalisez et soumettez le langage JCL FEK.#CUST.JCL(CRA#VCAS). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRA\$VSTR. CRA#VCAS crée et amorce le fichier VSAM d'informations personnalisées CARMA CRASTRS.

**Remarque :**

- CA Endevor SCM RAM utilise l'ID RAM 0 par défaut.
- Voir le modèle de travail FEK.#CUST.JCL(CRA#UADD) si vous souhaitez fusionner les définitions d'un RAM (personnalisé) dans une configuration VSAM existante. Ce travail doit être personnalisé et soumis pour chaque VSAM CARMA modifié. Voir *Rational Developer for System z Common Access Repository Manager Developer's Guide* (SC23-7660) pour plus d'informations la structure d'enregistrement utilisée par les différents VSAM CARMA.
- Utilisez le modèle de travail FEK.#CUST.JCL(CRA#UQRY) pour extraire les définitions actives d'un VSAM dans un fichier séquentiel.

## Démarrage de CA Endevor SCM RAM à l'aide de la soumission par lots

Ne suivez pas cette étape si vous utilisez la méthode CRASTART pour démarrer le serveur CARMA avec CA Endevor SCM RAM.

Developer for System z peut utiliser la méthode de démarrage du serveur CARMA par soumission par lots pour démarrer CA Endevor SCM RAM. La méthode soumet le serveur CARMA sous la forme d'un travail par lot à exécution longue dans JES.

Pour plus d'informations sur la méthode de démarrage par soumission par lots, «Démarrage du serveur CARMA à l'aide de la soumission par lots», à la page 56.

### Ajustement de CRASRV.properties

Le serveur RSE utilise les paramètres de /etc/rdz/CRASRV.properties pour démarrer et se connecter à un serveur CARMA, tel qu'indiqué dans «interface RSE avec CARMA», à la page 53. Vous pouvez modifier le fichier à l'aide de la commande TSO OEDIT. Notez que RSE doit être redémarré pour que les modifications prennent effet.

Changez la valeur de la directive `clist.dsname` en indiquant le nom du fichier ou du membre de la liste CLIST de démarrage du serveur CARMA CRASUBCA, comme indiqué dans l'exemple ci-dessous. Pour plus d'informations sur les différentes directives, voir «interface RSE avec CARMA», à la page 53.

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

Figure 17. Figure x1. CRASRV.properties - Démarrage de CA Endevor SCM RAM à l'aide de la soumission par lots

## Ajustement de CRASUBCA

Personnalisez la liste CLIST CRASUBCA, comme indiqué dans l'exemple de code ci-dessous. Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRASUBCA. La liste de commandes CRASUBCA soumet un serveur CARMA pour le gestionnaire de configuration logicielle (SCM) de CA Endevor.

CRASUBCA se trouve dans `FEK.#CUST.CNTL`, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

```

PROC 1 PORT TIMEOUT(420)
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT,
//          PARM='%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&PORT &TIMEOUT)'
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//          DD DISP=SHR,DSN=CA.NDVR.AUTHLIB
//          DD DISP=SHR,DSN=CA.NDVRO.AUTHLIB
//CRADEF DD DISP=SHR,DSN=FEK.#CUST.CRADEF
//CRAMSG DD DISP=SHR,DSN=FEK.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEK.#CUST.CRASTRS
//*
//SYSPROC DD DISP=SHR,DSN=ISP.SISPCLIB
//          DD DISP=SHR,DSN=FEK.SFEKPROC
//ISPEXEC DD DISP=SHR,DSN=ISP.SISPEXEC
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//ISPTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//*
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DUMMY
//*
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CONLIB
//JCLOUT DD SYSOUT=(A,INTRDR),DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
//EXT1ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT1DEP DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//MSG3FILE DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//C1MSG51 DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//C1EXMSG5 DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=133,BLKSIZE=27930,SPACE=(TRK,(5,5))
//TYPEMAP DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEK.#CUST.PARMLIB(CRASHOW)
$$
EXIT CODE(0)

```

Figure 18. Figure x2. CRASUBCA - Démarrage de CA Endeavor SCM RAM a l'aide de la soumission par lots

#### Remarque :

- Vous pouvez ajouter vos propres instructions de définition de données et concaténations de fichiers pour personnaliser l'environnement TSO CARMA et imiter ainsi une procédure de connexion TSO.
- Vous pouvez éventuellement modifier la valeur du délai d'attente CARMA en modifiant la ligne PROC 1 PORT TIMEOUT(420) dans la liste de commandes. Cette valeur correspond au nombre de secondes que CARMA devra attendre avant de recevoir la commande suivante du client. Définir une valeur sur 0 entraîne l'application de la valeur du délai d'attente par défaut, actuellement 420 secondes (7 minutes).

- Les détails du processus de démarrage CARMA se trouvent dans rsecomm.log. Voir «(Facultatif) Fonction de trace RSE», à la page 98 pour de plus amples informations sur la définition du niveau de détail de rsecomm.log.
- Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

## Démarrage de CA Endevor SCM RAM à l'aide de CRAFTSTART

Ne suivez pas cette étape si vous utilisez la méthode de soumission par lots pour démarrer le serveur CARMA avec CA Endevor SCM RAM.

Developer for System z peut utiliser la méthode de démarrage du serveur CARMA à l'aide de CRAFTSTART pour démarrer CA Endevor SCM RAM. La méthode utilise CRAFTSTART pour démarrer le serveur CARMA comme une sous-tâche au sein de RSE.

Pour plus d'informations sur la méthode de démarrage CRAFTSTART, «(Facultatif) Démarrage du serveur CARMA alternatif à l'aide de CRAFTSTART», à la page 57.

**Remarque :** Les détails du processus de démarrage CARMA se trouvent dans rsecomm.log. Voir «(Facultatif) Fonction de trace RSE», à la page 98 pour de plus amples informations sur la définition du niveau de détail de rsecomm.log.

### Ajustement de CRASRV.properties

Le serveur RSE utilise les paramètres de /etc/rdz/CRASRV.properties pour démarrer et se connecter à un serveur CARMA, tel qu'indiqué dans «interface RSE avec CARMA», à la page 53. Vous pouvez modifier le fichier à l'aide de la commande TSO OEDIT. Notez que RSE doit être redémarré pour que les modifications prennent effet.

Modifiez la valeur de la directive clist.dsname par \*CRAFTSTART et entrez les bonnes valeurs pour les directives craftstart.\* comme illustré dans l'exemple suivant. Pour plus d'informations sur les différentes directives, voir «interface RSE avec CARMA», à la page 53.

```
port.start=5227
port.range=100
startup.script.name=/usr/lpp/rdz/bin/carma.startup.rex
clist.dsname=*CRAFTSTART
craftstart.stub=/usr/lpp/rdz/bin/CRAFTSTART
craftstart.configuration.file=/etc/rdz/craftstart.endevor.conf
craftstart.syslog=Partial
craftstart.timeout=420
#craftstart.step1lib=FEK.SFEKLPA
#craftstart.tasklib=TASKLIB
```

Figure 19. Figure x3. CRASRV.properties - Démarrage de CA Endevor SCM RAM à l'aide de CRAFTSTART

**Remarque :** Un arrêt système 522 pour le module CRASERV se produira si le paramètre JWT du membre SMFPRMxx est défini à une valeur inférieure à la valeur d'expiration définie dans CRASRV.properties. Cette configuration n'a pas d'incidence sur les opérations CARMA car le serveur est redémarré automatiquement, si nécessaire.

## Ajustement de crastart.endevor.conf

CRASTART utilise les définitions de crastart.endevor.conf pour créer un environnement (TSO/ISPF) valide pour appeler CA Endevor SCM. Developer for System z utilise cet environnement pour exécuter CA Endevor SCM RAM.

Le fichier crastart.endevor.conf se trouve dans /etc/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**.

**Remarque :** Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

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

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

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

TYPEMAP = FEK.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEK.#CUST.PARMLIB(CRASHOW)
CONLIB   = CA.NDVR.CONLIB
-COMMAND=ALLOC FI(JCLOUT) SYSOUT(A) WRITER(INTRDR) RECFM(F) LRECL(80)
BLKSIZE(80)
-COMMAND=ALLOC FI(EXT1ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT1DEP) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(MSG3FILE) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(C1EXMSGs) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(C1MSGs1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(133)
BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)

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

Figure 20. crastart.conf - Démarrage de CA Endevor SCM RAM à l'aide de CRASTART

## (Facultatif) Personnaliser CRANDVRA

Les deux méthodes de démarrage (soumission par lots et CRASTART) appellent une commande exec REXX CRANDVRA pour allouer des fichiers spécifiques de l'utilisateur utilisés par CA Endevor SCM RAM.

DD	Nom du fichier	Type
DEPEND	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.DEPEND	Permanent
BROWSE	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.BROWSE	Temporaire
C1PRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	Temporaire

Vous pouvez personnaliser une copie de cette commande exec REXX d'allocation REXX si certaines valeurs par défaut comme le nom du fichier, ne correspondent pas aux normes établies pour votre site. CRANDVRA se trouve dans FEK.SFEKPROC, sauf si vous avez utilisé un qualificateur de niveau supérieur durant l'installation de SMP/E de System z.

Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRANDVRA.

**Remarque :** Vous devez copier le modèle REXX d'allocation dans un nouveau fichier et personnaliser cette copie, ceci pour que ce paramétrage ne soit pas écrasé lors d'une opération de maintenance. En procédant ainsi, vous devez mettre à jour la référence à SFEKPROC dans le SYSEXEC DD de la méthode de démarrage CARMA choisie par vos soins afin que le nouveau nom de fichier puisse correspondre.

## (Facultatif) Personnaliser CA Endevor SCM RAM

Les composants CARMA suivants peuvent être personnalisés, indépendamment de la méthode de démarrage choisie. Les exemples de membre indiqués ci-après se trouvent dans FEK.#CUST.PARMLIB, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

1. (Facultatif) Personnaliser FEK.#CUST.PARMLIB(CRASHOW). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRASHOW. CRASHOW définit des filtres par défaut pour des environnements CA Endevor SCM, des systèmes, etc.
2. (Facultatif) Personnaliser FEK.#CUST.PARMLIB(CRATMAP). Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans CRATMAP. CRATMAP substitue un type CA Endevor SCM aux mappages d'extension de fichier.

## (Facultatif) Prise en charge de plusieurs RAM

CARMA permet que plusieurs RAM soient définis et peut les exécuter simultanément. Toutefois, puisqu'un seul serveur CARMA est actif par utilisateur, même s'il existe plusieurs RAM, certains changements de configuration peuvent s'avérer nécessaires pour permettre à cette configuration de fonctionner.

Un développeur RAM définit les RAM dans le fichier VSAM de configuration CARMA, CRADEF. Durant le démarrage, le serveur CARMA, CRASERV, identifie tous les RAM définis et présente les informations au client CARMA. L'utilisateur peut alors sélectionner un ou plusieurs RAM qui seront chargés dans le serveur CARMA.

Puisque les RAM sont actifs comme des plug-ins du serveur CARMA, vous devez vous assurer que tous les éléments prérequis (comme toutes les allocations de fichiers) de chaque RAM sont disponibles dans l'espace adresse du serveur CARMA. Ceci pourrait nécessiter des changements dans les modèles de configuration CARMA, comme CRASUBMT ou crastart.conf, qui sont fournis avec Developer for System z.

## Exemple

Dans l'exemple suivant, vous démarrez à partir d'une configuration existante avec CA Endevor SCM RAM, à l'aide de la méthode de démarrage CRASTART et ajoutez le modèle PDS RAM.

Définition pour CA Endevor SCM RAM :

- FEK.SFEKVSM2(CRA0VCAD) - Définitions CRADEF
- FEK.SFEKVSM2(CRA0VCAS) - Définitions CRASTRS
- /etc/rdz/crastart.endevor.conf - fichier de configuration CRASTART

Définitions pour PDS RAM:

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

Le processus démarre avec un développeur RAM réunissant les données et informations requises par le programmeur système pour compléter la configuration.

1. Extrayez les données spécifiques pour le PDS RAM à partir des membres SFEKVSM2 (ces membres détiennent des informations pour tous les modèles RAM et pas seulement le PDS RAM).
2. Fusionnez ces données avec les membres CA Endevor SCM RAM SFEKVSM2.
3. Créez une liste d'éléments prérequis spécifiques du PDS RAM :
  - DD CRARAM1, lié à FEK.#CUST.CRARAM1
  - Environnement TSO

Le programmeur système utilise ensuite ces données pour créer les fichiers VSAM CARMA mis à jour et utilise les informations des éléments prérequis pour créer un fichier de configuration CRASTART qui est en mesure de prendre en charge les deux RAM.

1. Utilisez les données combinées comme entrée pour les tâches CRA\$VDEF et CRA\$VSTR afin de créer la configuration CARMA mise à jour ainsi que les fichiers VSAM d'informations personnalisés, CRADEF and CRASTRS.
2. Ajoutez une définition CRARAM1 à crastart.endevor.conf:

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

3. Vérifiez l'instruction PROGRAM dans crastart.endevor.conf pour vous assurer qu'elle est en mesure de fournir les environnements requis par les deux RAM :

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

- IKJEFT01 : TSO, utilisé pour permettre certains appels autorisés dans un environnement non-autorisé et utilisé comme environnement d'exécution de la commande exec de pré-allocation de CA Endevor SCM RAM.



- %CRANDVRA : commande exec de pré-allocation de CA Endevor SCM RAM (située dans FEK.SFEKPROC), qui alloue des fichiers de travail temporaires (et permanents) spécifiques de l'utilisateur.
- NDVRC1 : dorsale CA Endevor, disposant d'un mécanisme intégré pour l'exécution des commandes TSO et ISPF.
- PGM(CRASERV) : commande de démarrage d'un serveur CARMA, dans un format de commande ISPF
- PARM(&CRAPRM1. &CRAPRM2.) : paramètres pour CRASERV, dans un format de commande ISPF. &CRAPRM1 est le port à utiliser et &CRAPRM2 est la valeur du délai d'attente.

CA Endevor SCM RAM est actif dans un environnement ISPF, ce qui implique que l'environnement TSO requis par le PDS RAM est également disponible.

---

## (Facultatif) IRXJCL versus CRAXJCL

Si le serveur CARMA est démarré à l'aide de TSO (IKJEFTxx), vous pouvez rencontrer des incidents si vos RAM appellent des services qui à leur tour appellent l'interface de traitement par lots REXX IRXJCL. L'incident peut survenir lorsque les processeurs appelés par le RAM étaient précédemment exécutés sans TSO ou uniquement dans le TSO en ligne et attribuaient dynamiquement des définitions de données SYSTSIN ou SYSTSPRT. Un exemple de programme, CRAXJCL, est fourni afin de contourner l'incident.

Votre processeur peut échouer s'il tente d'attribuer SYSTSIN ou SYSTSPRT (requis pour IRXJCL) car le TSO de traitement par lots (requis pour CARMA) a déjà attribué et ouvert ses noms de définition de données. Le module de remplacement CRAXJCL tente d'attribuer SYSTSIN et SYSTSPRT à DUMMY mais ignore les erreurs qui se produisent si les attributions échouent.

Cela signifie que lorsque vos processeurs sont exécutés dans un environnement CARMA démarré par TSP, les attributions à SYSTSIN et SYSTSPRT sont identiques à celles utilisées par CARMA. Lorsque les processeurs sont exécutés à l'extérieur de TSO/CARMA, les attributions SYSTSIN et SYSTSPRT seront créées par CRAXJCL. Par conséquent, vos processeurs ne doivent pas se fonder sur le contenu du fichier attribué à SYSTSIN.

Les appels vers IRXJCL sont supposés utiliser le champ PARM pour transmettre le nom REXX et les paramètres de démarrage (voir le document *TSO/E REXX Reference* (SA22-7790)). Cela signifie que SYSTSIN peut être utilisé en toute sécurité par CARMA. Toute sortie envoyée vers SYSTSPRT par IRXJCL finira dans le journal CARMA.

Les processeurs qui appellent le module de remplacement CRAXJCL ne doivent pas essayer d'attribuer des définitions de données SYSTSIN ou SYSTSPRT avant d'appeler CRAXJCL.

## Création de CRAXJCL

Le module de remplacement CRAXJCL est fourni au format source car vous aurez besoin de le personnaliser pour spécifier des attributions spécifiques que vous souhaitez utiliser pour SYSTSPRT. SYSTSIN doit généralement être attribué à un fichier factice.

L'exemple de code source en assembleur et l'exemple de travail de compilation/liaison sont fournis respectivement en tant que



FEK.#CUST.ASM(CRAXJCL) et FEK.#CUST.JCL(CRA#CIRX) à moins que vous n'ayez spécifié un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Personnalisez le code source en assembleur CRAXJCL en fonction de vos besoins en utilisant la documentation fournie dans le membre. Ensuite, personnalisez et soumettez le langage JCL CRA#CIRX pour créer le module de chargement CRAXJCL. Voir la documentation fournie dans CRA#CIRX pour consulter les instructions de personnalisation.



---

## Chapitre 4. (Facultatif) Gestionnaire de déploiement d'application

Developer for System z utilise certaines fonctions du Gestionnaire de déploiement d'application comme approche commune de déploiement pour plusieurs composants. Les étapes de personnalisation répertoriées dans le présent chapitre sont requises si vos développeurs utilisent l'une des fonctions suivantes :

- Enterprise Service Tools (EST)
- BMS Screen Designer
- MFS Screen Designer
- Génération de code CICSTS

**Remarque :** Enterprise Service Tools (EST) regroupe plusieurs outils (le modélisateur de flux de services (SFM) et les services XML pour l'entreprise (XSE), par exemple).

Le gestionnaire de personnalisation de déploiement d'application ajoute le serveur de définition de ressources CICS, qui s'exécute en tant qu'application CICS sur z/OS pour prendre en charge les fonctions suivantes :

- Demandes de ressources CICS
- Demandes d'installation et de désinstallation de définition de ressources CICS dans les environnements SM CICSplex SM et non SM CICSplex
- Demandes de mise en place progressive de programme et de groupe de mappes
- Demandes d'analyse du pipeline
- Demandes d'exportation, d'importation et de mise à jour de manifeste

Les administrateurs CICS peuvent consulter davantage d'informations relatives au serveur CRD dans le Chapitre 15, «Remarques à propos de CICSTS», à la page 257.

---

### Configuration requise et liste de contrôle

Vous aurez besoin de l'aide d'un administrateur CICS, d'un administrateur TCP/IP et d'un administrateur de sécurité pour effectuer cette tâche de personnalisation, qui requiert les tâches de personnalisation spéciales ou les ressources suivantes :

- Port TCP/IP pour les communications externes
- Mettre à jour le langage de contrôle des travaux de la région CICS
- Mettre à jour la définition CSD de la région CICS
- Définir le groupe dans la région CICS
- Règle de sécurité pour permettre une mise à jour des administrateurs dans une méthode d'accès VSAM du gestionnaire de déploiement d'application
- Configuration de sécurité CICSTS
- (Facultatif) Définition des noms de transaction CICS
- (Facultatif) Règle de sécurité afin de permettre une mise à jour des utilisateurs dans une méthode d'accès VSAM du gestionnaire de déploiement d'application

Pour commencer à utiliser le gestionnaire de déploiement d'application sur votre site, vous devez effectuer les tâches ci-après : Sauf indication contraire, toutes les tâches sont obligatoires.

1. Créez le référentiel CRD. Pour plus d'informations, voir «Référentiel de CRD».
2. Choisissez l'interface CICS (RESTful ou Web Service) à utiliser (les interfaces peuvent cohabiter). Pour plus d'informations, voir «RESTful par opposition à Web Service», à la page 77.
3. Le cas échéant, procédez aux personnalisations spécifiques de RESTful. Pour plus d'informations, voir «Serveur CRD utilisant l'interface RESTful», à la page 77.
  - Définissez le serveur CRD dans la région de connexion primaire CICS
  - Définissez éventuellement le serveur CRD dans les régions de connexion secondaires CICS.
  - Eventuellement, personnalisez les ID transaction du serveur CRD.
4. Le cas échéant, procédez aux personnalisations spécifiques du service Web. Pour plus d'informations, voir «Serveur CRD utilisant l'interface Web Service», à la page 79.
  - Ajoutez le gestionnaire de message de pipeline (éventuellement personnalisé) à la concaténation RPL CICS.
  - Définissez le serveur CRD dans la région de connexion primaire CICS.
  - Définissez éventuellement le serveur CRD dans les régions de connexion secondaires CICS.
5. Créez le référentiel de manifeste, si vous le souhaitez. Pour plus d'informations, voir «(Facultatif) Référentiel de manifestes», à la page 81.

---

## Référentiel de CRD

Personnalisez et soumettez un travail ADNVCRD pour attribuer et initialiser le fichier de la méthode d'accès VSAM du référentiel CRD. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre.

ADNVCRD se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Il est recommandé de créer un référentiel séparé pour chaque région de connexion primaire CICS. Le partage du référentiel implique que toutes les régions CICS concernées utiliseront les mêmes valeurs stockées dans le référentiel.

### Remarque :

- Un référentiel existant du serveur CRD doit être agrandi pour activer la prise en charge d'URIMAP ajouté à l'utilitaire Administrative dans Developer for System z version 7.6.1. Pour plus d'informations, voir «Notes de migration de l'utilitaire d'administration», à la page 265.
- Sauf indication contraire, votre référentiel de serveur CRD actuel (qui comporte les valeurs personnalisées) peut être réutilisé d'une édition de Developer for System z à l'autre.

Les utilisateurs doivent posséder les droits d'accès READ au référentiel de CRD, les administrateurs CICS devant posséder les droits d'accès UPDATE.

## Utilitaire d'administration CICS

Developer for System z offre l'utilitaire d'administration qui permet aux administrateurs CICS de fournir des valeurs par défaut pour les définitions de ressource CICS. Ces valeurs par défaut peuvent être accessibles en lecture seulement ou peuvent être éditées par le développeur d'application.

L'utilitaire d'administration est appelé par le modèle de travail ADNJSPAU. L'utilisation de cet utilitaire requiert les droits d'accès UPDATE au référentiel CRD.

ADNJSPAU se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Pour de plus amples informations, voir Chapitre 15, «Remarques à propos de CICS», à la page 257.

---

## RESTful par opposition à Web Service

Les versions 1.4 et ultérieures de CICS Transaction Server proposent un support pour l'interface HTTP reposant sur les principes RESTful (Representational State Transfer). Cette interface RESTful est désormais l'interface CICS stratégique utilisée par les applications client. L'ancienne interface Web Service a été stabilisée, les améliorations concernant uniquement l'interface RESTful.

Le gestionnaire de déploiement d'application suit cette logique et demande au serveur CRD RESTful tous les nouveaux services de Developer for System version 7.6 ou ultérieures.

Les interfaces RESTful et Web Service peuvent être activées simultanément dans une seule région CICS, le cas échéant. Dans ce cas, la région contient deux serveurs CRD actifs. Les deux serveurs partagent le même référentiel de CRD. Notez que CICS émet des avertissements relatives aux définitions en double lorsque la deuxième interface est définie dans la région.

---

## Serveur CRD utilisant l'interface RESTful

Les informations de la présente section expliquent comment définir le serveur CRD qui utilise l'interface RESTful pour communiquer avec le client Developer for System z.

Les interfaces RESTful et Web Service peuvent être activées simultanément dans une seule région CICS, le cas échéant. Dans ce cas, la région contient deux serveurs CRD actifs. Les deux serveurs partagent le même référentiel de CRD. Notez que CICS émet des avertissements relatives aux définitions en double lorsque la deuxième interface est définie dans la région.

## Région de connexion CICS primaire

Le serveur CRD doit être défini pour la région de connexion primaire. Il s'agit de la région gérant le Web (WOR) qui traite les requêtes de service Web en provenance de Developer for System z.

- Placez les modules de chargement FEK.SFEKLOAD(ADNCRD\*, ADNANAL et ADNREST) dans la concaténation RPL CICS (instruction de définition de données DFHRPL) de la région de connexion primaire CICS. La méthode recommandée consiste à

ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS.

- Personnalisez et soumettez un travail ADNCSDRS pour mettre à jour la définition de système CICS (CSD) pour la région de connexion CICS primaire. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre.

ADNCSDRS se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

- Utilisez la commande CEDA appropriée pour installer le groupe Gestionnaire de déploiement d'application pour cette région, par exemple :

```
CEDA INSTALL GROUP(ADNPCRGP)
```

## Régions de connexion CICS secondaires

Le serveur CRD peut également être utilisé avec une ou plusieurs régions de connexion secondaires supplémentaires, qui sont généralement des régions gérant les applications (AOR).

**Remarque :** Il n'est pas nécessaire de suivre ces étapes si CICSplex SM Business Application Services (BAS) est utilisé pour la gestion des définitions de ressource CICS.

- Placez le module de chargement du gestionnaire de déploiement d'application FEK.SFEKLOAD(ADNCRD\*) dans la concaténation RPL CICS (instruction de définition de données DFHRPL) de ces régions de connexion non primaires. La méthode recommandée consiste à ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS.

- Personnalisez et soumettez un travail ADNCSDAR pour mettre à jour le CSD pour ces régions de connexion non primaires. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre.

ADNCSDAR se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

- Utilisez la commande CEDA appropriée pour installer le groupe Gestionnaire de déploiement d'application pour ces régions, par exemple :

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

## (Facultatif) Personnalisation des ID transaction du serveur CRD

Developer for System z fournit plusieurs transactions utilisées par le serveur CRD lors de la définition et de la consultation des ressources CICS.

Tableau 11. ID transaction du serveur CRD par défaut

Transaction	Description
ADMS	Pour les demandes, par l'outil de traitement des manifestes, de modification des ressources CICS. En règle générale, destiné aux administrateurs CICS.
ADMI	Pour les demandes qui définissent, installent ou désinstallent les ressources CICS.

Tableau 11. ID transaction du serveur CRD par défaut (suite)

Transaction	Description
ADMR	Pour toutes les autres demandes qui récupèrent des informations sur les ressources ou l'environnement CICS.

Vous pouvez modifier les ID transaction de sorte qu'ils correspondent aux normes de votre site en suivant la procédure ci-dessous :

1. Personnalisez et soumettez ADNTXNC afin de créer le module de chargement ADNRCUST. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre.
2. Placez le module de chargement obtenu ADNRCUST dans la concaténation RPL CICS (instruction de définition de données DFHRPL) des régions CICS dans lesquelles est défini le serveur CRD.
3. Personnalisez et soumettez ADNCSDTX afin de définir ADNRCUST comme programme aux régions CICS dans lesquelles est défini le serveur CRD. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre.

**Remarque :** Le serveur CRD RESTful va toujours tenter de charger le module de chargement ADNRCUST. Vous pouvez sensiblement améliorer les performances en créant et définissant le module de chargement ADNRCUST, même si vous ne modifiez pas les ID transaction.

## Serveur CRD utilisant l'interface Web Service

Les informations de la présente section expliquent comment définir le serveur CRD qui utilise l'interface Web Service pour communiquer avec le client Developer for System z.

Les interfaces RESTful et Web Service peuvent être activées simultanément dans une seule région CICS, le cas échéant. Dans ce cas, la région contient deux serveurs CRD actifs. Les deux serveurs partagent le même référentiel de CRD. Notez que CICS émet des avertissements relatives aux définitions en double lorsque la deuxième interface est définie dans la région.

## Gestionnaire de message de pipeline

Le gestionnaire de message de pipeline (ADNTMSGH) est utilisé par sécurité dans le traitement de l'ID utilisateur et des mots de passe dans l'en-tête du protocole SOAP. ADNTMSGH est référencé par le fichier de configuration de pipeline et doit donc être placé dans la concaténation RPL CICS. Voir Chapitre 15, «Remarques à propos de CICSTS», à la page 257 pour en savoir davantage sur le gestionnaire de message de pipeline et la configuration de sécurité requise.

Developer for System z met à disposition plusieurs transactions qui sont utilisées par le serveur CRD lors de la définition et de la consultation des ressources CICS. Ces ID de transaction sont définis par ADNTMSGH en fonction de l'opération demandée. Un exemple de langage COBOL est proposé afin de permettre une personnalisation spécifique au site dans ADNTMSGH :

Tableau 12. ID transaction du serveur CRD par défaut

Transaction	Description
ADMS	Pour les demandes, par l'outil de traitement des manifestes, de modification des ressources CICS. Généralement destiné aux administrateurs CICS.

Tableau 12. ID transaction du serveur CRD par défaut (suite)

Transaction	Description
ADMI	Pour les demandes qui définissent, installent ou désinstallent les ressources CICS.
ADMR	Pour toutes les autres demandes qui récupèrent des informations sur les ressources ou l'environnement CICS.

Utilisation des valeurs par défaut :

- Placez le module de chargement FEK.SFEKLOAD(ADNTMSGH) dans la concaténation RPL CICS (instruction de définition de données DFHRPL) de la région de connexion CICS primaire. La méthode recommandée consiste à ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS.

Personnalisation d'ADNTMSGH :

Les exemples de membres ADNMSGH\* se trouvent dans FEK.#CUST.JCL et dans FEK.#CUST.COBOLE, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

- Personnalisez le code source du gestionnaire des messages de pipeline (COBOL) FEK.#CUST.COBOLE(ADNMSGHS) en fonction des normes de votre site.
- Personnalisez et soumettez un travail FEK.#CUST.JCL(ADNMSGHC) pour compiler la source personnalisée ADNMSGHS. Pour obtenir des instructions de personnalisation, consultez la documentation qui se trouve dans ADNMSGHC. Notez que le module de chargement obtenu doit s'appeler ADNTMSGH.
- Placez le module de chargement résultant ADNTMSGH dans la concaténation RPL CICS (instruction de définition de données DFHRPL) de la région de connexion CICS primaire.

**Remarque :** Assurez-vous que le module de chargement personnalisé ADNTMSGH est localisé avant toute référence à FEK.SFEKLOAD, sinon la valeur par défaut sera utilisée.

## région de connexion CICS primaire

Le serveur CRD doit être défini pour la région de connexion primaire. Il s'agit de la région qui traite les demandes de service provenant de Developer for System z.

- Placez les modules de chargement FEK.SFEKLOAD(ADNCRD\*, ADNANAL et ADNREST) dans la concaténation RPL CICS (instruction de définition de données DFHRPL) de la région de connexion primaire CICS. La méthode recommandée consiste à ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS. Notez que le module de chargement du gestionnaire de message de pipeline, ADNTMSGH, doit également être placé dans la concaténation RPL (voir «Gestionnaire de message de pipeline», à la page 79).
- Personnalisez et soumettez le travail ADNCSOWS pour mettre à jour la définition de système CICS (CSD) pour la région de connexion CICS primaire. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre. Notez que les ID transaction utilisés dans ce travail doivent correspondre à ceux utilisés par le gestionnaire de message de pipeline (qui peut avoir fait l'objet d'une personnalisation).



ADNCSDWS se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

- Utilisez la commande CEDA appropriée pour installer le groupe Gestionnaire de déploiement d'application pour cette région, par exemple :  
CEDA INSTALL GROUP(ADNPCRGP)

## régions de connexion CICS secondaires

Le serveur CRD peut également être utilisé avec une ou plusieurs régions de connexion secondaires supplémentaires, qui sont généralement des régions gérant les applications (AOR).

**Remarque :** Il n'est pas nécessaire de suivre cette procédure si CICSplex SM Business Application Services (BAS) est utilisé pour gérer vos définitions de ressource CICS.

- Placez les modules de chargement du gestionnaire de déploiement d'application FEK.SFEKLOAD(ADNCRD\*) dans la concaténation RPL CICS (instruction de définition de données DFHRPL) de ces régions de connexion non primaires. Pour ce faire, il est recommandé d'ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS.
- Personnalisez et soumettez un travail ADNCSDAR pour mettre à jour le CSD pour ces régions de connexion non primaires. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre. ADNCSDAR se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.
- Utilisez la commande CEDA appropriée pour installer le groupe Gestionnaire de déploiement d'application pour ces régions, par exemple :  
CEDA INSTALL GROUP(ADNARRGP)

---

## (Facultatif) Référentiel de manifestes

Developer for System z permet aux clients de parcourir et éventuellement de modifier les manifestes qui décrivent les ressources CICS sélectionnées. En fonction des droits d'accès définis par l'administrateur CICS, les modifications peuvent être effectuées directement dans le référentiel de manifestes ou exportées dans celui-ci afin d'être traitées ultérieurement par l'administrateur CICS.

**Remarque :**

- Cette étape n'est requise que pour les clients qui exportent des manifestes depuis Developer for System z en vue de leur traitement par l'outil de traitement des manifestes.
- L'outil de traitement de manifestes est un module d'extension pour IBM CICS Explorer.

Personnalisez et soumettez le travail ADNMFST pour allouer et initialiser le fichier VSAM du référentiel de manifestes et le définir dans la région de connexion CICS primaire. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre. Un référentiel de manifestes distinct

doit être créé pour chaque région de connexion primaire CICS. Tous les utilisateurs doivent posséder des droits d'accès UPDATE au référentiel de manifestes.

ADNVMFST se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

---

## Chapitre 5. (Facultatif) SCLM Developer Toolkit

SCLM Developer Toolkit fournit les outils nécessaires à l'extension des fonctions de SCLM sur le client. SCLM est lui-même un gestionnaire de code source hôte livré comme partie intégrante d'ISPF.

SCLM Developer Toolkit intègre un module d'extension reposant sur Eclipse qui sert d'interface avec SCLM et fournit l'accès à tous les processus SCLM de développement du code existant. Il assure également la prise en charge du développement intégral de Java et J2EE sur le poste de travail avec la synchronisation de SCLM sur le grand système, y compris la construction, l'assemblage et le déploiement du code J2EE à partir du grand système.

---

### Configuration requise et liste de contrôle

Vous aurez besoin de l'aide d'un administrateur SCLM et éventuellement d'un administrateur de sécurité pour effectuer cette tâche de personnalisation, qui requiert les tâches de personnalisation spéciales et/ou les ressources suivantes :

- Mises à jour APF et LINKLIST
- Définition de traducteurs de langage pour le support JAVA/J2EE
- Définition de types SCLM pour le support JAVA/J2EE
- (Facultatif) Règle de sécurité permettant aux utilisateurs de mettre à jour une méthode d'accès VSAM SCLM
- (Facultatif) Installation d'Ant

Pour commencer à utiliser SCLM Developer Toolkit sur votre site, vous devez effectuer les tâches ci-après. Sauf indication contraire, toutes les tâches sont obligatoires.

1. Vérifiez et adaptez les éléments prérequis et les mises à jour PARMLIB. Pour plus d'informations, voir «Conditions préalables», à la page 84.
2. Personnalisez les fichiers de configuration de Developer for System z. Pour plus d'informations, voir :
  - «Mises à jour d'ISPF.conf pour SCLMDT», à la page 84
  - «Mises à jour de rsed.envvars pour SCLMDT», à la page 85
3. Si vous le souhaitez, configurez le support de conversion de noms longs/courts. Pour plus d'informations, voir «(Facultatif) Conversion de nom long/court», à la page 86.
4. Vous pouvez installer et personnaliser Ant pour utiliser le support de génération JAVA/J2EE. Pour plus d'informations, voir «(Facultatif) Installation et personnalisation d'Ant», à la page 88.
5. Mettez à jour SCLM pour définir des parties propre à SCLMDT. Pour plus d'informations, voir «Mises à jour SCLM pour SCLMDT», à la page 89.
6. Si vous le souhaitez, vous pouvez configurer l'automatisation pour nettoyer régulièrement la zone de travail SCLMDT. Pour plus d'informations, voir «Suppression des fichiers antérieurs du répertoire WORKAREA», à la page 90.

---

## Conditions préalables

Pour obtenir la liste relative à la maintenance SCLM requise, voir Annexe E, «Conditions requises», à la page 339.

Cette annexe fournit également les spécifications Ant requises pour les générations JAVA/J2EE dans SCLM Developer Toolkit.

**Avertissement :** SCLM Developer Toolkit nécessite l'utilisation de la passerelle client TSO/ISPF d'ISPF et, par conséquent, z/OS version 1.8 ou supérieure est requis.

Comme décrit dans la section «Modifications de PARMLIB», à la page 16, SCLM Developer Toolkit requiert une personnalisation supplémentaire des paramètres système. Ces modifications sont les suivantes :

- (BPXPRMxx) Augmente le nombre maximal de processus par ID utilisateur z/OS UNIX.
- (PROGxx) APF autorise SYS1.LINKLIB et l'environnement d'exécution REXX, REXX.V1R4M0.SEAGLPA ou REXX.V1R4M0.SEAGALT.
- (PROGxx/LPALSTxx) Placez ISP.SISPLPA, ISP.SISPLPAD, SYS1.LINKLIB et l'environnement d'exécution REXX dans LINKLIST/LPALIB.

SCLM Developer Toolkit utilise également l'utilitaire SDSF ou la commande TSO **OUTPUT** pour extraire l'état d'achèvement et la sortie du travail. Voici quelques détails supplémentaires sur ces deux méthodes :

- L'utilitaire SDSF doit être commandé, installé et configuré séparément. Il requiert également l'utilisation de JES2.
- Les paramètres par défaut de la commande TSO **OUTPUT** permettent à l'utilisateur d'extraire la sortie de travail commençant par son ID utilisateur uniquement. Pour optimiser l'utilisation de la fonction **OUTPUT**, le modèle d'exit TSO/E IKJEFF53 doit être modifié pour qu'un utilisateur puisse extraire la sortie d'un travail dont il est propriétaire mais qui ne commence pas par cet ID utilisateur. Pour de plus amples informations sur cette sortie, voir le document *TSO/E Customization* (SA22-7783).

Les utilisateurs doivent détenir les droits d'accès READ, WRITE et EXECUTE aux répertoires z/OS UNIX /tmp/ and /var/rdz/WORKAREA/. Le répertoire WORKAREA/ se trouve dans /var/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

---

## Mises à jour d'ISPF.conf pour SCLMDT

SCLM Developer Toolkit utilise les squelettes ISPF/SCLM standard. Par conséquent, vous devez vous assurer que la bibliothèque de squelettes ISP.SISPSLIB est allouée à la concaténation ISPSLIB dans ISPF.conf. L'utilisation du fichier ISP.SISPSENU est facultative.

Le fichier ISPF.conf se trouve dans /etc/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**.

**Remarque :** Les modifications sont appliquées pour tous les clients qui se connectent à l'hôte après la mise à jour.

L'exemple de code suivant illustre le fichier `ISPF.conf` qui doit être personnalisé pour correspondre à votre environnement système. Les lignes mises en commentaire commencent par un astérisque (\*). Ajoutez des fichiers à la concaténation sur la même ligne et séparez les noms par une virgule (.). Pour plus d'informations sur la personnalisation d'`ISPF.conf`, voir la section «`ISPF.conf`, fichier de configuration de la passerelle client TSO/ISPF d'ISPF», à la page 47.

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

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

*Figure 21. Mises à jour d'`ISPF.conf` pour SCLMDT*

**Remarque :**

- Vous pouvez ajouter vos propres instructions de type définition de données et concaténations de fichiers pour personnaliser l'environnement TSO, ce qui permet de simuler une procédure d'ouverture de session TSO. Pour plus d'informations, voir le Chapitre 16, «Personnalisation de l'environnement TSO», à la page 269.
- Lors de la génération par lots, assurez-vous de concaténer la version personnalisée du squelette FLMLIBS avant la bibliothèque de squelettes ISPF/SCLM.

```
ispslib=h1q.USERSKEL,ISP.SISPSLIB
```

---

## Mises à jour de `rsed.envvars` pour SCLMDT

SCLM Developer Toolkit utilise certaines directives définies dans `rsed.envvars` pour localiser les fichiers et les répertoires.

Le fichier `rsed.envvars` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO `OEDIT`.

**Remarque :** Vous devez relancer la tâche démarrée `RSED` pour appliquer les modifications apportées.

L'exemple de code suivant illustre les directives SCLMDT dans `rsed.envvars` qui doivent être personnalisées pour correspondre à votre environnement système. Voir «`rsed.envvars` - Fichier de configuration RSE», à la page 32 pour obtenir de plus amples informations sur la personnalisation de `rsed.envvars`.

```

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

```

Figure 22. Mises à jour de *rsed.envvars* pour SCLMDT

---

## (Facultatif) Conversion de nom long/court

SCLM Developer Toolkit permet d'enregistrer dans SCLM des fichiers dont le nom est long (comportant plus de 8 caractères ou une casse mixte). Cela s'effectue à l'aide d'un fichier VSAM contenant le mappage du nom de fichier long sur le nom de membre à 8 caractères utilisé dans SCLM.

### Remarque :

- Pour les versions antérieures à z/OS 1.8, cette fonction est fournie via une modification provisoire du logiciel ISPF/SCLM de base relative à l'APAR OA11426.
- La conversion de nom long/abrégé est également utilisée par d'autres produits associés à SCLM (IBM SCLM Administrator Toolkit, par exemple).

## Création du fichier LSTRANS.FILE, VSAM de conversion des noms longs/courts

Personnalisez et soumettez un exemple de membre FLM02LST dans l'exemple de bibliothèque ISPF ISP.SISPSAMP, pour créer la conversion de nom long/abrégé VSAM. Les étapes de configuration dans la présente publication s'attendent à ce que VSAM s'appelle FEK.#CUST.LSTRANS.FILE, comme illustré dans l'exemple de JCL de configuration.

```

//FLM02LST JOB <paramètres du travail>
//*
/* CAUTION: This is neither a JCL procedure nor a complete job.
/* Before using this sample, you will have to make the following
/* modifications:
/* 1. Change the job parameters to meet your system requirements.
/* 2. Change ***** to the volume that will hold the VSAM.
/* 3. Change all references of FEK.#CUST.LSTRANS.FILE to
/*     match your naming convention for the SCLM translate VSAM.
/*
//CREATE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE FEK.#CUST.LSTRANS.FILE
SET MAXCC=0
DEFINE CLUSTER(NAME(FEK.#CUST.LSTRANS.FILE) -
               VOLUMES(*****)) -
               RECORDSIZE(58 2048) -
               SHAREOPTIONS(3 3) -
               CYLINDERS(1 1) -
               KEYS(8 0) -
               INDEXED) -
DATA (NAME(FEK.#CUST.LSTRANS.FILE.DATA)) -
INDEX (NAME(FEK.#CUST.LSTRANS.FILE.INDEX))

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

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

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

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

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

```

Figure 23. FLM02LST - JCL de configuration de la conversion de nom long/abrégé

**Remarque :** Les utilisateurs doivent disposer de droit d'accès de mise à jour (UPDATE) sur le fichier VSAM, comme décrit dans le Chapitre 10, «Remarques relatives à la sécurité», à la page 161.



## Mises à jour de rsed.envvars pour la conversion de noms longs/courts

Avant d'utiliser la conversion de noms longs/courts, supprimez la mise en commentaire et définissez la variable d'environnement `_SCLMDT_TRANTABLE` de `rsed.envvars` pour rechercher le nom de la méthode VSAM de conversion de noms longs/courts.

Le fichier `rsed.envvars` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**.

**Remarque :** Vous devez relancer la tâche démarrée RSED pour appliquer les modifications apportées.

---

## (Facultatif) Installation et personnalisation d'Ant

Cette étape est requise uniquement si vous projetez d'utiliser la prise en charge de la génération JAVA/J2EE dans SCLM.

Apache Ant est un outil de génération Java Open Source qui peut être téléchargé à partir du site <http://ant.apache.org/>. Ant est constitué de fichiers texte et de scripts distribués au format ASCII, et qui requièrent donc l'exécution d'une conversion ASCII/EBCDIC dans z/OS UNIX.

Effectuez les étapes suivantes afin de mettre en oeuvre Ant sur z/OS et de le définir dans Developer for System z:

- Téléchargez le dernier fichier Ant compressé au format binaire dans le système de fichiers z/OS UNIX. Il est recommandé de télécharger la version .zip d'ANT pour éviter les inconvénients que peut générer dans z/OS la décompression de versions ayant le format de suffixe `tar.gz` ou `tar.bz2`.
- Ouvrez une session de ligne de commande z/OS UNIX pour continuer l'installation (avec la commande **TSO OMVS**, par exemple).
- Créez un répertoire `home` pour l'installation du programme Ant à l'aide de la commande **mkdir -p /home-dir** et accédez à ce répertoire en utilisant la commande **cd /home-dir**.
- Utilisez la commande d'extraction JAR **jar -xf apache-ant-1.7.1.zip** pour décompresser le fichier dans le répertoire en cours. Un répertoire bin Java doit se trouver dans la variable locale `PATH` de z/OS UNIX pour pouvoir utiliser la commande **jar**. Sinon, indiquez un chemin qualifié complet pour la commande à l'aide de l'emplacement bin Java (**/usr/lpp/java/J5.0/bin/jar -xf apache-ant-1.7.1.zip**, par exemple).
- Convertissez tous les fichiers texte Ant en EBCDIC en exécutant (et en personnalisant éventuellement) l'exemple de script `/usr/lpp/rdz/samples/BWBTRANT`.

**Remarque :** N'exécutez ce script qu'une seule fois. Plusieurs exécutions entraînerait l'altération de l'installation Ant.

- Pour vérifier la bonne exécution de la conversion, localisez et parcourez un fichier texte dans le répertoire ANT, tel que `apache-ant-1.7.1/README`. Si le fichier est lisible, cela signifie que la conversion a réussi.

- Utilisez la commande **chmod -R 755 \*** pour permettre aux utilisateurs de LIRE et d'EXECUTER des fichiers dans le répertoire ANT.
- Pour pouvoir utiliser Ant, vous devez définir les variables d'environnement `rsed.envvars` `JAVA_HOME` et `ANT_HOME`.
  - La variable `JAVA_HOME` est requise pour pointer vers le répertoire de base Java, par exemple :  
`JAVA_HOME=/usr/lpp/java/IBM/J5.0`
  - `ANT_HOME` est requis pour pointer sur le répertoire initial d'Ant, par exemple:  
`ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1`

Par exemple :

- `TSO OMVS`
- `mkdir -p /usr/lpp/Apache/Ant`
- `cd /usr/lpp/Apache/Ant`
- `jar -xf /u/userid/apache-ant-1.7.1`
- `/usr/lpp/rdz/samples/BWBTRANT`
- `cat ./apache-ant-1.7.1/README`
- `chmod -R 755 *`
- `oedit /etc/rsed.envvars`

Pour tester la réussite de l'initialisation d'Ant, procédez comme suit :

- Ajoutez les répertoires bin Ant et Java à la variable d'environnement `PATH`.

Exemple :

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

- Exécutez Ant **ant -version** pour d'afficher la version, si le programme a été correctement installé.

Exemple :

```
ant -version
```

**Remarque :** Il est nécessaire de définir l'instruction `PATH` de cette manière à des fins de test uniquement et non pour une utilisation opérationnelle.

---

## Mises à jour SCLM pour SCLMDT

SCLM requiert également que la personnalisation fonctionne avec SCLM Developer Toolkit. Pour plus d'informations sur les tâches de personnalisation requises, voir le document *IBM Rational Developer for System z SCLM Developer Toolkit - Guide d'administration* (SC11-6464) :

- Définition de traducteurs de langage pour le support JAVA/J2EE
- Définition de types SCLM pour le support JAVA/J2EE

Pour effectuer des tâches de personnalisation et de définition de projet, l'administrateur SCLM doit connaître plusieurs valeurs personnalisables de Developer for System z, tel que décrit dans le tableau 13, à la page 90.

Tableau 13. Liste de contrôle de l'administrateur SCLM

Description	<ul style="list-style-type: none"> <li>Valeur par défaut</li> <li>Emplacement de la réponse</li> </ul>	Valeur
Exemple de bibliothèque Developer for System z	<ul style="list-style-type: none"> <li>FEK.SFEKSAMV</li> <li>Installation SMP/E</li> </ul>	
Exemple de répertoire Developer for System z	<ul style="list-style-type: none"> <li>/usr/lpp/rdz/samples</li> <li>Installation SMP/E</li> </ul>	
Répertoire bin Java	<ul style="list-style-type: none"> <li>/usr/lpp/java/J5.0/bin</li> <li>rsed.envvars - \$JAVA_HOME/bin</li> </ul>	
Répertoire bin Ant	<ul style="list-style-type: none"> <li>/usr/lpp/Apache/Ant/apache-ant-1.7.1/bin</li> <li>rsed.envvars - \$ANT_HOME/bin</li> </ul>	
Répertoire initial WORKAREA	<ul style="list-style-type: none"> <li>/var/rdz</li> <li>rsed.envvars - \$_CMDSERV_CONF_HOME</li> </ul>	
Répertoire initial de configuration de projet SCLMDT	<ul style="list-style-type: none"> <li>/var/rdz/sclmdt</li> <li>rsed.envvars - \$_SCLMDT_CONF_HOME</li> </ul>	
Méthode d'accès VSAM de conversion de nom long/court	<ul style="list-style-type: none"> <li>FEK.#CUST.LSTRANS.FILE</li> <li>rsed.envvars - \$_SCLMDT_TRANTABLE</li> </ul>	

## Suppression des fichiers antérieurs du répertoire WORKAREA

SCLM Developer Toolkit et la passerelle client TSO/ISPF d'ISPF partagent le même répertoire WORKAREA, dans lequel il peut s'avérer utile de procéder à un nettoyage périodique. Voir «(Facultatif) Nettoyage du répertoire WORKAREA», à la page 110 pour de plus amples informations.

---

## Chapitre 6. (Facultatif) Tâches de personnalisation alternatives

La présente section regroupe diverses tâches de personnalisation facultatives. Suivez les instructions de la section adéquate pour configurer le service souhaité.

- «(Facultatif) Procédure mémorisée DB2»
- «(Facultatif) Support EST (Enterprise Service Tools)», à la page 93
- «(Facultatif) Support de langue bidirectionnelle CICS», à la page 94
- «(Facultatif) Messages d'erreur de diagnostic IRZ», à la page 95
- «(Facultatif) Chiffrement SSL RSE», à la page 95
- «(Facultatif) Fonction de trace RSE», à la page 98
- «(Facultatif) Groupes de propriétés basés sur l'hôte», à la page 100
- «(Facultatif) Projets basés sur l'hôte», à la page 101
- «(Facultatif) Intégration de File Manager», à la page 102
- «(Facultatif) Caractères non éditables», à la page 103
- «(Facultatif) Utilisation de REXEC (ou SSH)», à la page 104
- «(Facultatif) Transaction APPC pour le service Commandes TSO», à la page 106
- «(Facultatif) Nettoyage du répertoire WORKAREA», à la page 110

---

### (Facultatif) Procédure mémorisée DB2

Vous aurez besoin de l'aide d'un administrateur WLM et d'un administrateur DB2 pour effectuer ces tâches de personnalisation qui requièrent les ressources ou les tâches de personnalisation spécifiques suivantes :

- Mise à jour WLM
- Nouveau membre PROCLIB
- Mise à jour DB2

---

Developer for System z propose un modèle de procédure mémorisée DB2 (compilateur de procédure mémorisée PL/I et COBOL) permettant de générer des procédures mémorisées COBOL et PL/I dans le client Developer for System z.

**Remarque :** Les exemples de membre ELAXM\* se trouvent dans FEK.#CUST.JCL et FEK.#CUST.PROCLIB, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

### Modifications de Workload Manager (WLM)

Au moyen des panneaux de gestion de charge de travail (WLM), associez un environnement d'application à la procédure JCL de l'espace adresse WLM du compilateur de procédures mémorisées PL/I et COBOL. Voir le document *MVS Planning Workload Management* (SA22-7602) pour de plus amples informations sur cette opération.

**Remarque :** Vous pouvez créer un nouvel environnement d'applications dans WLM pour le compilateur de procédures mémorisées PL/I et COBOL, ou ajouter les définitions nécessaires à un environnement existant.

## Modifications de PROCLIB

Personnalisez l'exemple de tâche de procédure mémorisée FEK.#CUST.PROCLIB(ELAXMSAM), comme décrit dans le membre, et copiez-le dans SYS1.PROCLIB. Comme indiqué dans l'exemple de code suivant, vous devez spécifier les éléments suivants :

- Le nom de l'environnement d'application défini dans WLM pour cette procédure mémorisée
- Le nom du sous-système DB2
- Le qualificatif de haut niveau des différents fichiers

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

Figure 24. ELAXMSAM - Tâche de procédure mémorisée DB2

### Remarque :

- La procédure mémorisée DB2 utilise la commande exec ELAXMREX REXX qui se trouve dans FEK.SFEKPROC. Ne modifiez pas cet emplacement si vous souhaitez avoir la possibilité d'activer automatiquement une opération de maintenance SMP/E.
- Si vous souhaitez renommer les membres ELAXMSAM ou ELAXMREX, voir le Chapitre 17, «Exécution de plusieurs instances», à la page 277.

## Modifications de DB2

Personnalisez et soumettez l'exemple de membre ELAXMJCL du fichier FEK.#CUST.JCL afin de définir la procédure mémorisée sur DB2. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre.

```

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

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

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

Figure 25. ELAXMJCL – Définition d'une procédure mémorisée DB2

**Remarque :** Vérifiez que la clause WLM ENVIRONMENT de l'instruction CREATE PROCEDURE spécifie le nom de la procédure d'environnement WLM qui a été défini pour le compilateur de procédures mémorisées PL/I et COBOL (ELAXMSAM, par défaut).

---

## (Facultatif) Support EST (Enterprise Service Tools)

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

---

Le client Developer for System z contient un composant de génération de code appelé Enterprise Service Tools (EST). Selon le type de code généré, ce code s'appuie sur les fonctions fournies par l'installation de l'hôte Developer for System z. Les sections suivantes expliquent comment rendre ces fonctions de l'hôte disponibles :

- Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75
- «(Facultatif) Support de langue bidirectionnelle CICS», à la page 94
- «(Facultatif) Messages d'erreur de diagnostic IRZ», à la page 95

**Remarque :** Enterprise Service Tools (EST) regroupe plusieurs outils (le modélisateur de flux de services (SFM) et les services XML pour l'entreprise (XSE), par exemple).

---

## (Facultatif) Support de langue bidirectionnelle CICS

Vous aurez besoin de l'aide d'un administrateur CICS pour effectuer cette tâche de personnalisation qui requiert les ressources ou les tâches de personnalisation spécifiques suivantes :

- Mise à jour du JCL de la région CICS
  - Définition d'un programme dans CICS
- 

Les composants de l'outil EST (Developer for System z Enterprise Service Tools) prennent en charge différents formats de messages d'interface en arabe et en hébreu, ainsi que la présentation et l'édition des données bidirectionnelles dans tous les éditeurs et dans toutes les vues. Dans les applications de terminal, les écrans de gauche à droite et de droite à gauche sont pris en charge, ainsi que les zones numériques et les zones orientées dans le sens contraire de l'écran.

Les fonctions et fonctionnalités bidirectionnelles supplémentaires comprennent notamment :

- La spécification dynamique par le demandeur de service EST, des attributs bidirectionnels des messages d'interface.
- Le traitement des données bidirectionnelles dans les flux de services, est basé sur des attributs bidirectionnels (type de texte, orientation du texte, permutation numérique et permutation symétrique). Ces attributs peuvent être spécifiés à différentes étapes de la création des flux, à la fois pour les flux d'interface et de terminal.
- Le code d'exécution généré par EST comprend la conversion des données entre des zones de message dont les attributs bidirectionnels sont différents.

De plus, le code généré par EST peut prendre en charge la transformation bidirectionnelle dans d'autres environnements que SFR CICS (Service Flow Runtime). Les applications par lots en sont un exemple. Vous pouvez inclure dans les générateurs EST des appels de routines de conversion bidirectionnelle en spécifiant les options de transformation bidi appropriées dans les assistants de génération EST et en éditant des liens entre les programmes générés et la bibliothèque de conversion bidirectionnelle appropriée, FEK.SFEKLOAD.

Appliquez la procédure suivante pour activer la prise en charge de la langue bidirectionnelle CICS :

1. Placez les modules de chargement FEK.SFEKLOAD, FEJBDCMP et FEJBDTRX dans la concaténation RPL CICS (instruction de définition de données DFHRPL). Pour ce faire , il est recommandé d'ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS.

**Remarque :** Si vous ne concaténez pas le fichier d'installation mais copiez les modules dans un fichier nouveau ou existant, notez que ces modules sont des bibliothèques DLL qui DOIVENT résider dans une bibliothèque PDSE.

2. Définissez FEJBDCMP et FEJBDTRX en tant que programmes pour que CICS utilise la commande CEDA appropriée. Par exemple:



---

## (Facultatif) Messages d'erreur de diagnostic IRZ

Vous n'avez pas besoin d'aide pour effectuer cette tâche de personnalisation, mais des ressources ou tâches de personnalisation spéciales ci-dessous :

- Mise à jour LINKLIST
  - Mettre à jour le langage de contrôle des travaux de la région CICS
- 

Le client Developer for System z contient un composant de génération de code appelé Enterprise Service Tools (EST). Pour que le code généré émette des messages d'erreur de diagnostic, tous les modules IRZ\* et IIRZ\* de la bibliothèque de charge FEK.SFEKLOAD doivent être mis à sa disposition. EST peut générer un code pour les environnements suivants :

- CICS
- IMS
- Lot MVS

Lorsque le code généré est exécuté dans une transaction CICS, ajoutez tous les modules IRZ\* et IIRZ\* de FEK.SFEKLOAD dans la définition de données DFHRPL de la région CICS. Pour ce faire , il est recommandé d'ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS.

Dans toutes les autres situations, mettez à disposition tous les modules IRZ\* et IIRZ\* de FEK.SFEKLOAD par l'intermédiaire de STEPLIB ou de LINKLIST. Pour ce faire , il est recommandé d'ajouter le fichier d'installation à la concaténation pour que l'opération de maintenance appliquée soit automatiquement disponible dans CICS.

Si vous décidez d'utiliser STEPLIB, vous devez définir les modules non disponibles via LINKLIST dans la directive STEPLIB de la tâche qui exécute le code.

Si les modules de chargement ne sont pas disponibles et que le code généré rencontre une erreur, le message suivant s'affiche :

IRZ9999S L'extraction du texte d'un message d'exécution Language Environment a échoué. Vérifiez que le module de messages d'exécution Language Environment pour la fonction IRZ est installé dans DFHRPL ou STEPLIB.

---

## (Facultatif) Chiffrement SSL RSE

Vous aurez besoin de l'aide d'un administrateur de sécurité pour effectuer cette tâche de personnalisation, qui requiert les tâches de personnalisation spéciales ou les ressources suivantes :

- Mise à jour LINKLIST
  - Règle de sécurité pour l'ajout de fichiers contrôlés par un programme
  - (Facultatif) Règle de sécurité pour l'ajout d'un certificat pour le SSL
- 

Les communications externes (client-hôte) peuvent être chiffrées à l'aide de SSL (Secure Socket Layer). Cette fonction est désactivée par défaut et est contrôlée par les paramètres du fichier ssl.properties.

Le fichier `ssl.properties` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Notez que RSE doit être redémarré pour que les modifications prennent effet.

Le client communique avec le démon RSE pendant la configuration de la connexion et avec le serveur RSE pendant la session réelle. Les deux flux de données sont chiffrés lorsque SSL est activé.

Le démon RSE et le serveur RSE prennent en charge des mécanismes différents pour stocker des certificats en raison leurs différences architecturales. Cela signifie que des définitions SSL sont nécessaires pour le démon et le serveur RSE. Un certificat partagé peut être utilisé si le démon et le serveur RSE utilisent la même méthode de gestion des certificats.

*Tableau 14. Mécanismes de stockage des certificats SSL*

Stockage des certificats	Créé et géré par	Démon RSE	Serveur RSE
Fichier de clés	Produit de sécurité compatible avec SAF	pris en charge	pris en charge
Base de données de clés	gskkyman de z/OS UNIX	pris en charge	/
Magasin de clés	Outil de clé de Java	/	pris en charge

**Remarque :**

- Il est conseillé d'utiliser des fichiers de clés conformes à SAF pour la gestion des certificats.
- Les fichiers de clés conformes à SAF peuvent stocker la clé privée du certificat dans la base de données de sécurité ou en utilisant ICSF, l'interface vers le matériel de chiffrement de System z. L'accès à ICSF est protégé par des profils dans la classe de sécurité CSFSERV.

Le démon RSE utilise les fonctions System SSL pour gérer SSL. Cela signifie que `SYS1.SIEALNKE` doit être contrôlé par programme via le logiciel de sécurité et être à la disposition de RSE via `LINKLIST` ou la directive `STEPLIB` dans `rsed.envvars`.

L'exemple de code suivant présente le fichier `ssl.properties` qui doit être personnalisé en fonction de votre environnement système. Les lignes mises en commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne. La continuation de ligne n'est pas prise en charge.

```
# ssl.properties – fichier de configuration SSL
enable_ssl=false

# Daemon Properties

#daemon_keydb_file=
#daemon_keydb_password=
#daemon_key_label=

# Server Properties

#server_keystore_file=
#server_keystore_password=
#server_keystore_label=
#server_keystore_type=JCECERACFKS
```

Figure 26. *ssl.properties* – Fichier de configuration RSE

Les propriétés du serveur et du démon doivent être configurées uniquement si vous activez la couche SSL. Pour plus d'informations sur la configuration de SSL, voir l'Annexe A, «Configuration de l'authentification SSL et X.509», à la page 299.

#### **enable\_ssl**

Active ou désactive les communications SSL. La valeur par défaut est `false`. Les seules options valides sont `true` et `false`.

#### **daemon\_keydb\_file**

Nom du fichier de clés RACF (ou produit de sécurité similaire). Indiquez le nom de la base de données de clés si vous avez utilisé **gskkyman** pour créer une base de données de clés au lieu d'utiliser un fichier de clés. Supprimez la mise en commentaire et personnalisez cette directive si la couche SSL est activée.

#### **daemon\_keydb\_password**

Laissez ce paramètre en commentaire ou à blanc si vous utilisez un fichier de clés. Sinon, fournissez le mot de passe de la base de données de clés. Supprimez la mise en commentaire et personnalisez cette directive si SSL est activé et si vous n'utilisez pas la base de données de clés **gskkyman**.

#### **daemon\_key\_label**

Libellé de certificat utilisé dans le fichier de données ou la base de données de clés, s'il n'est pas défini par défaut. Doit être mis en commentaire si la valeur par défaut est utilisée. Supprimez la mise en commentaire et personnalisez cette directive si SSL est activé et si vous n'utilisez pas le certificat de sécurité par défaut.

#### **server\_keystore\_file**

Nom du fichier de clés créé par la commande **keytool** de Java ou du fichier de clés RACF (ou produit de sécurité analogue) si `server_keystore_type=JCECERACFKS`. Supprimez la mise en commentaire et personnalisez cette directive si la couche SSL est activée.

#### **server\_keystore\_password**

Laissez ce paramètre en commentaire ou à blanc si vous utilisez un fichier de clés. Sinon, fournissez le mot de passe du magasin de clés. Supprimez la mise en commentaire et personnalisez cette directive si SSL est activé et si vous n'utilisez pas un magasin de clés **keytool**.

#### **server\_keystore\_label**

Libellé du certificat utilisé dans le fichier ou la base de données de clés, s'il n'est pas défini par défaut. La valeur par défaut correspond au premier

certificat valide détecté. Supprimez la mise en commentaire et personnalisez cette directive si SSL est activé et si vous n'utilisez pas le certificat de sécurité par défaut.

#### **server\_keystore\_type**

Type de magasin de clés. La valeur par défaut est JKS. Les valeurs acceptées sont les suivantes :

*Tableau 15. Types de fichier de clés valides*

Mot clé	Type de fichier de clés
JKS	Fichier de clés Java
JCERACFKS	Fichier de clés conforme à SAF, où la clé privé du certificat est stockée dans la base de données de sécurité.
JCECCARACFKS	Fichier de clés conforme à SAF dans lequel la clé privée du certificat est stockée à l'aide d'ICSF, l'interface du matériel de chiffrement de System z.

**Remarque :** Au moment de la publication, IBM z/OS Java implique de mettre à jour le fichier /usr/lpp/java/J5.0/lib/security/java.security afin de prendre en charge JCECCARACFKS. La ligne suivante doit être ajoutée :

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

Le fichier de résultats doit ressembler à ce qui suit :

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

---

## **(Facultatif) Fonction de trace RSE**

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

---

Developer for System z prend en charge différents niveaux de trace du flux de programme interne à des fins de résolution des incidents. RSE et certains des services appelés par RSE, utilisent les paramètres du fichier rsecomm.properties pour obtenir le niveau de détails voulu dans les fichiers journaux de sortie.

**Avertissement :** La modification de ces paramètres réduit les performances et ne doit être effectuée que sur indication du centre de support IBM.

Le fichier rsecomm.properties se trouve dans /etc/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**.

L'exemple de code suivant illustre le fichier rsecomm.properties qui peut être personnalisé pour répondre à vos besoins de traçage. Les lignes de commentaire

commencent par un signe dièse (#) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne. La continuation de ligne n'est pas prise en charge.

```
# server.version - DO NOT MODIFY!
server.version=5.0.0

# Logging level
# 0 - Log error messages
# 1 - Log error and warning messages
# 2 - Log error, warning and info messages
debug_level=1

# Log location
# Log_To_StdOut
# Log_To_File
log_location=Log_To_File
```

Figure 27. *rsecomm.properties* – Fichier de configuration de consignation

#### server.version

Version du serveur de consignation. La valeur par défaut est 5.0.0. Ne pas modifier.

#### debug\_level

Niveau de détail pour les fichiers journaux de sortie. La valeur par défaut est 1 (messages d'erreurs de consignation et d'avertissement). Notez que `debug_level` permet de définir le niveau de détails de plusieurs services (ce qui génère plusieurs fichiers de sortie). L'augmentation du niveau de détails entraîne une altération des performances et doit être appliquée uniquement sur instruction du centre de support technique IBM. Pour plus d'informations sur les journaux contrôlés par cette directive, voir «Fonction de trace RSE», à la page 148.

Les valeurs acceptées sont les suivantes :

0	Messages d'erreurs de consignation uniquement.
1	Messages d'erreurs de consignation et d'avertissement.
2	Messages d'erreurs de consignation, d'avertissement et d'information.

**Remarque :** `debug_level` peut être modifié de manière dynamique avec la commande de l'opérateur **modify rsecommlog**, comme indiqué à la section Chapitre 8, «Commandes de l'opérateur», à la page 127.

#### log\_location

Support de sortie pour la consignation liée à RSE. La valeur par défaut est `Log_To_File`. Ne modifiez pas ce paramètre si vous utilisez la méthode de connexion du démon RSE (valeur par défaut), sauf instruction contraire du centre de support technique IBM.

Les valeurs acceptées sont les suivantes :

Log_To_File	<p>Envoie les messages de consignation dans un fichier distinct du répertoire de sortie de la consignation.</p> <ul style="list-style-type: none"> <li>Démon RSE : <code>rsedaemon.log</code> dans <code>daemonlog</code></li> <li>Pools d'unités d'exécution RSE : <code>rserver.log</code> dans <code>daemonlog</code></li> <li>Utilisateur : <code>rsecomm.log</code> in <code>userlog/.eclipse/RSE/\$LOGNAME</code></li> </ul>
-------------	--

Log_To_StdOut	<p>Envoie les messages de consignation à stdout.</p> <ul style="list-style-type: none"> <li>• Démon RSE : Réacheminé vers DD STDOUT dans la tâche démarrée RSED</li> <li>• Pools d'unités d'exécution RSE : Indéfini</li> <li>• Utilisateur : redirigé vers stdout.log dans userlog/.eclipse/RSE/\$LOGNAME</li> </ul>
---------------	---

daemonlog est la valeur de la directive daemon.log dans rsed.envvars. Si la directive daemon.log est mise en commentaire ou omise, le chemin du répertoire de base de l'ID utilisateur affecté à la tâche démarrée RSED est utilisé. Le chemin du répertoire de base est défini dans le segment de sécurité OMVS de l'ID utilisateur.

Les journaux propres à l'utilisateur sont placés dans userlog/.eclipse/RSE/\$LOGNAME, où userlog est la valeur de la directive user.log dans rsed.envvars et \$LOGNAME l'ID utilisateur de connexion (en majuscules). Si la directive user.log est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur.

---

## (Facultatif) Groupes de propriétés basés sur l'hôte

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

---

Les clients Developer for System z peuvent définir des groupes de propriétés qui contiennent des valeurs par défaut pour diverses propriétés (par exemple, les options de compilation COBOL à utiliser lorsque vous compilez le code source COBOL). Certaines valeurs par défaut sont intégrées à Developer for System z, mais il est également possible de définir des valeurs par défaut personnalisées spécifiques au système.

L'emplacement du groupe de propriétés personnalisé et des fichiers de configuration de valeur par défaut est défini dans propertiescfg.properties, qui se trouve dans /etc/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Notez que RSE doit être redémarré pour que les modifications prennent effet.

L'exemple de code suivant illustre le fichier propertiescfg.properties qui doit être personnalisé pour correspondre à votre environnement système. Les lignes mises en commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne. La continuation de ligne n'est pas prise en charge.

```
#
# host based property groups - root configuration file
#
ENABLED=FALSE
RDZ-VERSION=7.5.0.0
PROPERTY-GROUP=/var/rdz/properties
DEFAULT-VALUES=/var/rdz/properties
```

*Figure 28. propertiescfg.properties - Fichier de configuration des groupes de propriétés basés sur l'hôte*

### ENABLED

Indique si Developer for System z utilisera le groupe de propriétés et les fichiers de configuration de valeur par défaut. La valeur par défaut est FALSE. Les seules options acceptées sont TRUE et FALSE.

### RDZ-VERSION

Niveau client minimal de Developer for System z à utiliser pour les groupes de propriétés basés sur l'hôte. Le niveau par défaut est 7.5.0.0. Ne pas modifier.

### PROPERTY-GROUP

Emplacement du fichier de configuration de groupe de propriétés. L'emplacement par défaut est /var/rdz/properties.

### DEFAULT-VALUES

Emplacement du fichier de configuration de valeur par défaut. L'emplacement par défaut est /var/rdz/properties.

Pour plus d'informations sur la création du fichier de configuration des groupes de propriétés (propertygroups.xml) et du fichier de configuration des valeurs par défaut (defaultvalues.xml), voir le centre de documentation de Developer for System z (<http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp>).

---

## (Facultatif) Projets basés sur l'hôte

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

---

Vous pouvez définir des projets z/OS individuellement via la perspective Projets z/OS du client ou de manière centrale sur l'hôte et les envoyer au client sur une base 'Par utilisateur'. Ces "projets basés sur l'hôte" ressemblent et fonctionnent exactement comme des projets définis sur le client, sauf que leurs structure, membres et propriétés ne sont pas modifiables par le client et qu'ils sont accessibles uniquement lorsque ce dernier est connecté à l'hôte.

L'emplacement des définitions de projet est défini dans le fichier projectcfg.properties qui se trouve dans /etc/rdz/, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Notez que RSE doit être redémarré pour que les modifications prennent effet.

L'exemple de code suivant illustre le fichier projectcfg.properties qui doit être personnalisé pour correspondre à votre environnement système. Les lignes mises en commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne. La continuation de ligne n'est pas prise en charge.



```
#
# host based projects – root configuration file
#
# WSED-VERSION – do not modify !
WSED-VERSION=7.0.0.0
# specify the location of the host based project definition files
PROJECT-HOME=/var/rdz/projects
```

Figure 29. *projectcfg.properties* – Fichier de configuration de projets basé sur l'hôte

#### WSED-VERSION

Niveau client minimal de Developer for System z à utiliser pour les projets basés sur l'hôte. La valeur par défaut est 7.0.0.0. Ne pas modifier.

#### PROJECT-HOME

Répertoire de base pour les définitions de projet. La valeur par défaut est /var/rdz/projects.

**Remarque :** Afin d'activer les projets basés sur l'hôte, un fichier `project.instance` doit exister dans `/var/rdz/projects/IDUTILISATEUR`, où `/var/rdz/projects` est l'emplacement des fichiers de définition de projet et `IDUTILISATEUR` l'ID utilisateur (en majuscules) avec lequel le développeur se connecte.

Pour plus d'informations sur les projets basés sur l'hôte, voir le centre de documentation de Developer for System z (<http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp>).

---

## (Facultatif) Intégration de File Manager

Vous aurez besoin de l'aide d'un administrateur de la sécurité pour effectuer cette tâche de personnalisation qui requiert les ressources ou les tâches de personnalisation spécifiques suivantes :

- Règle de sécurité pour l'ajout de fichiers contrôlés par un programme
- 

Developer for System z prend en charge l'accès direct du client à un ensemble limité de fonctions d'IBM File Manager for z/OS. IBM File Manager for z/OS met à disposition des outils pour travailler avec les fichiers MVS, les fichiers z/OS UNIX, les données DB2, IMS et CICS. Ces outils comportent les fonctionnalités habituelles de navigation, d'édition, de copie et d'impression que l'on trouve dans ISPF, améliorées afin de répondre aux besoins des développeurs d'applications. Dans la version en cours de Developer for System z, prend uniquement en charge la visualisation et l'édition des fichiers MVS (y compris tous les types de VSAM), la création et l'édition des modèles de fichiers MVS (dont les modèles dynamiques), ainsi que les utilitaires de copie avancée.

Notez que le produit IBM File Manager for z/OS doit être commandé, installé et configuré séparément. Voir le document *Rational Developer for System z Prerequisites* (SC23-7659) pour connaître le niveau de File Manager requis pour votre version de Developer for System z. L'installation et la personnalisation de ce produit ne sont pas décrites dans le présent ouvrage.

Notez que Developer for System z et File Manager ne prennent plus en charge l'interface de traitement par lots pour accéder aux services de File Manager. Il est désormais obligatoire d'utiliser le programme d'écoute de File Manager.

**Remarque :** Outre les tâches de configuration du programme d'écoute standard décrites dans votre documentation File Manager, Developer for System z requiert que le fichier STEPLIB du serveur soit contrôlé par un programme.

Les définitions File Manager Integration requises par Developer for System z sont stockées dans le fichier `FMIEXT.properties` disponible dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Notez que RSE doit être redémarré pour que les modifications prennent effet.

L'exemple de code suivant illustre le fichier `FMIEXT.properties` qui doit être personnalisé pour correspondre à votre environnement système. Les lignes mises en commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US. Les lignes de données ne peuvent comporter qu'une directive et la valeur associée. Les commentaires ne sont pas autorisés sur la même ligne. La continuation de ligne n'est pas prise en charge.

```
# File Manager Integration (FMI) Extension properties
#
enabled=false
fmlistenport=1960
```

*Figure 30. FMIEXT.properties – Fichier de configuration de File Manager*

**enabled**

Indique si le programme d'écoute de File Manager est disponible sur le même hôte. La valeur par défaut est `false`. Les seules valeurs admises sont `true` et `false`.

**fmlistenport**

Port utilisé par le programme d'écoute de File Manager. Le port par défaut est 1960. La communication sur ce port ne concerne que la machine hôte.

---

## (Facultatif) Caractères non éditables

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

---

Certains caractères ne sont pas convertis correctement entre les pages de codes hôte (EBCDIC) et les pages de codes client (ASCII). L'éditeur client Developer for System z utilise les définitions du fichier `uchars.settings` afin d'identifier ces caractères non éditables. Lors de l'ouverture d'un fichier avec un caractère identifié dans `uchars.settings`, l'éditeur impose le mode lecture seule, pour éviter la corruption du fichier lors de sa sauvegarde.

Le fichier `uchars.settings` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Notez que RSE doit être redémarré pour que les modifications prennent effet. Il est recommandé de ne pas modifier ce fichier à moins d'y être invité par le centre de service technique IBM.

```
# uchars.settings - uneditable code points
#
*          *          0D 15 25;

# DBCS (Japonais, Coréen & Chinois)
IBM-930   *          0D 15 1E 1F 25;
IBM-933   *          0D 15 1E 1F 25;
IBM-935   *          0D 15 1E 1F 25;
IBM-937   *          0D 15 1E 1F 25;
IBM-939   *          0D 15 1E 1F 25;
IBM-1390  *          0D 15 1E 1F 25;
IBM-1399  *          0D 15 1E 1F 25;
IBM-1364  *          0D 15 1E 1F 25;
IBM-1371  *          0D 15 1E 1F 25;
IBM-1388  *          0D 15 1E 1F 25;

# UNICODE
UTF-8     *          0D 0A;
UTF-16BE  *          0D 0A;
UTF-16LE  *          0D 0A;
UTF-16    *          0D 0A;
```

Figure 31. *uchars.settings* - fichier de configuration des caractères non éditables

Le fichier se compose de plusieurs entrées au format suivant :

HOST-CODEPAGE LOCAL-CODEPAGE HEX-CODEPOINTS ;

HEX-CODEPOINTS représente une liste de points de code hexadécimaux à deux chiffres et séparés par des espaces vides qui identifient les caractères non éditables. Cette liste s'achève par un point-virgule (;).

Les règles de syntaxe suivantes s'appliquent :

- Les lignes de commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US.
- Les lignes de données ne peuvent comporter que des données. Les commentaires ne sont pas autorisés sur la même ligne.
- Un astérisque (\*) peut être utilisé pour HOST-CODEPAGE et/ou LOCAL-CODEPAGE. L'astérisque est utilisé comme caractère générique et désigne toutes les pages de codes.
- Les entrées spécifiques ont la priorité sur les entrées génériques (caractère générique).
- "host-cp \*" a la priorité si "host-cp \*" et "\* local-cp" sont tout deux spécifiés et ne sont pas remplacés par "host-cp local-cp".
- Si la même paire de pages de codes est spécifiée plusieurs fois, la dernière entrée sera utilisée.

---

## (Facultatif) Utilisation de REXEC (ou SSH)

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

---

REXEC (exécution à distance) est un service TCP/IP qui permet aux clients d'exécuter une commande sur l'hôte. SSH (interpréteur de commandes sécurisé) est un service similaire mais toutes les communications sont chiffrées à l'aide de la couche SSL (Secure Socket Layer). Developer for System z utilise l'un de ces services pour effectuer des actions à distance (basées sur l'hôte) dans des sous-projets z/OS UNIX.

Developer for System z peut également être configuré pour utiliser REXEC (ou SSH) afin de démarrer un serveur RSE sur l'hôte. Notez toutefois, que chaque connexion lancée à l'aide de ce procédé entraînera le démarrage d'un autre serveur RSE, chaque serveur utilisant un nombre considérable de ressources système. Aussi, cette méthode de connexion alternative est uniquement viable pour un petit nombre de connexions.

En outre, l'autre méthode de connexion REXEC (ou SSH) ignorant le démon RSE, elle n'a pas accès à tous les services hôte décrits dans le présent document (l'audit et le traitement d'un seul serveur, par exemple). Contactez le service de support technique IBM pour savoir si un service hôte spécifique est pris en charge par la méthode de connexion alternative REXEC.

**Remarque :** Developer for System z utilise la version z/OS UNIX de REXEC et non la version TSO.

## Actions à distance (basées sur l'hôte) dans les sous-projets z/OS UNIX

Les actions à distance (basées sur l'hôte) pour les sous-projets z/OS UNIX nécessitent que REXEC ou SSH soit actif sur l'hôte. Si REXEC/SSH n'est pas configuré pour utiliser le port par défaut, le client de Developer for System z doit définir le port correct à utiliser avec les sous-projets z/OS UNIX. Pour cela, sélectionnez la page de préférences **Fenêtre > Préférences... > Solutions z/OS > Sous-projets USS > Options d'action distante**. Voir «Configuration de REXEC (ou SSH)» pour connaître le port utilisé.

## Méthode de connexion RSE alternative

Les clients Developer for System z doivent connaître les deux valeurs suivantes pour démarrer une connexion RSE via REXEC (ou SSH) :

- Le répertoire dans lequel se trouve le script de démarrage `server.zseries`.

Le script `server.zseries` se trouve dans `/etc/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

**Remarque :** Pour ne pas encourager l'usage de REXEC (ou de SSH) comme méthode de connexion alternative, `server.zseries` n'est plus copié automatiquement dans `/etc/rdz`. Pour utiliser cette fonction, vous devez la copier manuellement à partir de `/usr/lpp/rdz/bin`, comme indiqué dans l'exemple de commande suivant :

```
cp /usr/lpp/rdz/bin/server.zseries /etc/rdz
```

- Le port qui est utilisé.

Voir «Configuration de REXEC (ou SSH)» pour connaître le port utilisé.

## Configuration de REXEC (ou SSH)

Le document *Communications Server IP Configuration Guide* (SC31-8775) décrit les étapes à suivre pour configurer REXEC (ou SSH). Voir Annexe C, «Configuration d'INETD», à la page 323 pour connaître les remarques de configuration spécifiques de Developer for System z (il n'y a pas d'étape de configuration spécifique pour Developer for System z).

Le port généralement utilisé par REXEC est le 512. Pour vérifier, vous pouvez contrôler `/etc/inetd.conf` et `/etc/services` afin de trouver le numéro de port utilisé.

- Recherchez le nom de service (1er mot, `exec` dans cet exemple) du serveur `rexecd` (7e mot) dans `/etc/inetd.conf`.  
`exec stream tcp nowait OMVSKERN /usr/sbin/orexecd rexecd -LV`
- Recherchez le port (2e mot, 512 dans cet exemple) rattaché à ce nom de service (1er mot) dans `/etc/services/`.  
`exec 512/tcp #REXEC Command Server`

Le même principe s'applique à SSH. Le port qu'il utilise généralement est le 22 et le nom du serveur est `sshd`.

**Remarque :** `/etc/inetd.conf` et `/etc/services` peuvent posséder des noms différents. Pour plus d'informations, voir l'Annexe C, «Configuration d'INETD», à la page 323.

---

## (Facultatif) Transaction APPC pour le service Commandes TSO

Vous aurez besoin de l'aide d'un administrateur APPC et d'un administrateur WLM pour effectuer cette tâche de personnalisation, qui requiert les tâches de personnalisation spéciales ou les ressources suivantes :

- Mise à jour LINKLIST
- Transaction APPC
- Mise à jour WLM

---

Le service Commandes TSO peut être mis en oeuvre comme un programme transactionnel APPC, FEKFRSRV. Cette transaction fait office de serveur hôte pour l'exécution des commandes TSO et ISPF soumises à partir du poste de travail. Il n'est pas nécessaire d'installer APPC sur le poste de travail, car le client communique avec FEKFRSRV via RSE. Chaque client peut disposer d'une connexion active à plusieurs hôtes simultanément.

**Remarque :**

- Par défaut, Developer for System z utilise la passerelle client TSO/ISPF d'ISPF pour accéder au service Commandes TSO.
- Si vous n'êtes pas familiarisé à l'utilisation d'APPC, voir l'Annexe D, «Configuration APPC», à la page 331 avant de poursuivre la section.
- RSE utilise l'API du socket TCP/IP REXX pour communiquer avec FEKFRSRV. Cela implique que la bibliothèque de chargement TCP/IP (TCPIP.SEZALOAD, par défaut) soit disponible par l'intermédiaire de la directive LINKLIST ou STEPLIB dans `rsed.envvars`.
- RSE doit être redémarré pour que les modifications décrites prennent effet.

## Préparation

- Les tâches suivantes sont prérequis et doivent être effectuées avant de configurer le serveur de Commandes TSO. Les publications indiquées décrivent ces tâches.

1. Installation, configuration, et démarrage de VTAM sur le système z/OS. Voir *Communications Server IP SNA Network Implementation Guide* (SC31-8777) pour de plus amples informations.
  2. Installation, configuration et démarrage de TCP/IP sur le système z/OS. Pour plus d'informations, voir l'Annexe B, «Configuration de TCP/IP», à la page 315.
  3. Configurez et démarrez APPC et le sous-système ASCH (planificateur de transactions APPC). Pour plus d'informations, voir l'Annexe D, «Configuration APPC», à la page 331.
- Le code REXX suivant permet de gérer APPC via les panneaux ISPF.

```
/* REXX -- Administration APPC utilisant les panneaux ISPF */
address ISPEXEC
"LIBDEF ISPLIB DATASET ID('ICQ.ICQMLIB') STACK"
"LIBDEF ISPLIB DATASET ID('ICQ.ICQPLIB') STACK"
"LIBDEF ISPLIB DATASET ID('ICQ.ICQSLIB') STACK"
"LIBDEF ISPLIB DATASET ID('ICQ.ICQTLIB') STACK"
address TSO "ALTLIB ACT APPLICATION(CLIST)",
            "DSN('ICQ.ICQCLIB') UNCOND QUIET"
"SELECT CMD(%ICQASRM0) NEWAPPL(ICQ) PASSLIB"
address TSO "ALTLIB DEACT APPLICATION(CLIST) QUIET"
"LIBDEF ISPLIB"
"LIBDEF ISPLIB"
"LIBDEF ISPLIB"
"LIBDEF ISPLIB"
"LIBDEF ISPLIB"
exit
```

Figure 32. Panneaux REXX pour APPC ISPF

**Remarque :** Sachez toutefois que vous pouvez désactiver la transaction APPC avec cet outil ; la transaction existera toujours mais n'acceptera aucune connexion.

- La définition de la transaction APPC nécessite des compétences dans plusieurs domaines du système d'exploitation MVS. Parcourez les différents points de la liste ci-dessous avec des administrateurs expérimentés avant de poursuivre.

Tableau 16. Liste de contrôle de transaction APPC

Compétences	Information obligatoire :	Valeur
	<ul style="list-style-type: none"> <li>• Valeur par défaut</li> <li>• Emplacement de la réponse</li> </ul>	
APPC	Nom de fichier de TPDATA <ul style="list-style-type: none"> <li>• Par défaut : SYS1.APPCTP</li> <li>• La valeur se trouve dans SYS1.PARMLIB(APPCPMxx)</li> </ul>	
APPC	Nom de transaction à utiliser (peut ne pas exister) <ul style="list-style-type: none"> <li>• Par défaut : FEKFRSRV</li> <li>• Les transactions existantes sont interrogées en sélectionnant "Administration de profil du programme transactionnel" dans le menu ISPF APPC</li> </ul>	
APPC	Classe de transaction APPC à utiliser <ul style="list-style-type: none"> <li>• Par défaut : A</li> <li>• Les classes APPC sont définies dans SYS1.PARMLIB(ASCHPMxx)</li> </ul>	
WLM/SRM	Domaine et groupe de performances TSO <ul style="list-style-type: none"> <li>• Aucune valeur IBM par défaut (dépend du site)</li> </ul>	

Tableau 16. Liste de contrôle de transaction APPC (suite)

Compétences	Information obligatoire :	Valeur
	<ul style="list-style-type: none"> <li>• Valeur par défaut</li> <li>• Emplacement de la réponse</li> </ul>	
RACF	<p>Chaque utilisateur de Developer for System z dispose d'un accès à un segment OMVS (obligatoire).</p> <ul style="list-style-type: none"> <li>• Aucune valeur IBM par défaut (dépend du site)</li> <li>• La commande RACF TSO <b>OMVS d'un ID utilisateur de l'unité logique</b> affiche un segment OMVS personnel existant</li> </ul>	
RACF	<p>Chaque utilisateur de Developer for System z doit disposer de droits d'accès READ à hlq.SFEKPROC(FEKFRSRV)</p> <ul style="list-style-type: none"> <li>• Aucune valeur IBM par défaut (dépend du site)</li> <li>• La commande RACF TSO <b>LD AUTHUSER DATASET('hlq.SFEKPROC.**')</b> affiche les personnes et groupes ainsi que leurs niveaux d'accès aux fichiers du profil de fichier.</li> </ul>	

Voir le document *MVS Planning Workload Management* (SA22-7602) pour de plus amples informations sur la gestion WLM/SRM. Pour plus d'informations sur les segments OMVS et les profils de protection des fichiers, voir le document *Security Server RACF Security Administrator's Guide* (SA22-7683).

## Implémentation

**Remarque :** Les exemples de membres FEKAPPC\* se trouvent dans FEK.#CUST.JCL, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

1. Définissez les informations de calendrier (classe) pour le planificateur de transactions APPC si vous n'utilisez pas de classe de transaction existante. Indiquez une définition dans SYS1.PARMLIB(ASCHPMxx) pour la classe que le programme transactionnel FEKFRSRV doit utiliser. Cette classe est utilisée dans l'exemple JCL FEK.#CUST.JCL(FEKAPPCC). Par conséquent, la classe FEKAPPCC doit correspondre à celle définie dans SYS1.PARMLIB(ASCHPMxx). Par exemple :

```
CLASSADD
  CLASSNAME(A)
  MAX(20)
  MIN(1)
  MSGLIMIT(200)
```

**Remarque :**

- Le service Commandes TSO nécessite également que les spécifications par défaut soient indiquées aux sections OPTIONS et TPDEFAULT de SYS1.PARMLIB(ASCHPMxx). Pour plus d'informations, voir l'Annexe D, «Configuration APPC», à la page 331.
  - La classe de transaction APPC utilisée doit comporter suffisamment de demandeurs APPC pour en autoriser un par utilisateur simultané de Developer for System z.
2. Définissez la transaction APPC qui agira en tant que serveur de commandes. Pour définir cette transaction, vous pouvez utiliser l'exemple JCL



FEK.#CUST.JCL(FEKAPPCC). Des instructions sur la méthode de personnalisation du langage JCL se trouvent dans le JCL.

**Remarque :**

- a. Si vous avez modifié le nom de programme transactionnel (par défaut FEKFRSRV) dans cette étape, le nouveau nom doit être attribué à `_FEKFSCMD_TP_NAME_` dans `rsed.envvars` (voir «rsed.envvars - Fichier de configuration RSE», à la page 32).
  - b. La transaction APPC utilise le code REXX exec FEKFRSRV situé dans FEK.SFEKPROC. Ne modifiez pas cet emplacement si vous souhaitez avoir la possibilité d'activer automatiquement une opération de maintenance SMP/E.
  - c. L'exemple JCL permet également d'afficher, FEK.#CUST.JCL(FEKAPPCL), ou de supprimer, FEK.#CUST.JCL(FEKAPPCX), la transaction.
3. Activez RSE pour l'utilisation d'APPC en supprimant la mise en commentaire de la directive `RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` dans `rsed.envvars` (voir «rsed.envvars - Fichier de configuration RSE», à la page 32).
  4. Contrôlez la priorité de distribution du programme transactionnel en associant FEKFRSRV à un domaine et un groupe de performances dans Workload Manager (WLM). FEKFRSRV émettant des commandes TSO, il doit être affecté à un groupe de performances TSO.
  5. Définissez un segment OMVS par défaut pour le système ou un segment individuel pour chaque utilisateur de Developer for System z.
  6. Donnez aux utilisateurs de Developer for System z l'accès en lecture (READ) à FEK.SFEKPROC, la bibliothèque des exécutables TSO de Developer for System z.

## Remarques relatives à l'utilisation d'APPC

- Lorsque vous utilisez APPC pour le service Commandes TSO, Developer for System z dépend de la validité du nom d'hôte du TCP/IP quand il est initialisé. Cela implique que les différents fichiers de configuration du TCP/IP et du programme de résolution Resolver soient configurés correctement. Pour plus d'informations sur la personnalisation de TCP/IP et du programme de résolution, voir Annexe B, «Configuration de TCP/IP», à la page 315 et la section *TCPIP.DATA configuration statements* dans le document *Communications Server IP Configuration Reference* (SC31-8776).

Vous pouvez tester la configuration TCP/IP en démarrant le démon RSE à l'aide du paramètre `IVP=IVP` ou du programme de vérification de l'installation `fekfivpt`, comme indiqué dans Chapitre 7, «Vérification de l'installation», à la page 111.

- Lorsque vous utilisez APPC pour le service Commandes TSO, Developer for System z requiert une connexion socket supplémentaire (port TCP/IP) pour les communications limitées au système hôte par fichier MVS ouvert. Tout port disponible sera utilisé. Ce mécanisme de sélection de port ne peut pas être modifié.
- Lorsque vous utilisez APPC pour le service Commandes TSO, la lecture et l'écriture d'un fichier MVS nécessitent l'emploi d'un socket de domaine de système de fichiers physique. Si le système de fichiers n'est pas correctement défini ou s'il n'a pas suffisamment de sockets, le gestionnaire de verrous (FFS) peut échouer dans les requêtes de lecture/écriture. Voir «Echec de l'ouverture des fichiers MVS», à la page 158.

- Lorsque vous utilisez APPC pour le service Commandes TSO afin d'éviter de configurer la passerelle client TSO/ISPF d'ISPF, sachez que d'autres services (SCLM Developer Toolkit, par exemple) reposent sur la passerelle client TSO/ISPF.
- Voir Annexe D, «Configuration APPC», à la page 331 pour consulter des remarques générales sur l'utilisation d'APPC.

---

## (Facultatif) Nettoyage du répertoire WORKAREA

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

---

La passerelle client TSO/ISPF d'ISPF et la fonction SCLM Developer Toolkit utilisent le répertoire WORKAREA pour stocker des fichiers de travail temporaires qui sont supprimés avant la fermeture de la session. Cependant, la sortie temporaire est parfois conservée, par exemple, en cas d'erreur de communication en cours de traitement. C'est la raison pour laquelle il est recommandé de nettoyer régulièrement le répertoire WORKAREA.

z/OS UNIX fournit un script de shell, `skulker`, qui supprime les fichiers en fonction du répertoire dans lequel ils se trouvent et de leur durée de vie. À l'aide du démon `cron` z/OS UNIX qui exécute les commandes à des dates et heures données, vous pouvez configurer un outil automatisé qui nettoie régulièrement le répertoire WORKAREA. Voir le document *UNIX System Services Command Reference* (SA22-7802) pour plus d'informations relatives au script `skulker` et au démon `cron`.

**Remarque :** Le répertoire WORKAREA se trouve dans `/var/rdz/`, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

---

## Chapitre 7. Vérification de l'installation

Une fois que vous avez terminé la personnalisation du produit, vous pouvez utiliser les programmes de vérification de l'installation décrits dans ce chapitre pour vérifier que l'installation des principaux composants du produit a abouti.

---

### Vérification des tâches démarrées

#### Moniteur de travaux JES (JMON)

Lancez la tâche démarrée JMON (ou le travail utilisateur). Les informations de démarrage de la définition de données STDOUT doivent se terminer par le message suivant :

```
JM200I Server initialization complete.
```

Si le travail s'arrête avec un code retour 66, cela signifie que l'APF n'autorise pas FEK.SFEKAUTH.

**Remarque :** Lancez le moniteur de travaux JES avant de continuer les autres tests IVP.

#### LOCKD, démon Lock

Lancez la tâche démarrée LOCKD (ou le travail utilisateur). Le démon lock émet le message de console suivant si le démarrage a abouti :

```
FEK501I Lock daemon started, port=4036, cleanup interval=1440,  
log level=1
```

#### Démon RSED, RSE

Lancez la tâche démarrée RSED (ou le travail utilisateur) avec le paramètre IVP=IVP. Avec ce paramètre, le serveur s'interrompt après avoir effectué certains tests de vérification de l'installation. Les résultats de ces tests sont disponibles dans DD STDOUT. Si des erreurs sont survenues, les données seront également disponibles dans DD STDERR. Les données STDOUT doivent ressembler à l'exemple suivant :

**Remarque :** Lancez le démon RSE sans le paramètre IVP avant de continuer les autres tests IVP. Le démon RSE émet le message de console suivant si le démarrage a abouti :

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

Test IVP du démon RSE

```
Wed Jul 2 17:11:52 2008 UTC  
uid=8(STCRSE) gid=1(STCGROUP)
```

```
RSE daemon port is 4035  
RSE configuration files located in /etc/rdz
```

```
-----  
current environment variables  
-----
```

```
@="/usr/lpp/rdz/bin/rsed.sh" @[1]="4035" @[2]="/etc/rdz"  
CGI_DTCONF="/var/rdz/scldmt"  
CGI_DTWORK="/var/rdz"  
CGI_TRANTABLE="FEK.#CUST.LSTRANS.FILE"  
CLASSPATH=".:usr/lpp/rdz/lib:usr/lpp/rdz/lib/dstore_core.jar:usr/lpp/
```

```

ERRNO="0"
HOME="/tmp"
IFS="
"
JAVA_HOME="/usr/lpp/java/J5.0"
JAVA_PROPAGATE="NO"
LANG="C"
LIBPATH=".:usr/lib:usr/lpp/java/J5.0/bin:usr/lpp/java/J5.0/bin/classi
LINENO="66"
LOGNAME="STCRSE"
MAILCHECK="600"
OLDPWD="/tmp"
OPTIND="1"
PATH=".:usr/lpp/java/J5.0/bin:usr/lpp/rdz/bin:usr/lpp/ispf/bin:bin:/
PPID="33554711"
PS1="\$ "
PS2="> "
PS3="#? "
PS4="+ "
PWD="/etc/rdz"
RANDOM="27298"
RSE_CFG="/etc/rdz"
RSE_HOME="/usr/lpp/rdz"
RSE_LIB="/usr/lpp/rdz/lib"
SECONDS="0"
SHELL="/bin/sh"
STEPLIB="NONE"
TZ="EST5EDT"
_BPX_SHAREAS="YES"
_BPX_SPAWN_SCRIPT="YES"
_CEE_DMPTARG="/tmp"
_CEE_RUNOPTS="ALL31(ON) HEAP(32M,32K,ANYWHERE,KEEP,,) TRAP(ON)"
_CMDSERV_BASE_HOME="/usr/lpp/ispf"
_CMDSERV_CONF_HOME="/etc/rdz"
_CMDSERV_WORK_HOME="/var/rdz"
_RSE_CMDSERV_OPTS="&SESSION=SPAWN"
_RSE_DAEMON_CLASS="com.ibm.etools.zos.server.RseDaemon"
_RSE_DAEMON_IVP_TEST="1"
_RSE_DAEMON_PORT="4035"
_RSE_JAVAOPTS=" -DISPF_OPTS='&SESSION=SPAWN' -DA_PLUGIN_PATH=/usr/lpp/rd
_RSE_POOL_SERVER_CLASS="com.ibm.etools.zos.server.ThreadPoolProcess"
_RSE_PWD="/tmp"
_RSE_SERVER_CLASS="org.eclipse.dstore.core.server.Server"
_RSE_SERVER_TIMEOUT="120000"
_SCLMDT_BASE_HOME="/usr/lpp/rdz"
_SCLMDT_CONF_HOME="/var/rdz/sclmdt"
_SCLMDT_TRANTABLE="FEK.#CUST.LSTRANS.FILE"
_SCLMDT_WORK_HOME="/var/rdz"
_SCLM_BASE="/var/rdz/WORKAREA"
_SCLM_BWBCALL="/usr/lpp/rdz/bin/BWBCALL"
_SCLM_DWGET="/var/rdz/WORKAREA"
_SCLM_DWTRANSFER="/var/rdz/WORKAREA"
_SCLM_J2EPUT="/var/rdz/WORKAREA"

-----
java startup test...
-----
java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition (build pmz31dev-2008031
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmmz3123-2008
J9VM - 20080314_17962_bHdSMr
JIT - 20080130_0718ifx2_r8
GC - 200802_08)
JCL - 20080314

-----
TCP/IP IVP test...

```

```

-----
Wed Jul  2 13:11:54 EDT 2008
uid=8(STCRSE) gid=1(STCGROUP)
using /etc/rdz/rsed.envvars

-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964

res_init Resolver values:
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset = /etc/resolv.conf
Translation Table = Default
UserId/JobName = STCRSE
Caller API = LE C Sockets
Caller Mode = EBCDIC
(L) DataSetPrefix = TCPIP
(L) HostName = CDFMVS08
(L) TcpIpJobName = TCPIP
(L) DomainOrigin = RALEIGH.IBM.COM
(L) NameServer = 9.42.206.2
                  9.42.206.3
(L) NsPortAddr = 53 (L) ResolverTimeout = 10
(L) ResolveVia = UDP (L) ResolverUdpRetries = 1
(*) Options NDots = 1
(*) SockNoTestStor
(*) AlwaysWto = NO (L) MessageCase = MIXED
(*) LookUp = DNS LOCAL
res_init Succeeded
res_init Started: 2008/07/02 13:11:54.755363
res_init Ended: 2008/07/02 13:11:54.755371
*****
MVS TCP/IP NETSTAT CS V1R9 TCPIP Name: TCPIP 13:11:54
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled

-----
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match

-----
PassTicket IVP test...
-----

Success, PassTicket IVP finished normally

-----
IVP du démon RSE terminé

```

---

## Vérification des services

L'installation de Developer for System z met à disposition plusieurs programmes de vérification de l'installation (IVP) pour les services de base et facultatifs. Les scripts IVP se trouvent dans le répertoire d'installation, /usr/lpp/rdz/bin/ par défaut.

Tableau 17. Programmes de vérification de l'installation pour les services

fekfivpa	«(Facultatif) Connexion au service Commandes TSO à l'aide d'APPC», à la page 120
fekfivpd	«Connexion du démon RSE», à la page 117
fekfivpi	«Connexion de la passerelle client TSO/ISPF d'ISPF», à la page 118
fekfivpj	«Connexion du moniteur de travaux JES», à la page 117
fekfivpl	«Connexion du démon lock», à la page 118
fekfivpr	«(Facultatif) Connexion à REXEC», à la page 122
fekfivps	«(Facultatif) Connexion à SCLMDT», à la page 120
fekfivpt	«Configuration TCP/IP», à la page 116
fekfivpz	«(Facultatif) Script de shell REXEC/SSH», à la page 123

Les tâches décrites ci-dessous nécessitent des actions de votre part dans le système z/OS UNIX. Vous pouvez les effectuer en lançant la commande TSO **OMVS**. Utilisez la commande **exit** pour retourner à TSO.

Une taille de région importante est nécessaire pour l'ID utilisateur qui exécute les IVP, car les fonctions telles que Java, qui demandent beaucoup de mémoire, vont être exécutées. Il est recommandé d'attribuer au moins 131072 octets (128 mégaotets) ou plus à la taille de région.

L'exemple d'erreur suivant est une indication précise d'une taille de région insuffisante (mais d'autres erreurs peuvent également se produire. Par exemple, Java peut ne pas démarrer).

```
CEE5213S The signal SIGPIPE was received.
%z/OS UNIX command%: command was killed by signal number 13
    %line-number% **    %REXX command%
    +++ RC(137) +++
```

**Remarque :** Les tâches démarrées Developer for System z doivent être actives avant le lancement du test IVP.

## Initialisation de programme de vérification de l'installation

Tous les exemples de commandes de la présente section sous-entendent que les variables de l'environnement sont configurées. De cette manière, les scripts IVP sont disponibles par l'intermédiaire de l'instruction PATH, et l'emplacement des fichiers de configuration personnalisés est connu. Utilisez les commandes **pwd** et **cd** pour vérifier votre répertoire de travail, et en changer pour le répertoire qui contient les fichiers de configuration personnalisés. Le script de shell **ivpinit** peut alors être utilisé pour configurer les variables d'environnement RSE, comme dans l'exemple ci-après (\$ est l'invite UNIX z/OS) :

```
$ pwd
/u/userid
$ cd /etc/rdz
$ ./ivpinit
RSE configuration files located in /etc/rdz --default
ont ajouté /usr/lpp/rdz/bin dans PATH
```

Le premier "." (point) dans `./ivpinit` est une commande z/OS UNIX qui permet d'exécuter le shell dans l'environnement en cours afin que les variables d'environnement définies dans le shell soient effectives même après la sortie du shell. Le second fait référence au répertoire de travail.

#### Remarque :

- Si `./ivpinit` n'est PAS exécuté avant les scripts `fekfivp*`, le chemin d'accès à ces scripts doit être indiqué lorsqu'ils sont appelés, comme dans l'exemple ci-après :  
`/usr/lpp/rdz/bin/fekfivpr 512 USERID`

La plupart des scripts `fekfivp*` demanderont également l'emplacement du `rsed.envvars` personnalisé si `./ivpinit` n'est pas exécuté en premier.

- Certains tests IVP utilisent l'API du socket TCP/IP REXX, ce qui nécessite que la bibliothèque de chargement TCP/IP, par défaut, `TCPIP.SEZALOAD`, soit dans `LINKLIST` ou `STEPLIB`. Les commandes suivantes peuvent être nécessaires pour exécuter ces tests IVP (\$ représente l'invite z/OS UNIX) :

```
$ EXPORT STEPLIB=$STEPLIB:TCPIP.SEZALOAD
```

Notez que l'ajout d'une bibliothèque non autorisée par APF dans un `STEPLIB` existant entraîne la suppression de l'autorisation APF pour les fichiers `STEPLIB` existants.

De même, si `CEE.SCEELKED` se trouve dans `LINKLIST` ou `STEPLIB`, `TCPIP.SEZALOAD` doit être placé avant `CEE.SCEELKED`. Tout manquement à cette règle entraîne une fin anormale du système 0C1 pour les appels de prise TCP/IP REXX.

Pour plus d'informations concernant le diagnostic des incidents de connexion RSE, voir Chapitre 9, «Identification des incidents liés à la configuration», à la page 139 ou la page relative aux notes techniques de Developer for System z (<http://www-306.ibm.com/software/awdtools/rdz/support/>).

## Disponibilité des ports

La disponibilité des ports du moniteur de travaux JES, du démon RSE et éventuellement de REXEC et/ou SSH peut être vérifiée en exécutant la commande **netstat**. Le résultat doit présenter les ports utilisés par ces services, comme dans les exemples ci-dessous (\$ est l'invite de z/OS UNIX) :

#### IPv4

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy      TCPIP Name: TCPIP      13:57:36
User Id  Conn      Local Socket          Foreign Socket         State
-----  ---
RSED     0000004B  0.0.0.0..4035        0.0.0.0..0            Listen
LOCKD    0000004C  0.0.0.0..4036        0.0.0.0..0            Listen
JMON     00000037  0.0.0.0..6715        0.0.0.0..0            Listen
```

#### IPv6

```
$ netstat
MVS TCP/IP NETSTAT CS VxRy      TCPIP Name: TCPIP      14:03:35
User Id  Conn      State
-----  ---
RSED     0000004B  Listen
Local Socket: 0.0.0.0..4035
Foreign Socket: 0.0.0.0..0
```



```

LOCKD    0000004C Listen
  Local Socket:  0.0.0.0..4036
  Foreign Socket: 0.0.0.0..0
JMON     00000037 Listen
  Local Socket:  0.0.0.0..6715
  Foreign Socket: 0.0.0.0..0

```

## Configuration TCP/IP

Lorsque vous utilisez APPC pour le service Commandes TSO, Developer for System z dépend de la validité du nom d'hôte du TCP/IP quand il est initialisé. Cela implique que les différents fichiers de configuration du TCP/IP et du programme de résolution Resolver soient configurés correctement. Voir l'Annexe B, «Configuration de TCP/IP», à la page 315 pour de plus amples informations sur la configuration de TCP/IP et du programme de résolution. Vérifiez les paramètres actuels en exécutant la commande suivante :

```
fekfivpt
```

**Remarque :** Cette IVP émet la commande TCPIP **netstat**, qui peut être protégée contre l'exécution par votre logiciel de sécurité.

La commande doit renvoyer un résultat comparable à celui de cet exemple (\$ correspond à l'invite z/OS UNIX) :

```
$ fekfivpt
```

```

Wed Jul  2 13:11:54 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

```

```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
```

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```

res_init Resolver values:
  Global Tcp/Ip Dataset = None
  Default Tcp/Ip Dataset = None
  Local Tcp/Ip Dataset  = /etc/resolv.conf
  Translation Table     = Default
  UserId/JobName         = USERID
  Caller API             = LE C Sockets
  Caller Mode            = EBCDIC
  (L) DataSetPrefix     = TCPIP
  (L) HostName           = CDFMVS08
  (L) TcpIpJobName       = TCPIP
  (L) DomainOrigin      = RALEIGH.IBM.COM
  (L) NameServer         = 9.42.206.2
                        9.42.206.3
  (L) NsPortAddr        = 53           (L) ResolverTimeout      = 10
  (L) ResolveVia        = UDP          (L) ResolverUdpRetries   = 1
  (*) Options NDots     = 1
  (*) SockNoTestStor    =
  (*) AlwaysWto         = NO           (L) MessageCase         = MIXED
  (*) LookUp            = DNS LOCAL

```

```
res_init Succeeded
```

```
res_init Started: 2008/07/02 13:11:54.755363
```

```
res_init Ended: 2008/07/02 13:11:54.755371
```

```
*****
```

```
MVS TCP/IP NETSTAT CS V1R9          TCPIP Name: TCPIP          13:11:54
```

```
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled
```

```

-----
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match

```

## Connexion du démon RSE

Testez la connexion du démon RSE en exécutant la commande ci-dessous. Remplacez 4035 par le port utilisé par le démon RSE et USERID par un ID utilisateur valide.

```
fekfivpd 4035 USERID
```

Après vous avoir demandé un mot de passe, la commande doit renvoyer un résultat semblable à celui de cet exemple (\$ est l'invite z/OS UNIX) :

```
$ fekfivpd 4035 USERID
```

```

Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

```

```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```

Password:
SSL is disabled
connected
8108
570655399
Success

```

**Remarque :** Lors du test d'une connexion SSL activée, vérifiez que vous avez spécifié le bon port si vous recevez le message d'erreur suivant :  
gsk\_secure\_socket\_init() failed: Socket closed by remote partner.

## Connexion du moniteur de travaux JES

Testez la connexion du moniteur de travaux JES en exécutant la commande ci-dessous. Remplacez 6715 par le numéro de port du moniteur de travaux JES.

```
fekfivpj 6715
```

La commande doit renvoyer l'accusé de réception du moniteur de travaux JES, comme dans l'exemple ci-dessous (\$ est l'invite de z/OS UNIX) :

```
$ fekfivpj 6715
```

```

Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars

```

```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```

hostName=CDFMVS08
hostAddr=9.42.112.75

```

```
Waiting for JES Job Monitor response...
ACKNOWLEDGE01v03
```

```
Success
```

## Connexion du démon lock

Vérifiez la connexion du démon lock en exécutant la commande ci-dessous.

```
fekfivpl
```

La commande doit renvoyer un résultat comparable à celui de l'exemple ci-dessous (\$ correspond à l'invite z/OS UNIX) :

```
$ fekfivpl
```

```
Mon Jun 29 08:00:38 EDT 2009
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
hostName=CDFMVS08
hostAddr=9.42.112.75
```

```
Registering user to Lock Daemon...
Waiting for Lock Daemon response...
```

```
Querying to Lock Daemon...
Waiting for Lock Daemon response...
USERID
```

```
Unregistering user from Lock Daemon...
Waiting for Lock Daemon response...
```

```
Querying to Lock Daemon...
Waiting for Lock Daemon response...
```

```
Success
```

## Connexion de la passerelle client TSO/ISPF d'ISPF

Vérifiez la connexion de la passerelle client TSO/ISPF d'ISPF en exécutant la commande suivante :

```
fekfivpi
```

La commande doit renvoyer le résultat des vérifications relatives à la passerelle client TSO/ISPF d'ISPF (variables, modules HFS, démarrage et arrêt de la session TSO/ISPF), comme celui de l'exemple ci-dessous (\$ représente l'invite z/OS UNIX) :

```
$ fekfivpi
```

```
Wed Jul 2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----
/etc/rdz/ISPF.conf content:
-----
```

```
isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispplib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
sysproc=ISP.SISPLIB,FEK.SFEKPROC
```

```
-----
Host install verification for RSE
Review IVP log messages from HOST below :
-----
```

RSE connection and base TSO/ISPF session initialization check only

\*\*\* CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

```
Server PATH          =
/usr/lpp/java/J5.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:
/bin:/usr/sbin:.
```

```
STEPLIB              = FEK.SFEKAUTH:FEK.SFEKLOAD
```

```
_CMDSERV_BASE_HOME   = /usr/lpp/ispf
_CMDSERV_CONF_HOME   = /etc/rdz
_CMDSERV_WORK_HOME    = /var/rdz
```

```
-----
*** CHECK : USS MODULES
Checking ISPF Directory : /usr/lpp/ispf
Checking modules in /usr/lpp/ispf/bin directory
Checking for ISPF configuration file ISPF.conf
RC=0
MSG: SUCCESSFUL
```

```
-----
*** CHECK : TSO/ISPF INITIALIZATION
( TSO/ISPF session will be initialized )
RC=0
MSG: SUCCESSFUL
```

```
-----
*** CHECK: Shutting down TSO/ISPF IVP session
RC=0
MSG: SUCCESSFUL
```

```
-----
Host installation verification completed successfully
-----
```

**Remarque :** Si l'une des vérifications ISPF échoue, de plus amples informations seront affichées.

fekfivpi présente plusieurs paramètres facultatifs ne dépendant pas de la position :

**-file** fekfivpi peut produire de grandes quantités en sortie (des centaines de lignes). Le paramètre -file envoie ce résultat à un fichier, userlog/.eclipse/RSE/\$LOGNAME/fekfivpi.log, où userlog est la valeur de la directive user.log dans rsed.envvars et \$LOGNAME est votre ID utilisateur (en majuscules). Si la directive user.log est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Le chemin du répertoire de base est défini dans votre segment de sécurité OMVS.

**-debug**

Le paramètre -debug crée une sortie détaillée des tests. N'utilisez pas cette option sans instruction du point service IBM.

## (Facultatif) Connexion au service Commandes TSO à l'aide d'APPC

Testez la connexion au serveur Commandes TSO (en utilisant APPC) en exécutant la commande ci-dessous. Remplacez USERID par un ID utilisateur valide :

```
fekfivpa USERID
```

Après vous avoir demandé un mot de passe, la commande doit renvoyer la conversation APPC, comme celle de l'exemple ci-dessous (\$ est l'invite de z/OS UNIX) :

```
$ fekfivpa USERID
Enter password:
```

```
Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
20070607 13:57:18.584060 /usr/lpp/rdz/bin/fekfscmd: version=Oct 2003
20070607 13:57:18.584326 Input parms: 1.2.3.4 * NOTRACE USERID *****
20070607 13:57:18.586800 APPC: Allocate succeeded
20070607 13:57:18.587022 Conversation id is 0DDBD3F800000000D
20070607 13:57:18.587380 APPC: Set Send Type succeeded
20070607 13:57:26.736674 APPC: Confirm succeeded
20070607 13:57:26.737027 Req to send recd value is 0
20070607 13:57:26.737546 APPC: SEND_DATA return_code = 0
20070607 13:57:26.737726 request_to_send_received = 0
20070607 13:57:26.737893 Send Data succeeded
20070607 13:57:26.738169 APPC: Set Prepare to Receive type succeeded
20070607 13:57:26.738580 APPC: Prepare to Receive succeeded
20070607 13:57:26.808899 APPC: Receive data
20070607 13:57:26.809122 RCV return_code = 0
20070607 13:57:26.809270 RCV data_received= 2
20070607 13:57:26.809415 RCV received_length= 29
20070607 13:57:26.809556 RCV status_received= 4
20070607 13:57:26.809712 RCV req_to_send= 0
20070607 13:57:26.809868 Receive succeeded
:IP: 0 9.42.112.75 1674 50246
20070607 13:57:26.810533 APPC: CONFIRMED succeeded
```

## (Facultatif) Connexion à SCLMDT

Vérifiez la connexion de SCLM Developer Toolkit en exécutant la commande suivante :

```
fekfivps
```

La commande doit renvoyer le résultat des vérifications relatives à SCLM Developer Toolkit (variables, modules HFS, environnement d'exécution REXX, démarrage et arrêt de session TSO/ISPF), comme celui de l'exemple suivant (\$ est l'invite de z/OS UNIX) :

```
$ fekfivps
```

```
Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----
/etc/rdz/ISPF.conf content:
```

```

-----
isplib=ISP.SISPMENU
isptlib=ISP.SISPTENU
ispllib=ISP.SISPPENU
ispslib=ISP.SISPSLIB
sysproc=ISP.SISPCLIB,FEK.SFEKPROC
-----

```

```

Host install verification for RSE
Review IVP log messages from HOST below :
-----

```

\*\*\* CHECK : ENVIRONMENT VARIABLES - key variables displayed below :

```

Server PATH          = /usr/lpp/java/J5.0/bin:/usr/lpp/rdz/lib:/usr/lpp/ispf/bin:
/bin:/usr/sbin:.

```

```

STEPLIB              = FEK.SFEKAUTH:FEK.SFEKLOAD

```

```

_CMDSERV_BASE_HOME   = /usr/lpp/ispf
_CMDSERV_CONF_HOME   = /etc/rdz
_CMDSERV_WORK_HOME    = /var/rdz
_SCLMDT_CONF_HOME     = /var/rdz/sclmdt
_SCLMDT_WORK_HOME     = /var/rdz
_SCLMDT_TRANTABLE     = FEK.#CUST.LSTRANS.FILE

```

\*\*\* CHECK : JAVA PATH SETUP VERIFICATION

```

RC=0
MSG: SUCCESSFUL

```

\*\*\* CHECK : USS MODULES

```

Checking ISPF Directory : /usr/lpp/ispf
Checking modules in /usr/lpp/ispf/bin directory
Checking for ISPF configuration file ISPF.conf
Checking install bin Directory : /usr/lpp/rdz/bin
RC=0
MSG: SUCCESSFUL

```

\*\*\* CHECK : REXX RUNTIME ENVIRONMENT

```

RC=0
MSG: SUCCESSFUL

```

\*\*\* CHECK : TSO/ISPF INITIALIZATION

```

( TSO/ISPF session will be initialized )
RC=0
MSG: SUCCESSFUL

```

\*\*\* CHECK: Shutting down TSO/ISPF IVP session

```

RC=0
MSG: SUCCESSFUL

```

```

-----
Host installation verification completed successfully
-----

```

**Remarque :** Si l'une des vérifications SCLMDT échoue, de plus amples informations seront affichées.

fekeivps présente plusieurs paramètres facultatifs ne dépendant pas de la position :

**-file** fekfivps peut produire de grandes quantités en sortie (des centaines de lignes). Le paramètre -file envoie ce résultat à un fichier, userlog/.eclipse/RSE/\$LOGNAME/fekfivps.log, où userlog est la valeur de la directive user.log dans rsed.envvars et \$LOGNAME est votre ID utilisateur (en majuscules). Si la directive user.log est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Le chemin du répertoire de base est défini dans votre segment de sécurité OMVS.

**-debug**

Le paramètre -debug crée une sortie détaillée des tests. N'utilisez pas cette option sans instruction du point service IBM.

## (Facultatif) Connexion à REXEC

Testez la connexion REXEC en exécutant la commande ci-dessous. Remplacez 512 par le port utilisé par REXEC et USERID par un ID utilisateur valide.

```
fekfivpr 512 USERID
```

Après vous avoir invité à saisir un mot de passe, la commande doit renvoyer la trace REXEC, un avertissement de dépassement du délai d'attente, la version de Java et le message du serveur RSE, comme dans l'exemple ci-après (\$ est l'invite de z/OS UNIX) :

```
$ fekfivpr 512 USERID
Enter password:
```

```
Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
$ EYZRC01I Calling function rexec_af with the following:
EYZRC02I Host: CDFMVS08, user USERID, cmd cd /etc/rdz;export RSE_USER_ID=USERI
D;./server.zseries -ivp, port 512
EYZRC19I Data socket = 4, Control socket = 6.
```

RSE server IVP test

```
CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)
```

RSE configuration files located in /etc/rdz --default

RSE userid is USERID --default

```
-----
Address Space size limits
```

```
-----
current address space size limit is 2147483647 (2048.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)
```

```
-----
service history
```

```
-----
Fri Jun 19 00:01:00 2009 -- COPY -- HHOP760 v7600 created 18 Jun 2009
```

```
-----
expect to see time out messages after a successful IVP test
```

```
-----
starting RSE server in background -- Fri Jun 19 15:59:05 EDT 2009
-----
```



```

java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition (build pmz31dev-20070201 (SR4))
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmmz3123-20070201 (JIT enabled))
J9VM - 20070131_11312_bHdSMr
JIT - 20070109_1805ifx1_r8
GC - 200701_09)
JCL - 20070126

```

```

DStore Server Starting...
Server Started Successfully
8108
Server running on: CDFMVS08

```

#### Remarque :

- Si vous n'obtenez aucune sortie de serveur Java et RSE, il est probable que la taille de la région INETD est trop petite (elle doit être de 2096128 ou plus pour un lancement à partir d'une session shell TSO/OMVS, ou d'une taille de région 0 pour BPXBATCH).
- Vous pouvez tester séparément le script de shell utilisé par REXEC, comme le décrit le prochain test IVP, «(Facultatif) Script de shell REXEC/SSH».
- Le serveur est démarré sans qu'un client tente de se connecter, par conséquent il va arriver au bout de son délai d'attente (au bout de 5 secondes). Il en résulte un message d'erreur de connexion qui ressemble à l'exemple suivant :

```

Connection error
Server: error initializing socket: java.net.SocketTimeoutException:
Accept timed out

```

### (Facultatif) Script de shell REXEC/SSH

Ce test IVP peut être évité si le test précédent décrit dans «(Facultatif) Connexion à REXEC», à la page 122, a été réussi.

Testez le script utilisé par la connexion REXEC et SSH en exécutant la commande ci-dessous :

```
fekfivpz
```

La commande doit renvoyer un avertissement de dépassement du délai d'attente, la version de Java et le message du serveur RSE, comme dans l'exemple ci-dessous (\$ est l'invite de z/OS UNIX) :

```
$ fekfivpz
```

```

Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)

```

```
using /etc/rdz/rsed.envvars
```

```

current address space size limit is 1914675200 (1826.0 MB)
maximum address space size limit is 2147483647 (2048.0 MB)

```

```
-----
```

```
RSE server IVP test
```

```

CDFMVS08 -- Wed Jul  2 15:00:27 EDT 2008
uid=1(USERID) gid=0(GROUP)

```

```

RSE configuration files located in /etc/rdz --default
RSE userid is USERID --default

```

```
-----  
Address Space size limits  
-----
```

```
current address space size limit is 2147483647 (2048.0 MB)  
maximum address space size limit is 2147483647 (2048.0 MB)  
-----
```

```
service history  
-----
```

```
Fri Jun 19 00:01:00 2009 -- COPY -- HHOP760 v7600 created 18 Jun 2009  
-----
```

```
expect to see time out messages after a successful IVP test  
-----
```

```
starting RSE server in background -- Fri Jun 19 15:59:05 EDT 2009  
-----
```

```
java version "1.5.0"  
Java(TM) 2 Runtime Environment, Standard Edition (build pmz31dev-20070201 (SR4))  
IBM J9 VM (build 2.3, J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmmz3123-20070201 (JIT enabled))  
J9VM - 20070131_11312_bHdSMr  
JIT - 20070109_1805ifx1_r8  
GC - 200701_09)  
JCL - 20070126
```

```
DStore Server Starting...  
Server Started Successfully  
8108  
Server running on: CDFMVS08
```

#### Remarque :

- Si vous n'obtenez aucun résultat, la taille de votre région (TSO) est probablement trop petite (doit être égale à 2096128).
- Le serveur est démarré sans qu'un client tente de se connecter, par conséquent il va arriver au bout de son délai d'attente (au bout de 5 secondes). Il en résulte un message d'erreur de connexion qui ressemble à l'exemple suivant :

```
Connection error  
Server: error initializing socket: java.net.SocketTimeoutException:  
Accept timed out
```

---

## **Partie 2. Informations relatives à Developer for System z**



---

## Chapitre 8. Commandes de l'opérateur

Le présent chapitre répertorie les commandes de l'opérateur (ou de la console) disponibles pour System z. Si vous avez besoin de vous familiariser avec les diagrammes de syntaxe utilisés pour expliquer le format de commande, reportez-vous à la section «Comment lire un diagramme de syntaxe», à la page 136.

---

### Start (S)

Utilisez la commande **START** pour lancer dynamiquement une tâche démarrée (STC). La version abrégée de la commande est la lettre **S**.

### Moniteur de travaux JES

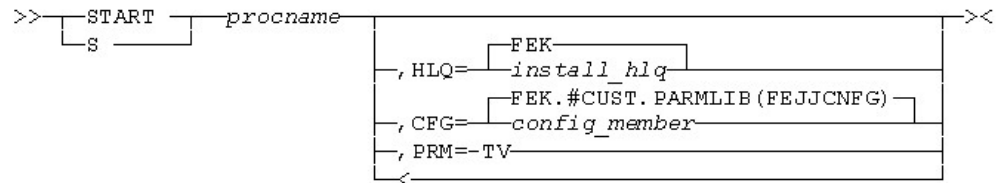


Figure 33. Commande de l'opérateur **START JMON**

#### **nomproc**

Nom du membre dans une bibliothèque de procédure utilisé pour démarrer le serveur. Le nom par défaut utilisé pendant la configuration de l'hôte Est JMON.

#### **HLQ=qualificatif\_haut\_niveau**

Qualificatif de haut niveau utilisé pour installer Developer for System z. La valeur par défaut est FEK.

#### **CFG=membre\_config**

Nom absolu de fichier et de membre du fichier de configuration du moniteur de travaux JES. La valeur par défaut est FEK.#CUST.PARMLIB (FEJJCNFG).

#### **PRM=-TV**

Active le mode prolixe (trace). La fonction de trace réduit les performances et ne doit être effectuée que sur indication du centre de support IBM.

## Démon RSE

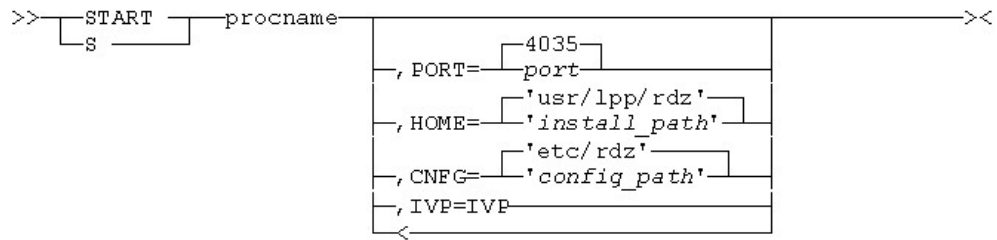


Figure 34. Commande de l'opérateur `START RSED`

### **nomproc**

Nom du membre dans une bibliothèque de procédure utilisé pour démarrer le serveur. Le nom par défaut utilisé pendant la configuration de l'hôte Est RSED.

### **PORT=port**

Port utilisé par le démon RSE pour la connexion des clients. La valeur par défaut est 4035.

### **HOME='chemin\_installation'**

Préfixe et chemin d'installation `/usr/lpp/rdz` obligatoires utilisés pour installer Developer for System z. Le chemin par défaut est `'/usr/lpp/rdz'`. Notez que le chemin z/OS UNIX doit respecter les majuscules/minuscules et être placé entre apostrophes (') pour préserver les minuscules.

### **CNFG='chemin\_config'**

Emplacement absolu des fichiers de configuration enregistrés dans z/OS UNIX. L'emplacement par défaut est `'/etc/rdz'`. Notez que le chemin z/OS UNIX doit respecter les majuscules/minuscules et être placé entre apostrophes (') pour préserver les minuscules.

### **IVP=IVP**

Ne démarrez pas le serveur mais exécutez le programme de vérification d'installation (IVP) du démon RSE.

## Démon lock

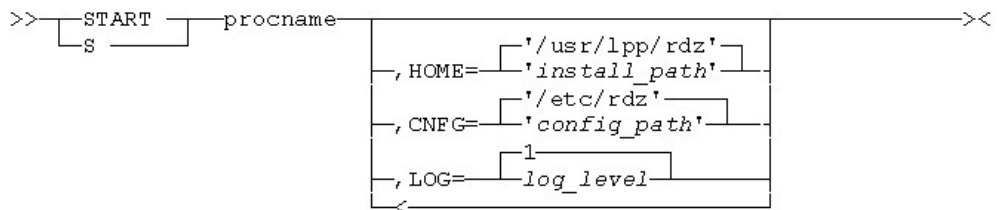


Figure 35. Commande de l'opérateur `START LOCKD`

### **nomproc**

Nom du membre dans une bibliothèque de procédure utilisé pour démarrer le serveur. Le nom par défaut utilisé pendant la configuration de l'hôte est LOCKD.

### **HOME='chemin\_installation'**

Préfixe et chemin d'installation `/usr/lpp/rdz` obligatoires utilisés pour

installer Developer for System z. Le chemin par défaut est '/usr/lpp/rdz'. Notez que le chemin z/OS UNIX doit respecter les majuscules/minuscules et être placé entre apostrophes (') pour préserver les minuscules.

**CNFG='chemin\_config'**

Emplacement absolu des fichiers de configuration enregistrés dans z/OS UNIX. L'emplacement par défaut est '/etc/rdz'. Notez que le chemin z/OS UNIX doit respecter les majuscules/minuscules et être placé entre apostrophes (') pour préserver les minuscules.

**LOG=niveau\_consignation**

Niveau de consignation de la sortie dans DD STDOUT.

- 0 : Uniquement les messages d'erreur.
- 1 : Les messages d'erreur et d'avertissement (valeur par défaut).
- 2 : Les messages d'erreur, d'avertissement et d'information.

---

## Modify (F)

La commande **MODIFY** permet d'interroger et de modifier de manière dynamique les caractéristiques d'une tâche active. La version abrégée de la commande est la lettre F.

## Moniteur de travaux JES

```
>> [MODIFY | F] [procname] [, APPL=-TV | , APPL=-TN] ><
```

Figure 36. Commande de l'opérateur **MODIFY JMON**

**nomproc**

Nom du membre dans une bibliothèque de procédure qui a été utilisé pour démarrer le serveur. Le nom par défaut utilisé pendant la configuration de l'hôte Est JMON.

- TV Active le mode prolix (trace). La fonction de trace réduit les performances et ne doit être effectuée que sur indication du centre de support IBM.
- TN Désactive le mode prolix (trace).

## Démon RSE

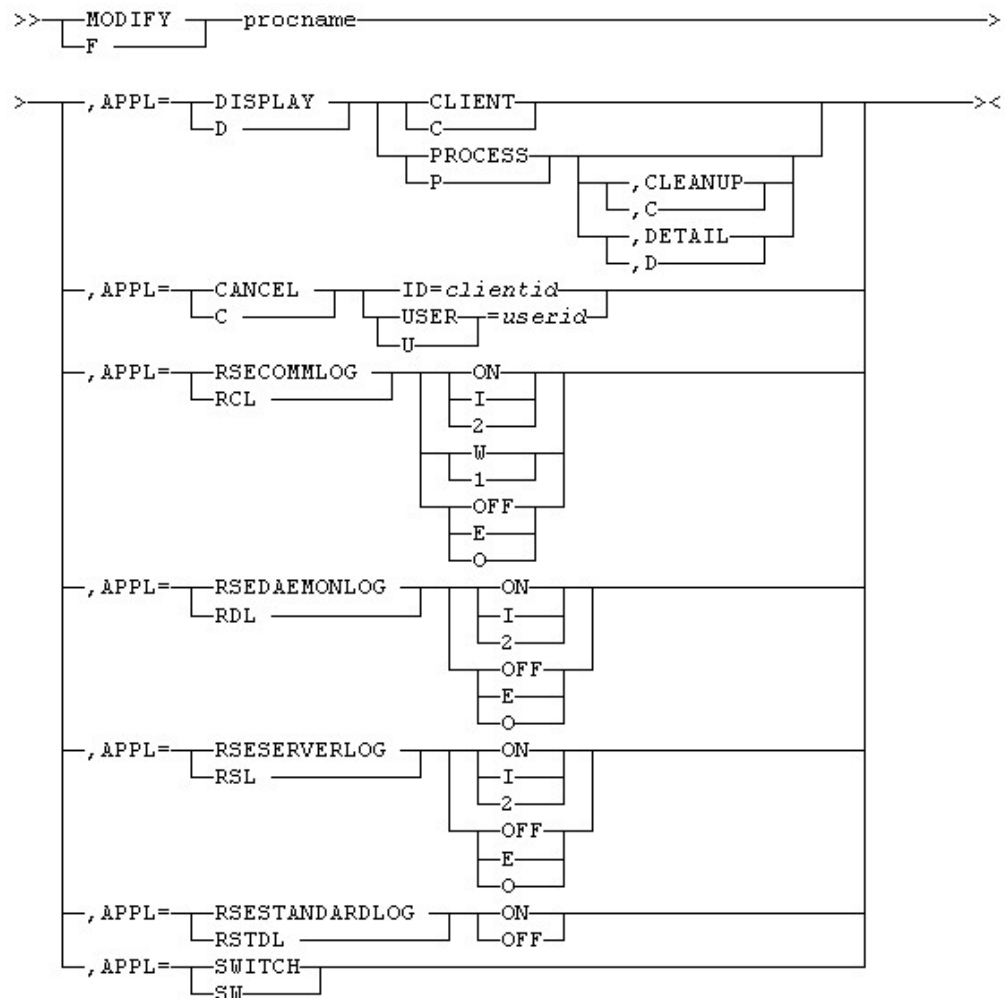


Figure 37. Commande de l'opérateur MODIFY RSED

### nomproc

Nom du membre dans une bibliothèque de procédure qui a été utilisé pour démarrer le serveur. Le nom par défaut utilisé pendant la configuration de l'hôte Est RSED.

### DISPLAY CLIENT

Affiche les clients actifs.

<IDclient> : <IDutilisateur> : <connecté depuis>

### DISPLAY PROCESS[,CLEANUP,DETAIL]

Affiche les processus du pool d'unités d'exécution RSE. Il peut exister plusieurs processus, qui sont utilisés pour l'équilibrage des charges des utilisateurs connectés.

ProcessId(<processid>) Memory Usage(<utilisation de segment de mémoire java>%)  
Clients(<nombre de clients>) Order(<ordre de démarrage>) <état d'erreur>

### Remarque :

- <processid> peut être utilisé dans les commandes z/OS UNIX spécifiques au processus.



- Chaque processus possède son propre segment de mémoire Java dont la taille peut être définie dans `rsed.envvars`.
- L'<ordre de démarrage> est un numéro de séquence qui indique l'ordre dans lequel les pools d'unités d'exécution ont été démarrés. Le numéro correspond au numéro utilisé dans le nom de fichier des fichiers `stderr.*.log` et `stdout.*.log`.

Dans les situations normales, <état d'erreur> est vide. Le tableau 18 documente les valeurs non-blanches possibles pour <état d'erreur>.

Tableau 18. Etat d'erreur du pool d'unités d'exécution

Etat	Description
*severe error*	Le processus du pool d'unités d'exécution a fait l'objet d'une erreur irrémédiable et a interrompu les opérations. Les autres zones d'état affichent les dernières valeurs connues. Utilisez l'option <b>CLEANUP</b> de la commande de modification <b>DISPLAY PROCESS</b> pour supprimer cette entrée de la table.
*killed process*	Le processus du pool d'unités d'exécution a été arrêté par Java, z/OS UNIX ou une commande de l'opérateur. Les autres zones d'état affichent les dernières valeurs connues. Utilisez l'option <b>CLEANUP</b> de la commande de modification <b>DISPLAY PROCESS</b> pour supprimer cette entrée de la table.
*timeout*	Le processus du pool d'unités d'exécution n'a pas répondu dans le délai imparti au démon RSE lors d'une demande de connexion d'un client. Les autres zones d'état affichent les valeurs en cours. Le pool d'unités d'exécution est exclu des futures demandes de connexion client. L'état *timeout* est réinitialisé lorsqu'un client pris en charge par ce pool d'unités d'exécution se déconnecte.

Des informations supplémentaires sont fournies lorsque vous utilisez l'option **DETAIL** de la commande de modification **DISPLAY PROCESS** :

```
ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
PROCESS LIMITS:  CURRENT  HIGHWATER  LIMIT
JAVA HEAP USAGE(%)  10      56      100
CLIENTS              0       25      60
MAXFILEPROC          83     103    64000
MAXPROCUSER          97      99     200
MAXTHREADS           9       14    1500
MAXTHREADTASKS       9       14    1500
```

Le champ **ASId** est l'ID d'espace adresse, en notation hexadécimale. La table des limites du processus montre l'usage actuel des ressources, la cote d'alerte haute pour l'usage des ressources et la liste des ressources. Notez qu'en raison d'autres facteurs de limitation, la limite définie ne pourrait jamais être atteinte.

**CANCEL ID=IDclient**

Annule une connexion client en fonction de l'ID client, indiqué dans la commande modify **DISPLAY CLIENT**.

**CANCEL USER=IDutilisateur**

Annule une connexion client en fonction de l'ID utilisateur du client, indiqué dans la commande Modify **DISPLAY CLIENT**.

**RSECOMMLOG {ON,OFF,I,W,E,2,1,0}**

Contrôle le niveau de consignation des données de trace pour le serveur RSE (rsecomm.log) et les services de fichier MVS (lock.log et ffs\*.log). La valeur de démarrage par défaut est définie dans rsecomm.properties. Trois niveaux sont disponibles :

E ou 0 ou OFF	Messages d'erreur uniquement.
W ou 1	Messages d'erreur et d'avertissement. Il s'agit du paramètre par défaut de rsecomm.properties.
I ou 2 ou ON	Messages d'erreur, d'avertissement et d'information.

La fonction de trace détaillée réduit les performances et ne doit être effectuée que sur indication du centre de support IBM.

**RSEDAEMONLOG {ON,OFF,I,E,2,0}**

Contrôle le niveau de consignation des données de trace pour le démon RSE (rsedaemon.log). La valeur de démarrage par défaut est définie dans rsecomm.properties. Deux niveaux sont disponibles :

E ou 0 ou OFF	Messages d'erreur uniquement.
I ou 2 ou ON	Messages d'erreur, d'avertissement et d'information.

La fonction de trace détaillée réduit les performances et ne doit être effectuée que sur indication du centre de support IBM.

**RSESERVERLOG {ON,OFF,I,E,2,0}**

Contrôle le niveau de consignation des données de trace pour les pools d'unités d'exécution RSE (rseserver.log). La valeur de démarrage par défaut est définie dans rsecomm.properties. Deux niveaux sont disponibles :

E ou 0 ou OFF	Messages d'erreur uniquement.
I ou 2 ou ON	Messages d'erreur, d'avertissement et d'information.

La fonction de trace détaillée réduit les performances et ne doit être effectuée que sur indication du centre de support IBM.

**RSESTANDARDLOG {ON,OFF}**

Désactive (OFF) ou active (ON) la mise à jour des fichiers journaux en gérant les flux stdout et stderr des pools d'unités d'exécution (stdout\*.log et stderr\*.log). La valeur de démarrage par défaut est définie par la directive enable.standard.log de rsed.envvars.

La fonction de trace détaillée réduit les performances et ne doit être effectuée que sur indication du centre de support IBM.

**SWITCH**

Bascule vers un nouveau journal d'audit.

**Remarque :**

- Pour plus d'informations sur les fichiers journaux mentionnés ci-dessus, voir «Fichiers journaux», à la page 140 dans le Chapitre 9, «Identification des incidents liés à la configuration», à la page 139.
- Pour plus d'informations sur l'audit, voir «Consignation dans le journal d'audit», à la page 167 dans le Chapitre 10, «Remarques relatives à la sécurité», à la page 161.

## Démon lock

```
>> MODIFY procname ----->
      F
> , APPL= QUERY dataset -----><
      Q      dataset (member)
```

Figure 38. Commande de l'opérateur MODIFY LOCKD

**nomproc**

Nom du membre dans une bibliothèque de procédure qui a été utilisé pour démarrer le serveur. Le nom par défaut utilisé pendant la configuration de l'hôte est LOCKD.

**QUERY dataset[(member)]**

Demande l'état de verrouillage du fichier ou du membre indiqué. Le serveur répond en renvoyant l'un des messages suivants :

```
BPXM023I (stclock) dataset[(member)] NOT LOCKED
BPXM023I (stclock) dataset[(member)] LOCKED BY userid
```

**Remarque :**

- Le serveur signale également les verrous maintenus par d'autres produits, tels que ISPF.
- Les verrous maintenus par des clients Developer for System z qui ne peuvent pas s'enregistrer auprès du démon lock entraîne l'enregistrement de l'espace adresse du serveur de pools d'unités d'exécution (RSEDx) comme propriétaire du verrou.

Le message de console FEK513W est généré lorsque le serveur RSE n'est pas en mesure d'enregistrer le client auprès du démon lock. Les valeurs ASID et TCB mentionnées dans ce message peuvent être comparées au résultat de la commande de l'opérateur **D GRS,RES=(\*,dataset[(member))** pour identifier l'utilisateur qui maintient le verrou.

---

## Stop (P)

Utilisez la commande **STOP** pour arrêter une tâche active. La version abrégée de la commande est la lettre P.

```
>> STOP procname -----><
      P
```

Figure 39. Commande de l'opérateur STOP

**nomproc**

Nom du membre dans une bibliothèque de procédure qui a été utilisé pour démarrer le serveur. Les noms par défaut utilisés pendant la configuration de l'hôte sont respectivement JMON, RSED et LOCKD pour le moniteur de travaux JES, le démon lock et le démon RSE.

---

## Messages de la console

### Moniteur de travaux JES

Le moniteur de travaux JES ne comporte pas de messages de console spécifiques au produit. Le serveur compte sur z/OS et JES pour générer des messages de console pour les actions effectuées par les clients Developer for System z.

### Démon RSE, serveur de pool d'unités d'exécution RSE et démon lock

Le tableau 19 répertorie les messages de console propres au produit générés par le démon RSE, le serveur de pool d'unités d'exécution RSE et le démon lock.

*Tableau 19. Messages de console RSE*

ID du message	Texte du message
FEK001I	RseDaemon étant initialisé en mode bit {0}
FEK002I	Démon RSE démarré. (port={0})
FEK003I	Commande d'arrêt en cours de traitement
FEK004I	Démon RSE : Taille de pile maximale={0}Mo et Taille AS privée={1}Mo
FEK005I	Processus serveur démarré (processId={0})
FEK009I	RseDaemon attend le démarrage du processus serveur.
FEK010I	(emplacement rsed.envvars = {0})
FEK011I	(répertoire de journalisation = {0})
FEK100E	Le port du démon / la valeur de délai d'attente doit être des chiffres
FEK101E	JRE {0} ou supérieur requis
FEK102E	Arguments non valides reçus : {0}
FEK103E	Le disque est presque plein dans {0}
FEK104E	Le nombre maximal de traitements a été atteint
FEK105E	Erreur lors de l'envoi des données d'audit (rc={0})
FEK110E	Echec de socket(). motif=({0})
FEK111E	Echec de setsockopt(). motif=({0})
FEK112E	Echec de bind(). motif=({0})
FEK113E	Echec de listen(). motif=({0})
FEK114E	Echec de accept(). motif=({0})
FEK115E	Echec de write(). motif=({0})
FEK116E	Echec de pipe(). motif=({0})
FEK117E	Echec de socketpair(). motif=({0})
FEK118E	Echec de select(). motif=({0})
FEK119E	Echec de _console(). motif=({0})
FEK130E	Echec de gsk_environment_open(). motif=({0})

Tableau 19. Messages de console RSE (suite)

ID du message	Texte du message
FEK131E	Echec de gsk_attribute_set_enum(GSK_PROTOCOL_SSLV2). motif=({0})
FEK132E	Echec de gsk_attribute_set_enum(GSK_PROTOCOL_SSLV3). motif=({0})
FEK133E	Echec de gsk_attribute_set_enum(GSK_PROTOCOL_TLSV1). motif=({0})
FEK134E	Echec de gsk_attribute_set_buffer(GSK_KEYRING_FILE). motif=({0})
FEK135E	Echec de gsk_attribute_set_buffer(GSK_KEYRING_PW). motif=({0})
FEK136E	Echec de gsk_environment_init(). motif=({0})
FEK137E	Echec de gsk_secure_socket_open(). motif=({0})
FEK138E	Echec de gsk_attribute_set_numeric_value(GSK_FD). motif=({0})
FEK139E	Echec de gsk_attribute_set_buffer(GSK_KEYRING_LABEL). motif=({0})
FEK140E	Echec de gsk_attribute_set_enum(GSK_SESSION_TYPE). motif=({0})
FEK141E	Echec de gsk_attribute_set_callback(GSK_IO_CALLBACK). motif=({0})
FEK142E	Echec de gsk_secure_socket_init(). motif=({0})
FEK143E	Echec de gsk_attribute_set_enum(GSK_CLIENT_AUTH_TYPE). motif=({0})
FEK144E	Echec de gsk_get_cert_info. motif=({0})
FEK145E	Echec de gsk_secure_socket_read(). motif=({0})
FEK146E	Echec de gsk_secure_socket_write(). motif=({0})
FEK150E	Fin anormale du démon RSE ; {0}
FEK201I	La commande {0} a été traitée
FEK202E	Commande non valide saisie
FEK203E	Commande d'affichage non valide : Afficher le processus   le client
FEK204E	Commande d'annulation non valide : ID d'annulation=   Utilisateur=
FEK205E	La commande n'a pas été traitée en raison de SWITCH consécutifs
FEK206E	La fonction de journal d'audit n'est pas active
FEK207I	Aucun client à afficher
FEK208I	{0} annulé
FEK209I	Aucun processus à afficher
FEK210I	{0} annulé(s) en raison d'une connexion en double
FEK501I	Démon lock démarré, port={0}, intervalle de nettoyage={1}, niveau de consignation={2}
FEK502I	Arrêt en cours du démon lock
FEK510E	Démon lock, port manquant
FEK511E	Démon lock, port erroné, port={0}
FEK512E	Démon lock, erreur de socket, port={0}
FEK513W	Démon lock, échec de l'enregistrement, ASID={0}, TCB={1}, UTILISATEUR={2}
FEK514W	Démon lock, niveau de consignation erroné, niveau de consignation={0}
BPXM023I	(stclock) dataset[(member)] NOT LOCKED

Tableau 19. Messages de console RSE (suite)

ID du message	Texte du message
BPXM023I	(stclock) dataset[(member)] LOCKED BY userid
BPXM023I	(stclock) command, WRONG COMMAND
BPXM023I	(stclock) command, MISSING ARGUMENT
BPXM023I	(stclock) argument, WRONG ARGUMENT

## Comment lire un diagramme de syntaxe

Le diagramme de syntaxe montre comment entrer une commande pour qu'elle soit correctement interprétée par le système d'exploitation. La lecture du diagramme de syntaxe s'effectue de la gauche vers la droite et de haut en bas, en suivant la ligne horizontale (chemin principal).

## Symboles

Les symboles utilisés dans les diagrammes de syntaxe sont les suivants :

Symbole	Description
>>	Marque le début du diagramme de syntaxe.
>	Indique que le diagramme de syntaxe a une suite.
	Marque le début et la fin d'un fragment ou d'une partie du diagramme de syntaxe.
><	Marque la fin du diagramme de syntaxe.

## Opérandes

Les types d'opérandes utilisés dans les diagrammes de syntaxe sont les suivants :

- Les opérandes requis sont affichés dans la ligne du chemin principal :  
 >>—OPERANDE\_REQUIS—><
- Les opérandes facultatifs sont affichés sous la ligne du chemin principal:

>>└──┬──┘><  
       └──OPERANDE\_FACULTATIF──┘

- Les opérandes par défaut sont affichés au-dessus de la ligne du chemin principal :

>>┌──┬──┐><  
       └──OPERANDE\_PAR\_DEFAUT──┘

Les opérandes sont classés comme mots clés ou variables :

- Les mots clés sont des constantes qui doivent être fournies. Si le mot clé apparaît dans le diagramme de syntaxe avec des majuscules et des minuscules, la portion en majuscules est l'abréviation du mot clé (par exemple, MOTclé). Les mots clés ne sont pas sensibles à la casse. Vous pouvez les coder en majuscules ou en minuscules.
- Les variables, en italiques, apparaissent en minuscules et représentent les noms ou les valeurs que vous fournissez. Par exemple, un nom de fichier est une variable. Il peut y avoir une différenciation maj/min dans les variables.

## Exemple de syntaxe

Dans l'exemple suivant, la commande USER est un mot clé. Le paramètre de variable requis est id\_utilisateur et le paramètre de variable facultatif est motdepasse. Remplacez les paramètres de variable par vos propres valeurs :

```
>>USER—id_utilisateur—><
      |
      | motdepasse |
      |
```

## Caractères non alphanumériques et espaces

Si un diagramme contient un caractère non alphanumérique (parenthèse, point, virgule, signe égal et espace), vous devez coder ce caractère dans la syntaxe. Dans notre exemple, vous devez coder `OPERAND=(001 0.001)` :

>>—OPERANDE—=(—001— —0.001—)————><

## Sélection de plusieurs opérandes

Une flèche orientée vers la gauche dans un groupe d'opérandes indique que plusieurs d'entre eux peuvent être sélectionnés et qu'un seul peut être répété: :

```

>>-----<<
|-----|
|OPERANDE_REPRODUCTIBLE_1|
|OPERANDE_REPRODUCTIBLE_2|
|-----|

```

## Diagramme sur plusieurs lignes

Si un diagramme dépasse une ligne, la première ligne se termine par une pointe de flèche et la deuxième ligne commence par une pointe de flèche :

```
>>| Première ligne d'un diagramme de syntaxe sur plusieurs lignes |->
>| Suite des sous-commandes, des paramètres ou des deux |-----><
```

## Fragments de syntaxe

Certains diagrammes peuvent contenir des fragments de syntaxe, qui servent à fractionner les diagrammes lorsqu'ils sont trop longs, trop complexes ou comportent trop de répétitions. Les noms de fragment de syntaxe peuvent avoir une casse mixte et ils apparaissent dans les diagrammes et dans l'en-tête du fragment. Le fragment est placé sous le diagramme principal :

>>—| Fragment de syntaxe

Fragment de syntaxe :

```
**|—1er OPERANDE—,—2ème OPERANDE—,—3ème OPERANDE—|
```





---

## Chapitre 9. Identification des incidents liés à la configuration

Le présent chapitre vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration de Developer for System z. Il se compose des sections suivantes :

- «Journal et analyse de configuration à l'aide de FEKLOGS»
- «Fichiers journaux», à la page 140
- «Fichiers de vidage», à la page 145
- «Fonction de trace», à la page 148
- «Données de droits z/OS UNIX», à la page 150
- «Ports TCP/IP réservés», à la page 153
- «Taille d'espace adresse», à la page 155
- «Transaction APPC et service Commandes TSO», à la page 156
- «Informations diverses», à la page 158

Pour plus d'informations, reportez-vous à la section Support du site Web de Developer for System z (<http://www-306.ibm.com/software/awdtools/rdz/support/>) pour consulter les notes techniques et disposer des dernières informations produites par notre équipe de support technique.

Dans la section Library du site Web (<http://www-306.ibm.com/software/awdtools/rdz/library/>), vous pouvez également consulter la dernière version de la documentation de Developer for System z, notamment les livres blancs.

Le centre de documentation de Developer for System z (<http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp>) fournit des informations sur le client Developer for System z et son interaction avec l'hôte (du point de vue du client).

La bibliothèque z/OS en ligne contient également des informations importantes, que vous pouvez consulter à l'adresse suivante : <http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/>.

Veillez nous tenir informé de l'absence d'une certaine fonction de Developer for System z. Vous pouvez ouvrir une demande d'amélioration (RFE) à l'adresse suivante :

<https://www.ibm.com/developerworks/support/rational/rfe/>

---

### Journal et analyse de configuration à l'aide de FEKLOGS

Developer for System z propose un modèle de travail (FEKLOGS), qui rassemble tous les fichiers journaux z/OS UNIX, ainsi que les informations d'installation et de configuration relatives à Developer for System z.

Le modèle de travail FEKLOGS se trouve dans FEK.#CUST.JCL, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

La personnalisation de FEKLOGS est présentée dans JCL. La personnalisation porte sur la mise à disposition de quelques variables clés.

**Remarque :** Les clients SDSF peuvent utiliser la ligne de commande **XDC** dans SDSF pour sauvegarder la sortie de travaux dans un fichier, qui peut être donné au point de service IBM.

---

## Fichiers journaux

Developer for System z crée des fichiers journaux utiles pour vous et pour le point de service IBM dans l'identification et la résolution des incidents. La liste ci-après présente les fichiers journaux que vous pouvez créer sur le système hôte z/OS. Situé en regard des journaux spécifiques au produit, vérifiez bien le SYSLOG de tous les messages associés.

Les fichiers journaux basés sur le système MVS peuvent être localisés par l'intermédiaire de l'instruction de définition de données appropriée. Les fichiers journaux basés sur z/OS UNIX sont situés dans les répertoires suivants :

- userlog/\$LOGNAME/

Les fichiers journaux propres à l'utilisateur sont placés dans userlog/\$LOGNAME/, où userlog est la valeur combinée des directives user.log et DSTORE\_LOG\_DIRECTORY dans rsed.envvars, et \$LOGNAME l'ID utilisateur de connexion (en majuscules). Si la directive user.log est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive DSTORE\_LOG\_DIRECTORY est mise en commentaire ou omise, .eclipse/RSE/ est ajouté à la valeur user.log.

- .dstoreMemLogging - Consignation sur l'utilisation de la mémoire DataStore
- .dstoreTrace - Consignation sur l'action DataStore
- fa.log - Journal d'intégration de Fault Analyzer
- fekfivpi.log - Journal du test du programme de vérification d'installation fekfivpi
- fekfivps.log - Journal du test du programme de vérification d'installation fekfivps
- ffs.log - Journal du serveur FFS (Foreign File System), qui exécute des fonctions MVS natives
- ffsgget.log - Journal du programme de lecture de fichier, qui lit un fichier séquentiel ou un membre d'un fichier partitionné
- ffsgput.log - Journal du programme d'écriture de fichier, qui écrit un fichier séquentiel ou un membre d'un fichier partitionné
- lock.log - Journal du gestionnaire de verrouillage, qui verrouille/déverrouille un fichier séquentiel ou un membre d'un fichier partitionné
- rmt\_class\_loader.cache.jar - Mémoire cache des classes chargées par le programme de chargement de classe distante RSE
- rsecomm.log - Journal du serveur RSE qui traite les commandes du client et la consignation des communications de tous les services qui utilisent RSE (peut contenir une trace de pile d'exception Java)
- stderr.log - Données réacheminées de stderr, la sortie d'erreur standard
- stdout.log - Données réacheminées de stdout, la sortie standard

**Remarque :** Les noms du répertoire .eclipse et des fichiers journaux .dstore\* commencent par un point (.) et sont par conséquent cachés. Utilisez

la commande z/OS UNIX **ls -lA** afin de répertorier les fichiers et répertoires cachés. Lorsque vous utilisez le client Developer for System z, sélectionnez la page de préférences **Fenêtre > Préférences... > Systèmes distants > Fichiers** et activez “Afficher les fichiers cachés”.

- **répertoire de base du démon**

Les fichiers journaux propres au démon RSE et au pool d'unités d'exécution RSE se trouvent dans le répertoire `rep_base_utilisateur`, où `rep_base_utilisateur` est la valeur de la directive `daemon.log` dans `rsed.envvars`. Si la directive `daemon.log` est mise en commentaire ou omise, le répertoire de base de l'ID utilisateur affecté à la tâche démarrée RSED est utilisé. Le répertoire de base est défini dans le segment de sécurité OMVS de l'ID utilisateur.

- `rsedaemon.log` - Journal du démon RSE
- `rseserver.log` - Journal des pools d'unités d'exécution RSE
- `audit.log` - Trace d'audit RSE
- `serverlogs.count` - Compteur des flux du pool d'unités d'exécution RSE
- `stderr.*.log` - Flux d'erreurs standard du pool d'unités d'exécution RSE
- `stdout.*.log` - Fichier stream de sortie standard du pool d'unités d'exécution RSE

**Remarque :** Il existe des commandes de l'opérateur qui permettent de contrôler la quantité de données consignées dans certains des fichiers journaux mentionnés. Pour plus d'informations, voir le Chapitre 8, «Commandes de l'opérateur», à la page 127.

## Journalisation du moniteur de travaux JES

- **SYSOUT DD**

Journalisation des opérations classiques. La valeur par défaut du modèle JCL `FEK.#CUST.PROCLIB(JMON)` est `SYSOUT=*`.

- **SYSPRINT DD**

Journalisation de trace. La valeur par défaut du modèle JCL `FEK.#CUST.PROCLIB(JMON)` est `SYSOUT=*`. La fonction de trace est activée à l'aide du paramètre `-TV`, voir «Fonction de trace du moniteur de travaux JES», à la page 148 pour des informations détaillées.

## Consignation du démon lock

- **STDOUT DD**

Les données réacheminées de `stdout`, la sortie standard Java. La valeur par défaut du modèle JCL `FEK.#CUST.PROCLIB(LOCKD)` est `SYSOUT=*`.

- **STDERR DD**

Les données réacheminées de `stderr`, la sortie des erreurs standard Java. La valeur par défaut du modèle JCL `FEK.#CUST.PROCLIB(LOCKD)` est `SYSOUT=*`.

## Journalisation du démon RSE et du pool d'unités d'exécution

- **STDOUT DD**

Données réacheminées de `stdout`, la sortie Java standard du démon RSE. La valeur par défaut du modèle JCL `FEK.#CUST.PROCLIB(RSED)` est `SYSOUT=*`.

- **STDERR DD**

Données réacheminées de `stderr`, la sortie d'erreur standard Javadu démon RSE. La valeur par défaut du modèle JCL `FEK.#CUST.PROCLIB(RSED)` est `SYSOUT=*`.

- **rép\_base\_utilisateur**

Les fichiers journaux propres au démon RSE et au pool d'unités d'exécution RSE se trouvent dans le répertoire `rép_base_utilisateur`, où `rép_base_utilisateur` est la valeur de la directive `daemon.log` dans `rsed.envvars`. Si la directive `daemon.log` est mise en commentaire ou omise, le répertoire de base de l'ID utilisateur affecté à la tâche démarrée RSED est utilisé. Le répertoire de base est défini dans le segment de sécurité OMVS de l'ID utilisateur.

- `rsedaemon.log` - Journal du démon RSE
- `rseserver.log` - Journal des pools d'unités d'exécution RSE
- `audit.log` - Trace d'audit RSE
- `serverlogs.count` - Compteur des flux du pool d'unités d'exécution RSE
- `stderr.*.log` - Flux d'erreurs standard du pool d'unités d'exécution RSE
- `stdout.*.log` - Fichier stream de sortie standard du pool d'unités d'exécution RSE

**Remarque :**

- `serverlogs.count`, `stderr.*.log` et `stdout.*.log` sont uniquement créés si la directive `enable.standard.log` de `rsed.envvars` est active ou si la fonction est activée dynamiquement avec la commande de l'opérateur **modify rsestandardlog on**.
- Par défaut, le signe `*` dans `stderr.*.log` et `stdout.*.log` est 1. Toutefois, il peut exister plusieurs pools d'unités d'exécution RSE, auquel cas le nombre est incrémenté pour chacun d'eux afin de garantir le caractère unique des noms de fichier.
- Lorsque la directive `enable.standard.log` est active, il n'existe aucun fichier journal `stdout.log` et `stderr.log` propre à l'utilisateur. Les données propres à l'utilisateur sont à présent écrites dans le flux du pool d'unités d'exécution RSE.
- Il existe des commandes de l'opérateur qui permettent de contrôler la quantité de données consignées dans certains des fichiers journaux mentionnés. Pour plus d'informations, voir le Chapitre 8, «Commandes de l'opérateur», à la page 127.

## Journalisation pour l'utilisateur RSE

- **userlog/\$LOGNAME/**

Plusieurs fichiers journaux sont créés par les composants associés à RSE. Ils sont tous placés dans `userlog/$LOGNAME/`, où `userlog` est la valeur combinée des directives `user.log` et `DSTORE_LOG_DIRECTORY` dans `rsed.envvars`, et `$LOGNAME` l'ID utilisateur de connexion (en majuscules). Si la directive `user.log` est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive `DSTORE_LOG_DIRECTORY` est mise en commentaire ou omise, `.eclipse/RSE/` est ajouté à la valeur `user.log`.

- `.dstoreMemLogging` - Consignation sur l'utilisation de la mémoire DataStore
- `.dstoreTrace` - Consignation sur l'action DataStore
- `ffs.log` - Journal du serveur FFS (Foreign File System), qui exécute des fonctions MVS natives
- `ffsget.log` - Journal du programme de lecture de fichier, qui lit un fichier séquentiel ou un membre d'un fichier partitionné
- `ffsput.log` - Journal du programme d'écriture de fichier, qui écrit un fichier séquentiel ou un membre d'un fichier partitionné

- lock.log - Journal du gestionnaire de verrous, qui verrouille et déverrouille un fichier séquentiel ou un membre d'un fichier partitionné
- rmt\_class\_loader.cache.jar - Mémoire cache des classes chargées par le programme de chargement de classe distante RSE
- rsecomm.log - Journal du serveur RSE qui traite les commandes du client et la consignation des communications de tous les services qui utilisent RSE (peut contenir une trace de pile d'exception Java)
- stderr.log - Données réacheminées de stderr, la sortie d'erreur standard
- stdout.log - Données réacheminées de stdout, la sortie standard

**Remarque :**

- Les noms du répertoire .eclipse et des fichiers journaux .dstore\* commencent par un point (.) et sont par conséquent cachés. Utilisez la commande z/OS UNIX **ls -lA** afin de répertorier les fichiers et répertoires cachés. Lorsque vous utilisez le client Developer for System z, sélectionnez la page de préférences **Fenêtre > Préférences... > Systèmes distants > Fichiers** et activez "Afficher les fichiers cachés".
- La création des fichiers journaux .dstore\* est contrôlée par les options de démarrage Java **-DDSTORE\_\*** (voir «Définition de paramètres de démarrage Java supplémentaires avec \_RSE\_JAVAOPTS», à la page 42).
- Les fichiers journaux .dstore\* sont créés en ASCII. Utilisez la commande z/OS UNIX **iconv -f ISO8859-1 -t IBM-1047 .dstore\*** pour les afficher en code EBCDIC (à l'aide de la page de codes IBM-1047).
- Lorsque la directive enable.standard.log est active, il n'existe aucun fichier journal stdout.log et stderr.log propre à l'utilisateur. Les données propres à l'utilisateur sont à présent écrites dans le flux du pool d'unités d'exécution RSE.
- Il existe des commandes de l'opérateur qui permettent de contrôler la quantité de données consignées dans certains des fichiers journaux mentionnés. Pour plus d'informations, voir le Chapitre 8, «Commandes de l'opérateur», à la page 127.

## Journal d'intégration de Fault Analyzer

- userlog/\$LOGNAME/

Journalisation de Fault Analyzer Integration, où userlog est la valeur combinée des directives user.log et DSTORE\_LOG\_DIRECTORY dans rsed.envvars, et \$LOGNAME l'ID utilisateur de connexion (en majuscules). Si la directive user.log est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive DSTORE\_LOG\_DIRECTORY est mise en commentaire ou omise, .eclipse/RSE/ est ajouté à la valeur user.log.

- fa.log - Journal d'intégration de Fault Analyzer
- rsecomm.log - Journal des communications de l'intégration de Fault Analyzer

## Journal d'intégration de File Manager

- userlog/\$LOGNAME/rsecomm.log

Journalisation de la communication de File Manager Integration, où `userlog` est la valeur combinée des directives `user.log` et `DSTORE_LOG_DIRECTORY` dans `rsed.envvars`, et `$LOGNAME` l'ID utilisateur de connexion (en majuscules). Si la directive `user.log` est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive `DSTORE_LOG_DIRECTORY` est mise en commentaire ou omise, `.eclipse/RSE/` est ajouté à la valeur `user.log`.

## Consignation de SCLM Developer Toolkit

- **userlog/\$LOGNAME/rsecomm.log**

Journalisation de la communication de SCLM Developer Toolkit, où `userlog` est la valeur combinée des directives `user.log` et `DSTORE_LOG_DIRECTORY` dans `rsed.envvars`, et `$LOGNAME` l'ID utilisateur de connexion (en majuscules). Si la directive `user.log` est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive `DSTORE_LOG_DIRECTORY` est mise en commentaire ou omise, `.eclipse/RSE/` est ajouté à la valeur `user.log`.

## Journalisation pour le gestionnaire CARMA

- **Travail de serveur CARMA**

Quand vous ouvrez une connexion avec CARMA à l'aide de l'interface de traitement par lots, `FEK.#CUST.SYSPROC(CRASUBMT)` démarre un travail de serveur (avec l'ID utilisateur de l'utilisateur en tant que propriétaires) appelé `CRAport`, où port est le port TCP/IP utilisé.

- **Définition de données CARMALOG**

Si l'instruction de définition de données `CARMALOG` est indiquée dans la méthode de démarrage CARMA sélectionnée, le journal de CARMA est réacheminé vers cette instruction de définition de données dans le travail de serveur. Dans le cas contraire, elle est dirigée vers `SYSPRINT`.

- **SYSPRINT DD**

Le `SYSPRINT` du travail de serveur contient le journal du serveur CARMA, dans le cas où l'instruction de définition de données `CARMALOG` n'est pas définie.

- **userlog/\$LOGNAME/rsecomm.log**

Journalisation de la communication de CARMA, où `userlog` est la valeur combinée des directives `user.log` et `DSTORE_LOG_DIRECTORY` dans `rsed.envvars`, et `$LOGNAME` l'ID utilisateur de connexion (en majuscules). Si la directive `user.log` est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive `DSTORE_LOG_DIRECTORY` est mise en commentaire ou omise, `.eclipse/RSE/` est ajouté à la valeur `user.log`.

## Journalisation de la transaction APPC (service Commandes TSO)

- **SYSPRINT DD**

Quand l'utilitaire d'administration APPC ajoute et modifie un profil de programme transactionnel, il vérifie que le profil du programme transactionnel et son JCL ne contiennent pas d'erreurs de syntaxe. Le résultat de cette phase se compose de messages d'erreur de syntaxe de profil du programme transactionnel, de messages de traitement de l'utilitaire et d'instructions de conversion JCL. La journalisation des messages de cette phase est contrôlée par l'instruction `SYSPRINT DD` pour l'utilitaire `ATBSDFMU`. La valeur par défaut du



modèle JCL FEK.SFEKSAMP(FEKAPPCC) est SYSOUT=\*. Pour des informations détaillées, voir *MVS Planning: APPC/MVS Management* (SA22-7599).

- **&SYSUID.FEKFRSRV.&TPDATE.&TPTIME.LOG**

Quand un programme transactionnel s'exécute, les messages d'exécution associés, tels que les messages d'allocation et de fin, sont envoyés vers un journal appelé par le mot clé MESSAGE\_DATA\_SET dans le profil du programme transactionnel. La valeur par défaut du modèle JCL FEK.SFEKSAMP(FEKAPPCC) est &SYSUID.FEKFRSRV.&TPDATE.&TPTIME.LOG. Pour des informations détaillées, voir *MVS Planning: APPC/MVS Management* (SA22-7599).

**Remarque :** En fonction de vos définitions de transaction APPC et des paramètres par défaut du site, ce fichier journal risque de ne pas apparaître sauf si le mot clé KEEP\_MESSAGE\_LOG(ALWAYS) est ajouté aux définitions de transaction. Pour des informations détaillées, voir *MVS Planning: APPC/MVS Management* (SA22-7599).

## Consignation des tests du programme de vérification de l'installation (IVP) fekfivpi

- **userlog/\$LOGNAME/fekfivpi.log**

Sortie de la commande fekfivpi -file (passerelle client TSO/ISPF associée au test du programme de vérification d'installation), où userlog est la valeur combinée des directives user.log et DSTORE\_LOG\_DIRECTORY dans rsed.envvars, et \$LOGNAME l'ID utilisateur de connexion (en majuscules). Si la directive user.log est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive DSTORE\_LOG\_DIRECTORY est mise en commentaire ou omise, .eclipse/RSE/ est ajouté à la valeur user.log.

## Consignation des tests du programme de vérification de l'installation (IVP) fekfivps

- **userlog/\$LOGNAME/fekfivps.log**

Sortie de la commande fekfivps -file (test du programme de vérification d'installation lié à SCLMDT), où userlog est la valeur combinée des directives user.log et DSTORE\_LOG\_DIRECTORY dans rsed.envvars, et \$LOGNAME l'ID utilisateur de connexion (en majuscules). Si la directive user.log est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive DSTORE\_LOG\_DIRECTORY est mise en commentaire ou omise, .eclipse/RSE/ est ajouté à la valeur user.log.

---

## Fichiers de vidage

Quand un produit subit une fin anormale, un vidage mémoire est créé pour aider à l'identification de l'incident. La disponibilité et l'emplacement de ces fichiers de vidage dépendent pour une grande part des paramètres spécifiques du site. Par conséquent, ils peuvent ne pas être créés, ou être créés à des emplacements différents que ceux mentionnés ci-dessous.

## Fichiers de vidage MVS

Quand le programme fonctionne sous MVS, vérifiez les fichiers de vidage système ainsi que votre JCL pour les instructions de définition de données suivantes (selon le produit) :

- SYSABEND
- SYSMDUMP
- SYSUDUMP
- CEEDUMP
- SYSPRINT
- SYSOUT

Pour plus d'informations sur ces instructions de définition de données, voir *MVS JCL Reference* (SA22-7597) et *Language Environment Debugging Guide* (GA22-7560).

## Fichiers de vidage Java

Dans z/OS UNIX, la plupart des fichiers de vidage de Developer for System z sont commandés par la machine virtuelle Java (JVM).

La JVM crée un ensemble d'agents de vidage par défaut lors de son initialisation (SYSTDUMP et JAVADUMP). Vous pouvez changer cet ensemble d'agents de vidage à l'aide de la variable d'environnement `JAVA_DUMP_OPTS` et même changer l'ensemble à l'aide de `-Xdump` sur la ligne de commande. Les options de ligne de commande de la JVM sont définies dans la directive `_RSE_JAVA0PTS` de `rsed.envvars`. Ne modifiez pas les paramètres de vidage sans instruction du point service IBM.

**Remarque :** Vous pouvez utiliser l'option `-Xdump:what` sur la ligne de commande pour déterminer quels agents de vidage existent à l'exécution du démarrage.

Les types de vidage qui peuvent être produits sont :

### SYSTDUMP

Cliché de transaction Java. Vidage mémoire non formaté généré par z/OS.

Le vidage est consigné dans un fichier séquentiel MVS avec un nom par défaut au format `%uid.JVM.TDUMP.%job.D%ym%d.T%H%M%S`, ou selon la configuration de la variable d'environnement `JAVA_DUMP_TDUMP_PATTERN`. Si vous ne désirez pas que des clichés de transaction soient réalisés, ajoutez la variable d'environnement `IBM_JAVA_ZOS_TDUMP=NO` à `rsed.envvars`.

**Remarque :** `JAVA_DUMP_TDUMP_PATTERN` permet l'utilisation de variables qui sont converties en valeurs réelles lorsque le cliché de la transaction est effectué.

Tableau 20. Variables `JAVA_DUMP_TDUMP_PATTERN`

Variable	Utilisation
<code>%uid</code>	ID utilisateur
<code>%job</code>	Nom du travail
<code>%y</code>	Année (2 chiffres)
<code>%m</code>	Mois (2 chiffres)
<code>%d</code>	Jour (2 chiffres)
<code>%H</code>	Heure (2 chiffres)
<code>%M</code>	Minute (2 chiffres)
<code>%S</code>	Seconde (2 chiffres)



### CEEDUMP

Fichier de vidage Language Environment (LE). Récapitulatif formaté de vidage système qui montre les traces de pile pour chaque unité d'exécution du processus JVM, avec les informations de registre et un stockage de vidage court pour chaque registre.

Le vidage est inscrit dans un fichier z/OS UNIX nommé `CEEDUMP.yyyymmdd.hhmmss.pid`, où `yyymmdd` est la date du jour, `hhmmss` est l'heure et `pid` est l'ID processus en cours. Les emplacements possibles de ce fichier sont décrits dans «Emplacements des fichiers de vidage z/OS UNIX».

### HEAPDUMP

Vidage formaté (liste) des objets qui sont sur le tas Java.

Le vidage est inscrit dans un fichier z/OS UNIX nommé `HEAPDUMP.yyyymmdd.hhmmss.pid.TXT`, où `yyymmdd` est la date du jour, `hhmmss` est l'heure et `pid` est l'ID processus en cours. Les emplacements possibles de ce fichier sont décrits dans «Emplacements des fichiers de vidage z/OS UNIX».

### JAVADUMP

Analyse formatée de la JVM. Contient des données de diagnostic relatives à la JVM et à l'application Java (l'environnement d'application, les unités d'exécution, les piles natives, les verrous et la mémoire, par exemple).

Le vidage est inscrit dans un fichier z/OS UNIX nommé `JAVADUMP.yyyymmdd.hhmmss.pid.TXT`, où `yyymmdd` est la date du jour, `hhmmss` est l'heure et `pid` est l'ID processus en cours. Les emplacements possibles de ce fichier sont décrits dans «Emplacements des fichiers de vidage z/OS UNIX».

Pour plus d'informations sur les fichiers de vidages JVM, voir *Java Diagnostic Guide* (SC34-6358) et pour des informations spécifiques de l'environnement de langage, voir *Language Environment Debugging Guide* (GA22-7560).

## Emplacements des fichiers de vidage z/OS UNIX

La machine virtuelle Java vérifie l'existence et les droits d'accès en écriture pour chacun des emplacements suivants, et stocke les fichiers CEEDUMP, HEAPDUMP et JAVADUMP dans le premier emplacement disponible. Notez que vous devez disposer d'un espace disque suffisant pour que le fichier de vidage soit écrit correctement.

1. Le répertoire dans la variable d'environnement `_CEE_DMPTARG`, s'il est trouvé. Cette variable est définie dans `rsed.envvars` en tant que `/tmp`. Elle peut être remplacée par `/dev/null` afin d'éviter de créer les fichiers de vidage.
2. Le répertoire de travail en cours, s'il ne s'agit pas du répertoire de base (`/`), et qu'il est inscriptible.
3. Le répertoire dans la variable d'environnement `TMPDIR` (une variable d'environnement qui indique l'emplacement d'un répertoire temporaire autre que `/tmp`), s'il est trouvé.
4. Le répertoire `/tmp`.
5. Si le vidage ne peut être stocké dans aucun des emplacements précédents, il est mis dans `stderr`.

---

## Fonction de trace

### Fonction de trace du moniteur de travaux JES

La fonction de trace du moniteur de travaux JES est contrôlée par l'opérateur système dans Chapitre 8, «Commandes de l'opérateur», à la page 127.

- Le lancement de la tâche activée JMON avec le paramètre PRM=-TV active le mode prolix (fonction de trace)
- Les commandes **modify -TV** et **modify -TN** permettent d'activer et de désactiver la fonction de trace

### Fonction de trace RSE

Plusieurs fichiers journaux sont créés par les composants associés à RSE. La plupart se trouve dans `userlog/$LOGNAME/`, où `userlog` est la valeur combinée des directives `user.log` et `DSTORE_LOG_DIRECTORY` dans `rsed.envvars`, et `$LOGNAME` l'ID utilisateur de connexion (en majuscules). Si la directive `user.log` est mise en commentaire ou omise, le chemin du répertoire de base est utilisé. Ce chemin est défini dans le segment de sécurité OMVS de l'ID utilisateur. Si la directive `DSTORE_LOG_DIRECTORY` est mise en commentaire ou omise, `.eclipse/RSE/` est ajouté à la valeur `user.log`.

La quantité de données consignées dans `ffs*.log`, `lock.log` et `rsecomm.log` est déterminée par la commande opérateur **modify rsecommlog** ou par le paramètre `debug_level` dans `rsecomm.properties`. Pour plus d'informations, voir le Chapitre 8, «Commandes de l'opérateur», à la page 127 et la section «(Facultatif) Fonction de trace RSE», à la page 98.

La création de fichiers journaux `.dstore*` est contrôlée par les options de démarrage Java `-DDSTORE_*` (voir «Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JVAOPTS`», à la page 42).

#### Remarque :

- Les noms du répertoire `.eclipse` et des fichiers journaux `.dstore*` commencent par un point (.) et sont par conséquent cachés. Utilisez la commande z/OS UNIX **ls -lA** afin de répertorier les fichiers et répertoires cachés. Lorsque vous utilisez le client Developer for System z, sélectionnez la page de préférences **Fenêtre > Préférences... > Systèmes distants > Fichiers** et activez "Afficher les fichiers cachés".
- Les fichiers journaux `.dstore*` sont créés en ASCII. Utilisez la commande z/OS UNIX **iconv -f ISO8859-1 -t IBM-1047 .dstore\*** pour les afficher en code EBCDIC (lors de l'utilisation de la page de codes IBM-1047).

Les fichiers journaux propres au démon RSE et au pool d'unités d'exécution RSE se trouvent dans le répertoire `rep_base_utilisateur`, où `rep_base_utilisateur` est la valeur de la directive `daemon.log` dans `rsed.envvars`. Si la directive `daemon.log` est mise en commentaire ou omise, le répertoire de base de l'ID utilisateur affecté à la tâche démarrée RSED est utilisé. Le répertoire de base est défini dans le segment de sécurité OMVS de l'ID utilisateur.

La quantité de données consignées dans `rsedaemon.log` et `rserver.log` est commandée par les commandes de l'opérateur **modify rsedaemonlog** et **modify rserverlog** ou par le paramètre `debug_level` de `rsecomm.properties`. Pour plus

d'informations, voir Chapitre 8, «Commandes de l'opérateur», à la page 127 et «(Facultatif) Fonction de trace RSE», à la page 98.

`serverlogs.count`, `stderr.*.log` et `stdout.*.log` sont uniquement créés si la directive `enable.standard.log` de `rsed.envvars` est active ou si la fonction est activée dynamiquement avec la commande de l'opérateur **modify rsestandardlog on**.

## Traçage du démon lock

Le journal propre au démon lock se trouve dans la définition de données STDOUT DD de la tâche démarrée LOCKD. Le volume des données consignées dans le journal est déterminé par le paramètre de démarrage LOG. Pour plus d'informations, voir Chapitre 8, «Commandes de l'opérateur», à la page 127 et «(Facultatif) Fonction de trace RSE», à la page 98.

## Fonction de trace CARMA

L'utilisateur peut contrôler la quantité d'informations de trace générées par CARMA en définissant le niveau de trace dans l'onglet des propriétés de la connexion CARMA du client. Les différents Niveaux de trace sont les suivants :

- Désactiver la journalisation
- Journalisation des erreurs
- Journalisation des avertissements
- Journalisation informative
- Déboguer la journalisation

La valeur par défaut est la suivante :  
Journalisation des erreurs

Pour plus d'informations sur l'emplacement des fichiers journaux, voir «Fichiers journaux», à la page 140.

## Traçage de suivi des erreurs

La procédure suivante permet de rassembler les informations nécessaires au diagnostic des incidents de suivi des erreurs avec les procédures d'assemblage à distance. La fonction de trace réduit les performances et ne doit être effectuée que sur indication du centre de support IBM. Toutes les références à h1q que vous trouverez dans cette section se rapportent au qualificatif de haut niveau utilisé au cours de l'installation de Developer for System z. L'installation par défaut est FEK, mais peut ne pas s'appliquer à votre site.

1. Faites une copie de sauvegarde de votre procédure de compilation active ELAXFCOC. Cette procédure est normalement présente dans le fichier h1q.SFEKSAMP à la livraison, mais peut avoir été copiée à un emplacement différent, comme SYS1.PROCLIB (voir «Procédures de construction à distance ELAXF\*», à la page 25).
2. Modifiez la procédure active ELAXFCOC pour inclure la chaîne 'MAXTRACE' dans l'option de compilation EXIT(ADEXIT(ELAXMGUX)).

```
//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)
```

**Remarque :** Vous devez doubler les apostrophes pour MAXTRACE. L'option est maintenant : EXIT(ADEXIT(' 'MAXTRACE' ',ELAXMGUX)).

3. Effectuez une vérification de la syntaxe à distance sur le programme en langage COBOL pour lequel vous souhaitez obtenir des données de trace détaillées.
4. La partie SYSOUT de la sortie JES démarre en générant la liste des noms de fichier pour SIDEFILE1, SIDEFILE2, SIDEFILE3 et SIDEFILE4.

```
ABOUT TOO OPEN SIDEFILE1 - NAME = 'uid.DT021207.TT110823.M0000045.C0000000'  
SUCCESSFUL OPEN SIDEFILE1 - NAME = 'uid.DT021207.TT110823.M0000045.C0000000'  
ABOUT TOO OPEN SIDEFILE2 - NAME = 'uid.DT021207.TT110823.M0000111.C0000001'  
SUCCESSFUL OPEN SIDEFILE2 - NAME = 'uid.DT021207.TT110823.M0000111.C0000001'  
ABOUT TOO OPEN SIDEFILE3 - NAME = 'uid.DT021207.TT110823.M0000174.C0000002'  
SUCCESSFUL OPEN SIDEFILE3 - NAME = 'uid.DT021207.TT110823.M0000174.C0000002'  
ABOUT TOO OPEN SIDEFILE4 - NAME = 'uid.DT021207.TT110823.M0000236.C0000003'  
SUCCESSFUL OPEN SIDEFILE4 - NAME = 'uid.DT021207.TT110823.M0000236.C0000003'
```

**Remarque :** Selon vos paramètres, SIDEFILE1 et SIDEFILE2 peuvent pointer vers une instruction de définition de données (SUCCESSFUL OPEN SIDEFILE1 - NAME = DD:WSEDSF1). Reportez-vous au composant JESJCL de la sortie (qui est situé avant le composant SYSOUT) pour connaître le nom réel du fichier.

```
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. Copiez ces quatre fichiers sur votre PC, par exemple en créant un projet local en langage COBOL dans Developer for System z et en ajoutant les fichiers SIDEFILE1->4.
6. Copiez le journal de travail JES sur votre PC, par exemple en ouvrant la sortie de travaux dans Developer for System z et en l'enregistrant dans le projet local. Pour cela, sélectionnez **Fichier > Enregistrer sous...** .
7. Restaurez la procédure ELAXFC0C à son état original, soit en annulant les modifications (retirez la chaîne "MAXTRACE" des options de compilation) soit en restaurant la sauvegarde.
8. Envoyez les fichiers collectés (SIDEFILE1->4 et le journal de travail) au point service IBM.

---

## Données de droits z/OS UNIX

Developer for System z requiert que le système de fichiers z/OS UNIX et certains fichiers z/OS UNIX comportent des données de droits spécifiques définies.

### Attribut du système de fichiers SETUID

L'Explorateur de systèmes éloignés (RSE) est le composant de Developer for System z qui fournit les services de base, comme la connexion du client à l'hôte. Il doit pouvoir effectuer des tâches telles que la création de l'environnement de sécurité de l'utilisateur.

Le système de fichiers (HFS ou zFS) dans lequel Developer for System z est installé doit être monté avec le contrôle des données de droits SETUID activé (il s'agit de la valeur par défaut du système). Le montage du système de fichier avec le paramètre NOSETUID empêchera Developer for System z de créer l'environnement de sécurité utilisateur et l'exécution de la requête de connexion.

Utilisez la commande TSO **ISHELL** afin de répertorier l'état actuel des données SETUID. Dans le panneau ISHELL, sélectionnez **Systèmes\_de\_fichiers > 1. Table de**

**montage...** afin de répertorier les systèmes de fichiers montés. La commande-ligne **a** affichera les attributs du système de fichiers sélectionné où le champ "Ignorer SETUID" sera 0.

## Autorisation de contrôle de programmes

L'Explorateur de systèmes éloignés (RSE) est le composant de Developer for System z qui fournit les services de base, comme la connexion du client à l'hôte. L'exécution doit être contrôlée par le programme afin d'effectuer des tâches telles que la commutation sur l'ID utilisateur du client.

Les données de contrôle de programmes z/OS UNIX sont définies au cours de l'installation SMP/E si nécessaire, sauf pour l'interface Java à votre produit de sécurité (voir le Chapitre 10, «Remarques relatives à la sécurité», à la page 161). Cette donnée de droits pourrait être perdue si vous ne la conservez pas lors de la copie manuelle des répertoires Developer for System z.

Les fichiers Developer for System z suivants doivent être contrôlés par programme :

- /usr/lpp/rdz/bin/
  - fekfdivp
  - fekfomvs
  - fekfriwp
- /usr/lpp/rdz/lib/
  - fekfdir.dll
  - libfekdcore.so
  - libfekfmain.so
- /usr/lpp/rdz/lib/icuc/
  - libicudata.dll
  - libicudata40.1.dll
  - libicudata40.dll
  - libicudata64.40.1.dll
  - libicudata64.40.dll
  - libicudata64.dll
  - libicuuc.dll
  - libicuuc40.1.dll
  - libicuuc40.dll
  - libicuuc64.40.1.dll
  - libicuuc64.40.dll
  - libicuuc64.dll

**Remarque :** Les fichiers libicu\*64.\* ne sont présents que si vous avez appliqué le PTF Developer for System z qui indique à APAR AM07305 d'activer la prise en charge 64 bits.

Utilisez la commande z/OS UNIX **ls -E** pour répertorier les attributs étendus dans lesquels le bit de contrôle par programme est marqué avec la lettre **p**, comme indiqué dans l'exemple ci-après (\$ est l'invite z/OS UNIX) :

```
$ cd /usr/lpp/rdz
$ ls -E lib/fekf*
-rwxr-xr-x -ps- 2 user      group      94208 Jul  8 12:31 lib/fekfdir.dll
```

Utilisez la commande z/OS UNIX **extattr +p** afin de définir les données de contrôle de programmes manuellement, comme indiqué dans l'exemple suivant (\$ et # représentent l'invite z/OS UNIX) :

```
$ 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
```

**Remarque :** Pour utiliser la commande **extattr +p**, vous devez disposer au moins d'un accès READ au profil BPX.FILEATTR.PROGCTL dans la classe FACILITY de votre logiciel de sécurité ou être un superutilisateur (UID 0) si ce profil n'est pas défini. Pour plus d'informations, voir *UNIX System Services Planning* (GA22-7800).

## Autorisation APF

Remote Systems Explorer (RSE) est le composant de Developer for System z qui fournit les services de base comme la connexion du client à l'hôte. Il doit exécuter un APF autorisé pour effectuer des tâches comme l'affichage de l'usage détaillé des ressources du processus.

Le bit APF z/OS UNIX est défini durant l'installation de SMP/E, si nécessaire. Cette donnée de droits pourrait être perdue si vous ne la conservez pas au cours d'une copie manuelle des répertoires de Developer for System z.

Les fichiers Developer for System z suivants doivent être autorisés APF :

- /usr/lpp/rdz/bin/
  - fekfomvs
  - fekfrivp

Utilisez la commande z/OS UNIX **ls -E** pour répertorier les attributs étendus dans lesquels le bit APF est marqué avec la lettre a, comme indiqué dans l'exemple ci-après (\$ est l'invite z/OS UNIX):

```
$ cd /usr/lpp/rdz
$ ls -E bin/fekfrivp
-rwxr-xr-x  aps-  2 user      group      114688 Sep 17 06:41 bin/fekfrivp
```

Utilisez la commande z/OS UNIX **extattr +a** pour définir le bit APF manuellement, comme indiqué dans l'exemple suivant (\$ et # sont les invites z/OS UNIX):

```
$ 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
```

**Remarque :** Pour utiliser la commande **extattr +a**, vous devez disposer au moins d'un accès READ au profil BPX.FILEATTR.APF dans la classe FACILITY de votre logiciel de sécurité ou être un superutilisateur (UID 0) si ce profil n'est pas défini. Pour plus d'informations, voir *UNIX System Services Planning* (GA22-7800).

## Données de rappel

Certains services Developer for System z facultatifs requièrent que les modules de chargement MVS soient disponibles pour z/OS UNIX. Pour ce faire, créez un fichier fictif dans z/OS UNIX avec le bit "sticky" activé. Lorsque le module de remplacement est exécuté, z/OS UNIX recherche un module de chargement MVS portant le même nom, puis l'exécute.

Le bit "sticky" de z/OS UNIX est défini pendant l'installation SMP/E, si nécessaire. Ces bits d'autorisation peuvent être perdus si vous ne les conservez pas lors de la copie manuelle des répertoires Developer for System z.

Les fichiers Developer for System z doivent avoir le bit "sticky" activé :

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

Utilisez la commande **ls -l** de z/OS UNIX pour afficher les droits. Les données de droit sont marquées par la lettre avec t, comme indiqué dans l'exemple ci-après (\$ représente l'invite de z/OS UNIX) :

```
$ cd /usr/lpp/rdz
$ ls -l bin/CRA*
-rwxr-xr-t  2 user      group          71 Jul  8 12:31 bin/CRASTART
```

Utilisez la commande **chmod +t** de z/OS UNIX pour définir le bit "sticky" manuellement, comme indiqué dans l'exemple ci-après (\$ et # représentent l'invite de z/OS UNIX) :

```
$ 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
```

**Remarque :** Afin de pouvoir utiliser la commande **chmod**, vous devez disposer au minimum des droits d'accès READ au profil SUPERUSER.FILESYS.CHANGEPERMS dans la classe UNIXPRIV de votre logiciel de sécurité ou être un superutilisateur (UID 0) si ce profil n'est pas défini. Pour plus d'informations, voir *UNIX System Services Planning* (GA22-7800).

---

## Ports TCP/IP réservés

A l'aide de la commande **netstat** (TSO ou z/OS UNIX), vous pouvez connaître les ports actuellement utilisés. Le résultat de cette commande doit s'apparenter à l'exemple ci-dessous. Les ports utilisés sont le dernier chiffre (derrière les deux points "..") dans la colonne "Local Socket". Ces ports étant déjà utilisés, vous ne pouvez pas vous en servir pour la configuration de Developer for System z.

### IPv4

MVS	TCP/IP	NETSTAT	CS	VxRy	TCP/IP Name:	TCPIP	16:36:42
User	Id	Conn	Local	Socket	Foreign	Socket	State
-----	----	-----	-----	-----	-----	-----	-----
BPXOINIT	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

```

Les ports TCP/IP réservés peuvent présenter une autre limitation. Il existe deux espaces communs pour réserver des ports TCP/IP :

- **PROFILE.TCPIP**

Il s'agit du fichier auquel se rapporte l'instruction de définition de données PROFILE de la tâche lancée par TCP/IP, souvent appelée SYS1.TCPPARMS(TCPPROF).

- PORT : réserve un port pour des noms de travaux spécifiés.
- PORTRANGE : réserve une plage de ports pour des noms de travaux spécifiés.

Pour plus d'informations sur ces instructions, voir *Communications Server: IP Configuration Guide* (SC31-8775).

- **SYS1.PARMLIB(BPXPRMxx)**

- INADDRANYPORT : indique le numéro de port de démarrage pour la série de numéros de port que le système réserve pour être utilisés avec les liaisons PORT 0, INADDR\_ANY. Cette valeur est uniquement nécessaire pour CINET (plusieurs piles TCP/IP actives sur un seul hôte).
- INADDRANYCOUNT : indique le nombre de ports que le système réserve, en commençant par le numéro de port spécifié dans le paramètre INADDRANYPORT. Cette valeur est uniquement nécessaire pour CINET (plusieurs piles TCP/IP actives sur un seul hôte).

Pour plus d'informations sur ces instructions, consultez les documents *UNIX System Services Planning* (GA22-7800) et *MVS Initialization and Tuning Reference* (SA22-7592).

Les ports réservés peuvent être répertoriés à l'aide de la commande **netstat portl** (TSO ou z/OS UNIX), qui crée une sortie comparable à celle de l'exemple ci-dessous :

```

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

```



Pour plus d'informations sur la commande **NETSTAT**, voir le document *Communications Server: IP System Administrator's Commands* (SC31-8781).

**Remarque :** La commande **NETSTAT** présente uniquement les informations définies dans **PROFILE.TCPIP**, qui doivent coïncider avec les définitions **BPXPRMxx**. En cas de doute ou d'incident, vérifiez dans le membre **parmlib BPXPRMxx** les ports réservés dans ce cas.

---

## Taille d'espace adresse

Le démon **RSE**, qui est un processus **z/OS UNIX Java**, requiert une taille de région élevée pour exécuter ses fonctions. Il est donc important de définir des limites de mémoire importantes pour les espaces adresse **OMVS**.

## Exigences liées au langage JCL de démarrage

Le démon **RSE** est démarré par le **JCL** à l'aide de **BPXBATSL**, dont la taille de la région doit être 0.

## Limitations définies dans SYS1.PARMLIB(BPXPRMxx)

Attribuez la valeur **2G** à **MAXASSIZE** dans **SYS1.PARMLIB(BPXPRMxx)** pour définir la taille de la région de l'espace adresse (processus) **OMVS** par défaut. Il s'agit de la valeur maximale autorisée. Il s'agit d'une limite à l'échelle du système. Elle est donc active pour tous les espaces adresses **z/OS UNIX**. Si elle ne répond pas à vos attentes, vous pouvez la définir uniquement pour **Developer for System z** dans votre logiciel de sécurité.

Cette valeur peut être vérifiée et définie de manière dynamique (jusqu'au prochain démarrage du système) à l'aide des commandes de console suivantes, décrites dans le document *MVS System Commands* (GC28-1781):

1. **DISPLAY OMVS,0**
2. **SETOMVS MAXASSIZE=2G**

## Limitations stockées dans le profil de sécurité

Vérifiez **ASSIZEMAX** dans le segment **OMVS** de l'**ID** utilisateur du démon et définissez-le à **2147483647** ou, de préférence, à **NONE** pour utiliser la valeur **SYS1.PARMLIB(BPXPRMxx)**.

Avec **RACF**, cette valeur peut être vérifiée et définie à l'aide des commandes **TSO** suivantes, décrites dans le document *Security Server RACF Command Language Reference* (SA22-7687) :

1. **LISTUSER userid NORACF OMVS**
2. **ALTUSER userid OMVS(NOASSIZEMAX)**

## Limitations forcées par les sorties du système

Assurez-vous que vous n'autorisez pas aux sorties du système **IEFUSI** ou **IEALIMIT** de contrôler les tailles de région d'adresse **OMVS**. Il est possible de réaliser cela en codant **SUBSYS(OMVS,NOEXITS)** dans **SYS1.PARMLIB(SMFPRMxx)**.

Les valeurs de **SYS1.PARMLIB(SMFPRMxx)** peuvent être vérifiées et définies à l'aide des commandes de console suivantes, décrites dans le document *MVS System Commands* (GC28-1781) :

1. **DISPLAY SMF,0**

## 2. SET SMF=xx

### Limitations pour adressage 64 bits

Le mot clé MEMLIMIT dans SYS1.PARMLIB(SMFPRMxx) limite la quantité de stockage virtuel qu'une tâche 64 bits peut allouer au-delà de la barre des 2 Go.

Contrairement au paramètre REGION du JCL, MEMLIMIT=0M signifie qu'il n'est pas possible d'utiliser un stockage virtuel au-delà de la barre.

Si MEMLIMIT n'est pas spécifié dans SMFPRMxx, la valeur par défaut est 0M, et donc des tâches sont liées aux 2 Go (31 bits) situés en dessous de la barre. La valeur par défaut changée sous z/OS 1.10 pour 2 G, ce qui permet à une tâche 64 bits d'utiliser jusqu'à 4 GB (les 2 Go en dessous de la barre et les 2 Go au-dessus de la barre, accordés par MEMLIMIT).

Les valeurs de SYS1.PARMLIB(SMFPRMxx) peuvent être vérifiées et définies à l'aide des commandes de console suivantes, décrites dans le document *MVS System Commands* (GC28-1781) :

1. DISPLAY SMF,0
2. SET SMF=xx

Vous pouvez aussi spécifier MEMLIMIT comme paramètre sur une carte EXEC dans un JCL. Si aucun paramètre MEMLIMIT n'est spécifié, la valeur par défaut est la valeur définie pour SMF, en revanche, si REGION=0M est spécifié, la valeur par défaut est NOLIMIT.

---

## Transaction APPC et service Commandes TSO

Si vous ne pouvez pas utiliser la version APPC du service Commandes TSO, des incidents peuvent survenir dans les deux domaines suivants : le démarrage de la transaction du serveur APPC et la connexion au RSE.

- Si aucun message sur la configuration d'APPC n'apparaît, recherchez dans le journal système les messages RACF (ID message ICHxxxxx) ou les autres messages concernant la commande lancée ou l'ID utilisateur qui a émis la commande. Les causes fréquentes des incidents sont notamment les suivantes :
  - Vous ne disposez pas des droits d'accès en lecture sur le fichier FEK.SFEKPROC.
  - Le protocole TCP/IP n'est pas actif, possède un nom DNS incorrect ou le système n'est pas accessible (la commande ping ne peut s'y exécuter) à cause d'incidents réseau, d'une adresse IP incorrecte ou d'autres causes.
- Si vous visualisez les messages de configuration APPC mais qu'aucun message ne confirme que la configuration a réussi, la transaction du serveur APPC n'est peut-être pas parvenue à démarrer. Consultez le journal des erreurs de transaction (userid.FEKFRSRV.&TPDATE.&TPTIME.LOG). Parmi les causes fréquentes d'incident, vous trouverez :
  - La pile TCP/IP n'utilise pas le nom par défaut TCPIP et la carte de pilote de périphérique SYSTCPD n'a pas été définie ou indique un fichier incorrect.
  - Le serveur ne parvient pas à allouer SYSPROC ou SYSTSPRT.
  - Le langage JCL pointe sur un SYSPROC incorrect (SYSPROC doit inclure FEK.SFEKPROC).
  - Le serveur ne parvient pas à ouvrir ou à accéder au fichier de messages (journal) indiqué par MESSAGE\_DATA\_SET.
  - Le nombre de demandeurs de planificateur APPC est insuffisant.
  - Les emplacements de l'adresse APPC ou ASCH ne sont pas actifs.

- La classe utilisée (nommée "A" par défaut) n'est pas définie pour l'ASCH du planificateur APPC.
- Il n'y a pas de segment OMVS par défaut pour le système et l'utilisateur ne possède pas de segment OMVS personnel, ou il y a une erreur de définition dans les deux.
- Le groupe par défaut du segment OMVS par défaut ou le groupe par défaut de l'utilisateur ne présente pas de numéro d'ID groupe.

Le REXX fourni dans «(Facultatif) Transaction APPC pour le service Commandes TSO», à la page 106 permet de résoudre les incidents APPC en vous donnant la possibilité de gérer les APPC de manière interactive via les panneaux ISPF. Toutefois, vous devez savoir que vous pouvez désactiver la transaction avec cet outil ; la transaction existe toujours mais n'accepte aucune connexion.

La liste suivante répertorie les fiches techniques disponibles dans la section Support du site Web (<http://www-306.ibm.com/software/awdtools/rdz/support/>). Pour de plus amples informations, consultez le site Web de support :

- Echec de la vérification APPC avec un code retour 2016 - EHOSTNOTFOUND
- Echec de la vérification APPC avec un code retour 1004 - EIBMIUCVERR
- Echec de la vérification APPC avec un code retour 9 - Nom du programme transactionnel non reconnu
- Echec de la vérification APPC avec un code retour 10 - Programme transactionnel non disponible, pas de relance
- Echec de la vérification APPC avec un code retour 19 - Erreur de paramètre
- Echec de la vérification APPC avec un code retour 20 - Erreur propre au produit
- Echec de la vérification APPC avec un code retour 26 - Panne de la ressource
- CEE3501S : Module IOSTREAM introuvable
- Echec du démarrage du serveur : EDC5129I Ce fichier ou ce répertoire est introuvable
- exec/tcp : Liaison : EDC5111I Droit refusé, rsn=744C7246
- Aucune réponse du serveur, avec l'un des messages suivants :
  - IEA995I SYMPTOM DUMP OUTPUT 473 USER COMPLETION CODE=4093 REASON CODE=0000001C (in SDSF LOG)
  - CEE3512S Le chargement du système hiérarchique de fichiers du module libicudata32.0.dll a échoué. Le code retour du système est 0000000157 ; le code anomalie est 0BDF019B. (dans CEEDUMP)
  - Echec d'obtention de mémoire (dans client .log)
- La commande C\_CONNECT n'est pas disponible
- "Echec de l'initialisation du serveur FFS" message d'erreur lors de la tentative de connexion à l'hôte
- "EDC5139I Opération non autorisée" lors de la tentative de connexion à l'hôte
- "RSEG1056U Echec de l'initialisation du serveur FFS" à l'ouverture d'un fichier MVS

**Remarque :** Cette liste n'est pas définitive. Consultez le site Web du support pour obtenir des fiches techniques supplémentaires.

---

## Informations diverses

### Limites du système

SYS1.PARMLIB(BPXPRMxx) définit plusieurs limitations liées à z/OS UNIX, qui peuvent être atteintes lorsque plusieurs clients de Developer for System z sont actifs. La plupart des valeurs BPXPRMxx peuvent être modifiées de façon dynamique avec les commandes de la console **SETOMVS** et **SET OMVS**.

Utilisez la commande de console **SETOMVS LIMMSG=ALL** pour que z/OS UNIX affiche les messages de console (BPXI040I) lorsque l'une des limites BPXPRMxx est sur le point d'être atteinte.

### Connexion refusée

Chaque connexion RSE lance plusieurs processus qui sont actifs de façon permanente. De nouvelles connexions peuvent être refusées en raison de la limite définie dans SYS1.PARMLIB(BPXPRMxx) concernant la quantité de processus, particulièrement lorsque les utilisateurs partagent le même ID utilisateur (lorsque le segment OMVS par défaut est utilisé, par exemple).

- La limite par ID utilisateur est définie par le mot clé MAXPROCUSER, et a une valeur par défaut de 25.
- La limite au niveau du système est définie par le mot clé MAXPROCSYS, et a une valeur par défaut de 200.

La limite d'espaces adresse z/OS et d'utilisateurs z/OS UNIX actifs représente une autre source de connexions refusées.

- La quantité maximale d'ID espace adresse (ASID) est définie dans SYS1.PARMLIB(IEASYSxx) avec le mot-clé MAXUSER et présente une valeur par défaut de 255.
- La quantité maximale d'ID utilisateur (UID)z/OS UNIX est définie dans SYS1.PARMLIB(BPXPRMxx) avec le mot clé MAXUIDS, et présente une valeur par défaut de 200.

### Incidents connus relatifs aux éléments requis

#### Echec de l'ouverture des fichiers MVS

Lorsque vous utilisez APPC pour le service Commandes TSO, la lecture et l'écriture d'un fichier MVS nécessitent l'emploi d'un socket de domaine de système de fichiers physique. Si le système de fichiers n'est pas correctement défini ou s'il n'a pas suffisamment de sockets, le gestionnaire de verrous (FFS) peut échouer dans les requêtes de lecture/écriture. Les fichiers ffs\*.log présenteront des messages comme :

- Erreur 127 à l'obtention de la paire de sockets - port défini à 0.
- Impossible de créer le socket dans le domaine UNIX. L'erreur est : "La famille d'adresses n'est pas prise en charge"

Assurez-vous que le membre SYS1.PARMLIB(BPXPRMxx) contient les instructions suivantes :

```
FILESYSTYPE TYPE(UDS) ENTRYPPOINT(BPXTUINT)
NETWORK DOMAINNAME(AF_UNIX)
          DOMAINNUMBER(1)
          MAXSOCKETS(2000)
          TYPE(UDS)
```

Un autre motif pouvant être à l'origine de cet incident lorsque vous utilisez APPC pour le service Commandes TSO est que le programme de résolution TCP/IP ne parvient pas à résoudre l'adresse hôte correctement en raison d'un fichier de configuration du programme de résolution manquant ou incomplet. Le message lock.log suivant est une indication évidente de l'apparition de cet incident :

```
clientip(0.0.0.0) <> callerip(<adresse IP de l'hôte>)
```

Exécutez le programme de vérification d'installation TCP/IP fekfivpt tel que décrit au Chapitre 7, «Vérification de l'installation», à la page 111. La section relative à la configuration du programme de résolution de la sortie ressemblera à l'exemple suivant :

```
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
```

Vérifiez que les définitions du fichier référencées par “Fichier Tcp/Ip local” sont correctes.

Cette zone est vide si vous n'utilisez pas un nom par défaut pour le fichier du programme de résolution IP (à l'aide de l'ordre de recherche z/OS UNIX). Si tel est le cas, ajoutez l'instruction suivante dans rsed.envvars, où <fichier du programme de résolution> ou <données du programme de résolution représente le nom de votre fichier de programme de résolution IP :

```
RESOLVER_CONFIG=<fichier du programme de résolution>
```

ou

```
RESOLVER_CONFIG='<fichier du programme de résolution>'
```

## Emulateur de connexion à l'hôte

- L'émulateur de connexion à l'hôte utilise telnet TN3270 et non le serveur RSE pour se connecter à l'hôte.
- Lorsque vous utilisez le Telnet sécurisé (SSL) et que vous travaillez avec des certificats qui ne sont pas signés par une autorité de certification bien connue, chaque client doit ajouter le certificat de CA à la liste d'autorités de certification dignes de confiance de son émulateur de connexion à l'hôte.
- L'option NOSNAEXT de TELNETPARMS de TCP/IP peut être nécessaire pour désactiver les extensions fonctionnelles de l'architecture SNA. Dans le cas où NOSNAEXT est indiqué, le serveur telnet TN3270 ne négocie pas pour la résolution des conflits et les fonctions de détection d'architecture SNA.



---

## Chapitre 10. Remarques relatives à la sécurité

Developer for System z fournit un accès grand système aux utilisateurs d'un poste de travail standard. La validation des demandes de connexion, l'établissement de communications sécurisées entre l'hôte et le poste de travail, l'autorisation et l'activité d'audit sont donc des aspects fondamentaux de la configuration d'un produit.

Pour être efficaces, les mécanismes de sécurité utilisés par les serveurs et les services Developer for System z doivent résider dans un système de fichiers sécurisé. Cela implique que seuls les administrateurs système habilités doivent pouvoir mettre à jour les bibliothèques de programmes et les fichiers de configuration.

Les rubriques suivantes sont traitées dans le présent chapitre :

- «Méthodes d'authentification»
- «Sécurité des connexions», à la page 163
- «Ports TCP/IP», à la page 164
- «Utilisation de PassTickets», à la page 166
- «Consignation dans le journal d'audit», à la page 167
- «Sécurité JES», à la page 168
- «Communication chiffrée à l'aide du protocole SSL», à la page 172
- «Authentification du client à l'aide de certificats X.509», à la page 173
- «Vérification du port d'entrée (POE)», à la page 176
- «Sécurité CICSTS», à la page 177
- «Sécurité SCLM», à la page 177
- «Fichiers de configuration Developer for System z», à la page 178
- «Définitions de sécurité», à la page 179

**Remarque :** L'Explorateur de systèmes distants (RSE), qui fournit des services de base comme la connexion du client à l'hôte, se compose de deux entités logiques.

- Le démon RSE qui gère la configuration de la connexion et qui est lancé en tant que tâche démarrée ou en tant que travail à exécution longue.
- Le serveur RSE qui gère des demandes client individuelles et qui est démarré en tant qu'unité d'exécution dans un ou plusieurs processus enfant par le démon RSE.

Voir le Chapitre 11, «Compréhension de Developer for System z», à la page 193 pour en savoir plus sur la conception de base de Developer for System z.

---

### Méthodes d'authentification

Developer for System z prend en charge plusieurs méthodes d'authentification d'un ID utilisateur fourni par un client lors de la connexion.

- Un ID utilisateur et un mot de passe
- Un ID utilisateur et un mot de passe utilisable une seule fois

- Un certificat X.509

Les données d'authentification fournies par le client ne sont utilisées qu'une seule fois, pendant la configuration de la connexion initiale. Une fois que l'ID utilisateur est authentifié, l'ID utilisateur et les mots de passe PassTicket générés automatiquement sont utilisés pour toutes les actions qui requièrent une authentification.

## **ID utilisateur et mot de passe**

Le client fournit un ID utilisateur et un mot de passe lors de la connexion. L'ID utilisateur et le mot de passe sont utilisés pour authentifier l'utilisateur auprès de votre logiciel de sécurité.

## **ID utilisateur et mot de passe utilisable une seule fois**

Un mot de passe utilisable une seule fois peut être généré par un produit tiers à partir d'un jeton unique. Ce type de mot de passe renforce la configuration de sécurité car le sème unique ne peut pas être copié ni être utilisé sans que l'utilisateur en soit informé et un mot de passe intercepté est inutilisable car il n'est valide qu'une seule fois.

Lors de la connexion, le client indique un ID utilisateur et un mot de passe utilisable une seule fois, qui permet d'authentifier l'ID utilisateur avec l'exit de sécurité fourni par le tiers. Cet exit de sécurité doit ignorer les mots de passe PassTicket utilisés pour traiter les demandes d'authentification lors d'un traitement normal. Les mots de passe PassTicket doivent être traités par votre logiciel de sécurité.

## **Certificat X.509**

Un tiers peut fournir un ou plusieurs certificats X.509 qui permettent l'authentification d'un utilisateur. Lorsqu'ils sont stockés sur des unités sécurisées, les certificats X.509 offrent une configuration sécurisée associée à une grande facilité d'utilisation (pas d'ID utilisateur ni de mot de passe nécessaires).

Lors de la connexion, le client fournit un certificat sélectionné et éventuellement une extension, qui permet d'authentifier l'ID utilisateur auprès de votre logiciel de sécurité.

Cette méthode d'authentification est prise en charge uniquement par la méthode de connexion du démon RSE. SSL doit également être activé.

## **Authentification du moniteur de travaux JES**

L'authentification du client est effectuée par le démon RSE (ou REXEC/SSH) dans le cadre d'une demande de connexion client. Une fois que l'utilisateur est authentifié, des mots de passe PassTicket générés automatiquement sont utilisés pour toutes les demandes d'authentification ultérieures, y compris la connexion automatique au moniteur de travaux JES.

Pour que le moniteur de travaux JES puisse valider l'ID utilisateur et le mot de passe PassTicket présenté par RSE, il doit être autorisé à évaluer le mot de passe PassTicket. Cette procédure implique les éléments suivants :

- Le module de chargement FEJMON, situé par défaut dans la bibliothèque de chargement FEK.SFEKAUTH, doit disposer d'une autorisation APF.



- RSE et le moniteur de travaux JES doivent utiliser le même ID d'application (APPLID). Par défaut, les deux serveurs utilisent FEKAPPL comme APPLID mais cette valeur peut être modifiée via la directive APPLID dans `rsed.envvars` pour RSE et FEJCNFG pour les moniteurs de travaux JES.

**Remarque :** Les anciens clients (version 7.0 et plus ancienne) communiquent directement avec le moniteur de travaux JES. Pour ces connexions, seule la méthode d'authentification par ID utilisateur et mot de passe est prise en charge.

---

## Sécurité des connexions

Différents niveaux de sécurité des communications sont pris en charge par RSE, qui contrôle toutes les communications entre le client et les services Developer for System z :

- Les communications (client-hôte) externes peuvent être limitées à des ports spécifiques. Cette fonction est désactivée par défaut.
- Les communications (client-hôte) externes peuvent être chiffrées à l'aide de la couche SSL. Cette fonction est désactivée par défaut.
- La vérification du port d'entrée peut être utilisée afin d'autoriser l'accès hôte uniquement aux adresses TCP/IP sécurisées. Cette fonction est désactivée par défaut.

### Limite des communications externes à des ports spécifiques

Le programmeur système peut spécifier les ports sur lesquels le serveur RSE peut communiquer avec le client. Par défaut, n'importe quel port disponible peut être utilisé. Cette gamme de ports n'a aucune connexion avec le port du démon RSE.

Afin de mieux comprendre l'utilisation des ports, une brève description du processus de connexion RSE est incluse ci-après :

1. Le client se connecte au port hôte 4035 du démon RSE.
2. Le démon RSE crée une unité d'exécution de serveur RSE.
3. Le serveur RSE ouvre un port hôte pour que le client se connecte. Le choix de ce port peut être configuré par l'utilisateur, soit au niveau du client dans l'onglet des propriétés du sous-système (méthode non recommandée) soit par l'intermédiaire de la définition `_RSE_PORTRANGE` du fichier `rsed.envvars`.
4. Le démon RSE renvoie le numéro de port au client.
5. Le client se connecte au port hôte.

**Remarque :** Le processus est identique pour la méthode de connexion alternative (facultative) utilisant REXEC/SSH, qui est décrite à la section «(Facultatif) Utilisation de REXEC (ou SSH)», à la page 104.

### Chiffrement des communications à l'aide de SSL

Tous les flux de données Developer for System z externes qui transitent par RSE peuvent être chiffrés à l'aide de SSL (Secure Socket Layer). L'utilisation de la couche SSL est contrôlée par les paramètres du fichier de configuration `ssl.properties`, comme décrit dans la section «Communication chiffrée à l'aide du protocole SSL», à la page 172.

L'émulateur de connexion à l'hôte sur le client se connecte à un serveur TN3270 sur l'hôte. L'utilisation de SSL est contrôlée par TN3270, comme indiqué dans le document *Communications Server IP Configuration Guide* (SC31-8775).

Le client du gestionnaire de déploiement d'application utilise le service Web TS CICS ou l'interface RESTful pour appeler les services hôte du gestionnaire de déploiement d'application. L'utilisation de SSL est contrôlée par TS CICS (voir *RACF Security Guide for CICS TS*).

## Vérification du port d'entrée

Developer for System z prend en charge la vérification du port d'entrée, ce qui permet à l'hôte d'accéder uniquement aux adresses TCP/IP sécurisées. L'utilisation du port d'entrée est contrôlée par la définition des profils spécifiques dans votre logiciel de sécurité et la directive `enable.port.of.entry` dans `rsed.envvars` (voir section «Vérification du port d'entrée (POE)», à la page 176).

Notez que l'activation du port d'entrée a une incidence sur d'autres applications TCP/IP prenant en charge la vérification du port d'entrée, telles que INETD.

## Ports TCP/IP

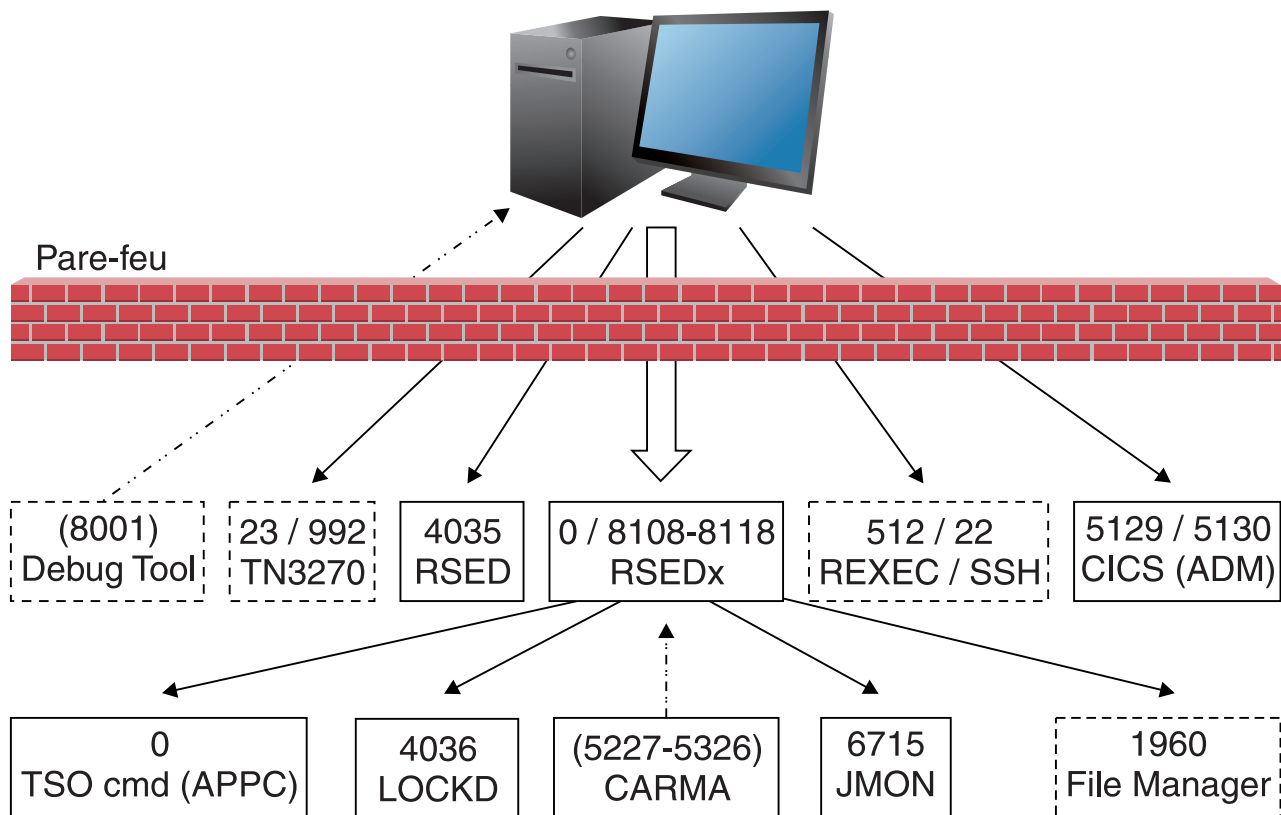


Figure 40. Ports TCP/IP

La figure 40 illustre les ports TCP/IP que Developer for System z peut utiliser. Les flèches indiquent la partie qui assure la liaison (tête de flèche) et celle qui assure la connexion.

## Communications externes

Définissez les ports suivants sur le pare-feu qui protège l'hôte z/OS car ils sont utilisés pour les communications client-hôte (via le protocole tcp) :

- Démon RSE pour la configuration des communications client-hôte, port 4035 par défaut. La communication sur ce port peut être chiffrée à l'aide de la fonction SSL.
- Serveur RSE pour la communication client-hôte. Par défaut, tout port disponible est utilisé, mais une plage de ports peut être définie à l'aide de la définition `_RSE_PORTRANGE` dans le fichier `rsed.envvars`. La communication sur ce port peut être chiffrée à l'aide de la fonction SSL.
- (facultatif) Service INETD pour les actions à distance (basées sur l'hôte) dans sous-projets z/OS UNIX :
  - REXEC (version z/OS UNIX), port par défaut 512.
  - SSH (version z/OS UNIX), port par défaut 22. La communication sur ce port est chiffrée à l'aide de la fonction SSL.
- (facultatif) Service Telnet TN3270 pour l'émulateur de connexion à l'hôte, port 23 par défaut. La communication peut être chiffrée à l'aide de la fonction SSL (port 992 par défaut). Le port par défaut affecté au service Telnet TN3270 dépend de l'utilisation ou non du chiffrement par l'utilisateur.
- (facultatif) Interfaces d'application CICSTS pour le gestionnaire de déploiement d'application :
  - Interface RESTful, port par défaut 5130.
  - Interface de services Web, port par défaut 5129. La communication sur ce port peut être chiffrée à l'aide de la fonction SSL.

**Remarque :**

- Les clients précédents (version 7.0 et précédentes) communiquent directement avec le moniteur de travaux JES, port par défaut 6715.
- Lors d'une session de débogage à distance pour Cobol, PL/I ou Assembler, IBM Debug Tool for z/OS est appelé. Ce produit communique directement avec le client. Cette communication est initiée sur l'hôte, et se connecte au port 8001 du client.

## Communication interne

Plusieurs services hôte Developer for System z s'exécutent dans des unités d'exécution ou espaces adresse séparés et utilisent des sockets TCP/IP comme mécanisme de communication. Tous ces services utilisent RSE pour communiquer avec le client et limitent leur flux de données à l'hôte. Pour certains services, n'importe quel port est utilisé. Pour d'autres, le programmeur système peut sélectionner un port ou une plage de ports à utiliser :

- Moniteur de travaux JES pour les services associés à JES, port par défaut 6715. Le port peut être défini dans le membre de configuration `FEJJCNFG`.
- Démon Lock pour des services liés au verrouillage des fichiers, port par défaut 4036. Vous pouvez définir le port dans le membre de configuration `rsed.envvars`.
- (Facultatif) File Manager Integration pour l'interaction avec IBM File Manager, port par défaut port 1960.
- (Facultatif) Communication CARMA, plage de ports 5227-5326 (100 ports) par défaut. La plage de ports peut être définie dans le fichier de configuration `CRASRV.properties`.
- (Facultatif) La version APPC du service Commandes TSO utilise n'importe quel socket disponible pour communiquer avec le gestionnaire de verrouillage (qui place en file d'attente des fichiers MVS pour les clients). Vous ne pouvez pas définir une plage de ports spécifiques à utiliser.

**Remarque :** Les anciens clients (version 7.0 et antérieure) communiquent directement avec le serveur du moniteur de travaux JES, port 6715 par défaut.

## CARMA et ports TCP/IP

Dans la plupart des cas, comme pour le démon RSE, un serveur assure la liaison à un port et écoute les demandes de connexion. Toutefois, CARMA utilise une démarche différente, étant donné que le serveur CARMA n'est pas encore actif lorsque le client lance la demande de connexion.

Lorsque le client envoie une demande de connexion, l'exploitant CARMA, qui est actif comme une unité d'exécution utilisateur d'un pool d'unités d'exécution RSE, trouve un port libre dans la plage indiquée dans le fichier de configuration CRASRV.properties et procède à la liaison. L'exploitant démarre le serveur CARMA et transmet le numéro de port, de sorte que le serveur sache à quel port se connecter. Une fois le serveur connecté, le client peut envoyer les demandes au serveur et recevoir les résultats.

Ainsi, dans une perspective TCP/IP, RSE est le serveur assurant la connexion au port (par l'intermédiaire de l'exploitant CARMA), le serveur CARMA étant le client qui s'y connecte.

---

## Utilisation de PassTickets

Après la connexion, des mots de passe PassTicket sont utilisés pour établir la sécurité des unités d'exécution sur le serveur RSE. Cette fonction ne peut pas être désactivée. Les PassTickets sont des mots de passe générés par le système pour une durée d'environ 10 minutes. Les mots de passe PassTicket générés s'appuient sur l'algorithme de chiffrement DES, l'ID utilisateur, l'ID d'application, un horodatage (heure/date) et une clé confidentielle. Cette clé confidentielle est un nombre de 64 bits (16 caractères hexadécimaux) qui doit être définie pour votre logiciel de sécurité.

Afin de mieux comprendre l'utilisation de PassTicket, une brève description du processus de sécurité RSE est incluse ci-après :

1. Le client se connecte au port hôte 4035 du démon RSE.
2. Le démon RSE authentifie le client en utilisant les données d'identification présentées par le client.
3. Le démon RSE crée un ID client unique et une unité d'exécution de serveur RSE.
4. Le serveur RSE génère un PassTicket et crée un environnement de sécurité pour le client, en utilisant le PassTicket comme mot de passe.
5. Le client se connecte au port hôte renvoyé par le démon RSE.
6. Le serveur RSE valide le client à l'aide de l'ID client.
7. Le serveur RSE utilise un PassTicket nouvellement généré comme mot de passe pour toutes les actions futures qui requièrent un mot de passe.

Le mot de passe réel du client n'est plus nécessaire après l'authentification initiale car les produits de sécurité conformes à SAF peuvent évaluer à la fois les mots de passe Passticket et les mots de passe standard. Le serveur RSE génère et utilise un mot de passe PassTicket chaque fois qu'un mot de passe est requis ; un mot de passe valide (temporaire) est ainsi disponible pour le client.

L'utilisation de mots de passe PassTicket permet à RSE de configurer un environnement de sécurité propre à l'utilisateur sans avoir à stocker tous les ID et les mots de passe dans une table, qui pourrait être illégalement consultée. Ils permettent également de mettre en oeuvre des méthodes d'authentification client qui n'utilisent pas de mots de passe réutilisables, tels que des certificats X.509.

Les profils de sécurité des classes APPL et PTKTDATA sont nécessaires pour permettre l'utilisation de mots de passe PassTicket. Ces profils sont propres à l'application et n'ont pas d'incidence sur la configuration système actuelle.

Comme les mots de passe PassTicket sont propres à l'application, RSE et le moniteur de travaux JES doivent utiliser le même ID d'application (APPLID). Par défaut, les deux serveurs utilisent FEKAPPL comme APPLID mais cette valeur peut être modifiée via la directive APPLID dans `rsed.envvars` pour RSE et `FEJJCNGF` pour le moniteur de travaux JES.

Vous ne devez pas utiliser OMVSAPPL comme ID d'application, car il ouvrira la clé confidentielle de la plupart des applications z/OS UNIX. De la même manière, vous ne devez pas utiliser l'ID d'application par défaut MVS, qui est MVS suivi par l'ID SMF du système, car il ouvrira la clé confidentielle de la plupart des applications MVS (y compris les travaux par lots des utilisateurs).

**Avertissement :** La demande de connexion du client n'aboutit pas si les mots de passe PassTickets ne sont pas correctement configurés.

---

## Consignation dans le journal d'audit

Developer for System z prend en charge la consignation dans le journal d'audit des actions gérées par le démon RSE. Les journaux d'audit sont conservés sous la forme de fichiers texte dans le répertoire de journalisation du démon, au format CSV.

### Contrôle d'audit

Plusieurs options de `rsed.envvars` influencent la fonction d'audit, comme indiqué à la section «Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JVAOPTS`», à la page 42.

- La fonction d'audit est activée/désactivée à l'aide de l'option `enable.audit.log`.
- Les valeurs d'audit par défaut sont contrôlées par les options `audit.*`.
- L'emplacement des journaux d'audit est contrôlé par l'option `daemon.log`.
- La page de codes utilisée pour rédiger le journal d'audit est contrôlée par la directive `_RSE_HOST_CODEPAGE` (voir «`rsed.envvars` - Fichier de configuration RSE», à la page 32).

La commande de l'opérateur **modify switch** peut être utilisée pour basculer manuellement vers un nouveau fichier journal d'audit, comme expliqué au Chapitre 8, «Commandes de l'opérateur», à la page 127.

Un message d'avertissement est envoyé à la console lorsque l'espace disponible dans le système de fichiers qui contient les fichiers journaux d'audit est insuffisant. Le message de console (FEK103E) s'affiche régulièrement tant que l'incident lié au manque d'espace n'a pas été résolu. Pour obtenir la liste des messages de la console générés par RSE, voir «Messages de la console», à la page 134.

## Données d'audit

Un nouveau fichier journal d'audit est démarré après un délai défini ou lorsque la commande de l'opérateur **modify switch** est exécutée. L'ancien fichier journal d'audit est enregistré sous `audit.log.yyyymmdd.hhmmss`, où `yyymmdd.hhmmss` représente la date/l'horodatage de fermeture de ce journal. La date/l'horodatage système attribué(e) au fichier indique la création du fichier journal. La combinaison des deux dates indique la période couverte par ce fichier journal d'audit.

Les actions suivantes sont consignées :

- Accès au système (connexion, déconnexion)
- Accès au spoule JES (soumission, affichage, mise en attente, publication, annulation et purge)
- Accès au fichier (lecture, écriture, création, suppression, modification de nom, compression, migration, rappel)
- Exécution de commandes TSO

Chaque action consignée est conservée (avec une date/un horodatage) au format CSV qui peut être lu par un outil d'automatisation ou d'analyse de données.

Les journaux d'audit dispose du masque de contrôle des données d'autorisation 640 (-rw-r-----), ce qui signifie que le propriétaire (ID utilisateur z/OS UNIX du démon RSE) dispose de droits en lecture et en écriture et que le groupe (par défaut) du propriétaire dispose d'un accès en lecture. Toutes les autres tentatives d'accès sont refusées, sauf si elles sont effectuées par un superutilisateur (UID 0) ou par un utilisateur disposant des droits d'accès suffisants au profil SUPERUSER.FILESYS dans la classe UNIXPRIV.

---

## Sécurité JES

Developer for System z permet aux clients d'accéder au spoule JES via le moniteur de travaux JES. Le serveur fournit un accès de base limité qui peut être étendu à l'aide des fonctions de protection du fichier spoule standard de votre produit de sécurité. Des actions opérateur (Mettre en attente, Publier, Annuler et Purger) sont effectuées sur les fichiers spoule via la console EMCS ; elles nécessitent des autorisations conditionnelles.

### Actions sur les travaux - Limitations sur les cibles

Le moniteur de travaux ne fournit pas aux utilisateurs de Developer for System z un accès opérateur intégral au spoule JES. Seules les commandes Mettre en attente, Publier, Annuler et Purger sont disponibles, et par défaut, uniquement pour les fichiers spoule dont l'utilisateur est le propriétaire. Les commandes sont exécutées par la sélection de l'option appropriée dans la structure de menu du client (il n'y a pas d'invite de commande). La portée des commandes peut être réduite à l'aide des profils de sécurité afin de définir les travaux pour lesquels les commandes sont disponibles.

Comparable à l'action SDSF **SJ** SDSF, le moniteur de travaux JES prend en charge la commande Afficher JCL pour extraire le code JCL qui a créé la sortie de travaux sélectionnée et l'afficher dans une fenêtre d'éditeur. Le moniteur de travaux JES extrait le code JCL de JES ce qui est utile dans les situations où le membre JCL d'origine n'est pas facilement localisé.

Tableau 21. Commandes de la console du moniteur de travaux JES

Action	JES2	JES3
Mettre en attente	\$Hx(jobid) avec x = {J, S ou T}	*F,J=jobid,H
Libérer	\$Ax(jobid) avec x = {J, S ou T}	*F,J=jobid,R
Annuler	\$Cx(jobid) avec x = {J, S ou T}	*F,J=jobid,C
Purger	\$Cx(jobid),P avec x = {J, S ou T}	*F,J=jobid,C
Afficher JCL	non applicable	non applicable

Les commandes JES disponibles répertoriées dans le tableau 21 sont, par défaut, limitées aux travaux dont l'utilisateur est le propriétaire. Cette limitation peut être modifiée avec la directive `LIMIT_COMMANDS`, comme spécifié dans «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27.

Tableau 22. Matrice des droits d'accès des commandes `LIMIT_COMMANDS`

	Propriétaire du travail	
<code>LIMIT_COMMANDS</code>	Utilisateur	Autre
<code>USERID</code> (valeur par défaut)	Autorisé	Non autorisé
<code>LIMITED</code>	Autorisé	Autorisé uniquement si permis de manière explicite par les profils de sécurité
<code>NOLIMIT</code>	Autorisé	Autorisé si les profils de sécurité l'acceptent ou lorsque la classe JESSPOOL n'est pas active

JES utilise la classe JESSPOOL pour protéger les fichiers SYSIN/SYSOUT. Comme SDSF, le moniteur de travaux JES étend l'utilisation de la classe JESSPOOL pour protéger également les ressources des travaux.

Si `LIMIT_COMMANDS` n'est pas `USERID`, le moniteur de travaux JES demandera le droit d'accès au profil associé dans de la classe JESSPOOL, comme indiqué dans le tableau suivant :

Tableau 23. Profils JESSPOOL étendus

Commande	Profil JESSPOOL	Droit d'accès requis
Mettre en attente	nodeid.userid.jobname.jobid	ALTER
Libérer	nodeid.userid.jobname.jobid	ALTER
Annuler	nodeid.userid.jobname.jobid	ALTER
Purger	nodeid.userid.jobname.jobid	ALTER
Afficher JCL	nodeid.userid.jobname.jobid.JCL	READ

Utilisez les substitutions suivantes dans le tableau précédent :



nodeid	ID du noeud NJE du sous-système JES cible
userid	ID utilisateur local du propriétaire du travail
jobname	Nom du travail
jobid	ID du travail JES

Si la classe JESSPOOL n'est pas active, les valeurs LIMITED et NOLIMIT de LIMIT\_COMMANDS auront un comportement différent, comme expliqué dans le tableau 9, à la page 30. Le comportement est identique lorsque la classe JESSPOOL est active, car, par défaut, elle refuse le droit d'accès si un profil n'est pas défini.

## Actions sur les travaux - Limitations liées à l'exécution

Après la définition des cibles autorisées, la seconde phase de la sécurité des commandes du spoule JES comprend la définition des autorisations nécessaires pour exécuter la commande de l'opérateur. Ce droit d'exécution est appliqué par les contrôles de sécurité z/OS et JES.

Notez que la commande Afficher JCL n'est pas une commande de l'opérateur comme une autre (par exemple, Mettre en attente, Libérer, Annuler et Purger). En conséquence, les limitations ci-dessous ne s'appliquent pas car il n'y a pas d'autre contrôle de sécurité.

Le moniteur de travaux JES émet toutes les commandes opérateur JES demandées par un utilisateur via une console EMCS dont le nom est contrôlé avec la directive CONSOLE\_NAME, comme indiqué à la section «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27.

Cette configuration permet à l'administrateur de sécurité de définir des droits d'exécution des commandes complexes en utilisant les classes OPERCMDS et CONSOLE.

- Pour utiliser une console EMCS, un utilisateur doit disposer au moins de droits READ dans le profil MVS.MCSOPER.nom-console de la classe OPERCMDS. Si aucun profil n'est défini, le système accorde la demande de droit.
- Pour exécuter une commande de l'opérateur JES, un utilisateur doit disposer des droits suffisants dans le profil JES%.\*\* (ou dans un profil plus restrictif) de la classe OPERCMDS. Si aucun profil n'est défini ou que la classe OPERCMDS n'est pas active, JES ne parvient pas à exécuter la commande.
- L'administrateur de sécurité peut également demander qu'un utilisateur fasse appel au moniteur de travaux JES lors de l'exécution de la commande de l'opérateur en indiquant WHEN(CONSOLE(JMON)) dans la définition PERMIT. La classe CONSOLE doit être active pour permettre le bon fonctionnement de cette configuration. Notez que la classe CONSOLE active est suffisante. Aucun profil n'est contrôlé pour les consoles EMCS.

Supposons que l'accès à l'identité du serveur du moniteur de travaux JES lors de la création d'une console JMON à partir d'une session TSO est empêché par votre logiciel de sécurité. Même si la console peut être créée, le point d'entrée est différent (moniteur de travaux JES/TSO). Les commandes JES exécutées par cette console échouent lors du contrôle de sécurité si la sécurité est configurée comme indiqué dans cette publication et que l'utilisateur ne dispose pas de droits d'accès aux commandes JES via d'autres procédures.

Notez que le moniteur de travaux JES ne peut pas créer la console JMON lorsqu'une commande doit être exécutée si le nom de la console est déjà utilisé. Pour éviter cela, le programmeur système peut définir la directive



GEN\_CONSOLE\_NAME=ON dans le fichier de configuration du moniteur de travaux JES ou l'administrateur de sécurité peut définir des profils de sécurité pour empêcher les utilisateurs TSO de créer une console. Les exemples de commandes RACF suivants empêchent tous les utilisateurs (sauf ceux qui sont autorisés) de créer une console TSO ou SDSF :

- 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)

**Remarque :** Sans autorisation pour ces commandes opérateur, les utilisateurs peuvent toujours soumettre des travaux et lire les sorties de travaux via le moniteur de travaux JES s'ils disposent de droits d'accès suffisants à des profils qui protègent ces ressources (comme celles des classes JESINPUT, JESJOBS et JESSPOOL).

Pour plus d'informations sur la protection des commandes opérateur, voir *Security Server RACF Security Administrator's Guide* (SA22-7683).

## Accès aux fichiers spoule

Le moniteur de travaux JES permet, par défaut, de parcourir tous les fichiers spoule. Cette autorisation par défaut peut être modifiée avec la directive LIMIT\_VIEW comme spécifié dans «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27.

Tableau 24. Matrice des droits d'accès de consultation LIMIT\_VIEW

LIMIT_VIEW	Propriétaire du travail	
	Utilisateur	Autre
USERID	Autorisé	Non autorisé
NOLIMIT (valeur par défaut)	Autorisé	Autorisé si les profils de sécurité l'acceptent ou lorsque la classe JESSPOOL n'est pas active

Pour limiter l'accès des utilisateurs à leurs propres travaux dans le spoule JES, définissez l'instruction "LIMIT\_VIEW=USERID" dans le fichier de configuration du moniteur de travaux JES, FEJJCNFG. Si les utilisateurs souhaitent accéder à davantage de travaux, mais pas à tous, utilisez les fonctions de protection du fichier spoule standard de votre produit de sécurité (la classe JESSPOOL, par exemple).

Quand vous définissez d'autres protections, notez que le moniteur de travaux fait appel à l'interface SAPI (SYSOUT application program interface) pour accéder au spoule. En conséquence l'utilisateur a besoin, au minimum, de droits d'accès de mise à jour (UPDATE) des fichiers spoule, même pour des fonctionnalités de navigation. Cette exigence ne s'applique pas si vous utilisez z/OS version 1.7 (z/OS 1.8 pour JES3) ou une version ultérieure. Dans ce cas, les droits d'accès en lecture (READ) suffisent pour les fonctionnalités de navigation.

Pour plus d'informations sur la protection du fichier spoule JES, voir *Security Server RACF Security Administrator's Guide* (SA22-7683).

## Communication chiffrée à l'aide du protocole SSL

Les communications externes (client-hôte) peuvent être chiffrées à l'aide de SSL (Secure Socket Layer). Cette fonction est désactivée par défaut et est contrôlée par les paramètres du fichier `ssl.properties` (voir la section «(Facultatif) Chiffrement SSL RSE», à la page 95).

Le démon RSE et le serveur RSE prennent en charge des mécanismes différents pour stocker des certificats en raison leurs différences architecturales. Cela signifie que des définitions et des certificats SSL sont nécessaires pour le démon et le serveur RSE. Un certificat partagé peut être utilisé si le démon et le serveur RSE utilisent la même méthode de gestion des certificats.

*Tableau 25. Mécanismes de stockage des certificats SSL*

Stockage des certificats	Créé et géré par	Démon RSE	Serveur RSE
Fichier de clés	Produit de sécurité compatible avec SAF	pris en charge	pris en charge
Base de données de clés	gskkyman de z/OS UNIX	pris en charge	/
Magasin de clés	Outil de clé de Java	/	pris en charge

**Remarque :** Il est conseillé d'utiliser des fichiers de clés conformes à SAF pour la gestion des certificats.

Les fichiers de clés conformes à SAF peuvent stocker la clé privée du certificat dans la base de données de sécurité ou en utilisant ICSF (Integrated Cryptographic Service Facility), l'interface vers le matériel de chiffrement de System z.

Il est recommandé d'utiliser ICSF pour le stockage des clés privées associées à des certificats numériques. En effet, il s'agit d'une solution plus sûre que la gestion de clé privée non ICSF. ICSF assure le chiffrement des clés privées dans la clé maîtresse ICSF, leur accès étant contrôlé par les ressources générales dans les classes de sécurité CSFKEYS et CSFSERV. De plus, les performances opérationnelles sont améliorées, car ICSF utilise un processeur cryptographique .

Le démon RSE utilise les fonctions SSL système pour gérer des communications chiffrées SSL. Cela signifie que `SYS1.SIEALNKE` doit être contrôlé par programme via le logiciel de sécurité et être à la disposition de RSE via `LINKLIST` ou la directive `STEPLIB` dans `rsed.envvars`.

L'ID utilisateur RSE (`STCRSE` dans l'exemple de commande ci-dessous) doit disposer des droits nécessaires pour accéder à son fichier de clés et aux certificats associés lorsque des fichiers de clés conformes à SAF sont utilisés pour le démon ou le serveur RSE.

- `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)`
- `SETRPTS RACLIST(FACILITY) REFRESH`

Pour plus d'informations sur l'activation de SSL pour Developer for System z, voir Annexe A, «Configuration de l'authentification SSL et X.509», à la page 299.

---

## Authentification du client à l'aide de certificats X.509

Le démon RSE prend en charge les utilisateurs qui s'authentifient eux-mêmes à l'aide d'un certificat X.509. L'utilisation de communications chiffrées SSL est indispensable pour cette fonction car il s'agit d'une extension de l'authentification hôte avec un certificat utilisé dans SSL.

Le démon RSE lance la procédure d'authentification client en validant le certificat client. Les principaux éléments vérifiés sont les dates de validité du certificat et le niveau de confiance de l'autorité de certification utilisée pour signer le certificat. Une liste de révocation de certificat (CRL) d'un tiers peut également être consultée.

Une fois que le démon RSE valide le certificat, celui-ci est traité pour l'authentification. Le certificat est transmis au produit de sécurité à des fins d'authentification, sauf si la directive `enable.certificate.mapping` de `rsed.envvars` correspond à `false`. Dans ce cas, le démon RSE effectue l'authentification.

Si elle aboutit, la procédure d'authentification détermine l'ID utilisateur à utiliser pour cette session et le soumet au test du démon RSE pour vérifier qu'il est utilisable sur le système hôte où le démon RSE s'exécute.

La dernière vérification (réalisée pour chaque mécanisme d'authentification, et pas simplement pour les certificats X.509) s'assure que l'ID utilisateur est autorisé à utiliser Developer for System z.

Si vous êtes familier des classifications de sécurité SSL utilisées par TCP/IP, la combinaison de ces procédures de validation correspond aux "spécification de niveau 3 du client" (la plus élevée).

## Validation de l'autorité de certification (CA)

Une partie de la procédure de validation du certificat consiste à vérifier que le certificat a été signé par une autorité de certification habilitée. Pour effectuer cette opération, le démon RSE doit avoir accès à un certificat qui identifie l'autorité de certification.

Si vous utilisez la base de données de clés **gskkyman** pour la connexion SSL, le certificat de l'autorité de certification doit être ajouté à la base de données de clés.

Si vous utilisez un fichier de clés SAF (méthode recommandée), vous devez ajouter le certificat de l'autorité de certification à votre base de données de sécurité sous la forme d'un certificat CERTAUTH associé à l'attribut TRUST ou HIGHTRUST, comme indiqué dans l'exemple de commande RACF ci-après :

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

La plupart des produits de sécurité possèdent des certificats d'autorité de certification reconnues dans leurs bases de données avec un état NOTRUST. Utilisez les exemples de commande RACF suivantes pour répertorier les certificats des autorités de certification et marquer un certificat comme sécurisé (trusted) en fonction du libellé qui lui est affecté.

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

**Remarque :** L'état HIGHTRUST est indispensable si vous vous appuyez sur RACF pour authentifier l'utilisateur en fonction de l'extension HostIdMappings dans le certificat. Pour plus d'informations, voir «Authentification par votre logiciel de sécurité».

Une fois que le certificat de l'autorité de certification est ajouté à la base de données de sécurité, il doit être connecté au fichier de clés RSE, comme indiqué dans l'exemple de commande RACF ci-après :

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

Pour obtenir des informations détaillées sur la commande **RACDCERT**, voir le document *Security Server RACF Command Language Reference* (SA22-7687).

**Attention :** Si vous faites appel au démon RSE au lieu du logiciel de sécurité pour authentifier un utilisateur, veillez à ne pas mélanger les autorités de certification avec un état TRUST et HIGHTRUST dans le fichier de clés SAF ou une base de données **gskkyman**. Le démon RSE n'est pas en mesure d'établir une distinction entre les deux. Les certificats signés par une autorité de certification avec l'état TRUST est valide pour une authentification de l'ID utilisateur.

## (Facultatif) Interrogation d'une liste de révocation de certificat (CRL)

Si vous le souhaitez, vous pouvez demander au démon RSE de vérifier une ou plusieurs listes de révocation de certificat (CRL) pour renforcer la sécurité de la procédure de validation. Cette opération est effectuée en ajoutant des variables d'environnement liées aux listes de révocation de certificat au fichier `rsed.envvars`. Pour plus d'informations sur les exemples de variable suivants, voir «rsed.envvars - Fichier de configuration RSE», à la page 32 :

- `GSK_CRL_SECURITY_LEVEL`
- `GSK_LDAP_SERVER`
- `GSK_LDAP_PORT`
- `GSK_LDAP_USER`
- `GSK_LDAP_PASSWORD`

Pour plus d'informations sur ces variables d'environnement et sur d'autres variables d'environnement utilisées par z/OS System SSL, voir *Cryptographic Services System Secure Sockets Layer Programming* (SC24-5901).

**Remarque :** Soyez prudent lorsque vous définissez d'autres variables d'environnement z/OS System SSL (`GSK_*`) dans le fichier `rsed.envvars` car elles risquent de modifier la manière dont le démon RSE traite les connexions SSL et l'authentification par certificat.

## Authentification par votre logiciel de sécurité

RACF effectue plusieurs vérifications pour authentifier un certificat et renvoyer l'ID utilisateur associé. Notez toutefois que d'autres produits de sécurité peuvent effectuer cette opération différemment. Pour plus d'informations sur la fonction `initACEE` utilisée pour effectuer l'authentification (mode requête), reportez-vous à la documentation du produit de sécurité.

1. RACF vérifie si le certificat est défini dans la classe DIGTCERT. Si c'est le cas, RACF renvoie l'ID utilisateur associé à ce certificat lorsque celui-ci a été ajouté à la base de données RACF.

Les certificats sont définis dans RACF à l'aide de la commande RACDCERT, comme indiqué dans l'exemple suivant :

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

2. Si le certificat n'est pas défini, RACF vérifie s'il n'y a pas de filtre de nom de certificat concordant défini dans les classes DIGTNMAP ou DIGTCRIT. Si tel est le cas, il renvoie l'ID utilisateur associé au filtre dont la concordance est la plus proche.

**Remarque :** Il est déconseillé d'utiliser des filtres de nom pour les certificats utilisés par Developer for System z, car ces filtres rattachent tous les certificats à un même ID utilisateur. Cela signifie que tous les utilisateurs Developer for System z se connectent avec le même ID utilisateur.

3. En l'absence de filtre de nom concordant, RACF recherche l'extension de certificat HostIdMappings et extrait la paire ID utilisateur et nom d'hôte imbriquée. Si l'extension est détectée et validée, RACF renvoie l'ID utilisateur défini au sein de l'extension HostIdMappings.

La paire ID utilisateur et nom d'hôte est valide si toutes ces conditions sont remplies :

- Le certificat de l'autorité de certification utilisé pour signer ce certificat est marqué comme HIGHTRUST dans la classe DIGTCERT.
- La longueur de l'ID utilisateur stocké dans l'extension est valide (de 1 à 8 caractères).
- L'ID utilisateur affecté au démon RSE dispose (au minimum) de droits READ dans le profil IRR.HOST.nomhôte de la classe SERVAUTH, où nomhôte est le nom d'hôte stocké dans l'extension. Il s'agit généralement d'un nom de domaine, par exemple CDFMVS08.RALEIGH.IBM.COM.

La définition de l'extension HostIdMappings dans la syntaxe ASN.1 est :

```
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
}
```

**Remarque :** L'extension HostIdMappings n'est pas prise en compte si l'ID utilisateur cible a été créé après la date de début de validité du certificat contenant l'extension HostIdMappings. Si vous créez des ID utilisateur spécialement pour des certificats avec des extensions HostIdMappings, vérifiez qu'ils sont créés avant la soumission des demandes de certificat.

Pour plus d'informations sur les certificats X.509, leur mode de gestion par RACF et les modalités de définition des filtres de nom de certificat, voir *Security Server RACF Security Administrator's Guide* (SA22-7683). Pour obtenir des informations détaillées sur la commande **RACDCERT**, voir le document *Security Server RACF Command Language Reference* (SA22-7687).

## Authentification effectuée par le démon RSE

Developer for System z peut effectuer une authentification de base des certificats X.509 sans faire appel à votre produit de sécurité. L'authentification effectuée par

le démon RSE requiert la définition d'un ID utilisateur et d'un nom d'hôte dans une extension de certificat et est activée uniquement si la directive `enable.certificate.mapping` définie dans le fichier `rsed.envvars` correspond à `FALSE`.

Cette fonction doit être utilisée si votre produit de sécurité ne prend pas en charge l'authentification d'un utilisateur via un certificat X.509 ou si un certificat échoue aux tests effectués par le produit de sécurité (par exemple, le certificat possède un identificateur erroné pour l'extension `HostIdMappings` ou il n'y a pas de filtre ou de définition de nom dans `DIGTCERT`).

Le client demande à l'utilisateur l'identificateur d'extension (OID) à utiliser. Par défaut, il s'agit de l'OID `HostIdMappings`, {1 3 18 0 2 18 1}.

Le démon RSE doit extraire l'ID utilisateur et le nom d'hôte en utilisant l'extension de format `HostIdMappings`. Ce format est décrit à la section «Authentification par votre logiciel de sécurité», à la page 174.

La paire ID utilisateur et nom d'hôte est valide si toutes ces conditions sont remplies :

- La longueur de l'ID utilisateur stocké dans l'extension est valide (de 1 à 8 caractères).
- L'ID utilisateur affecté au démon RSE dispose (au minimum) de droits `READ` dans le profil `IRR.HOST.nomhôte` de la classe `SERVAUTH`, où `nomhôte` est le nom d'hôte stocké dans l'extension. Il s'agit généralement d'un nom de domaine, par exemple `CDFMVS08.RALEIGH.IBM.COM`.

**Avertissement :** Il revient à l'administrateur de sécurité de vérifier que toutes les autorités de certification connues du démon RSE sont parfaitement dignes de confiance car le démon RSE ne peut pas vérifier si l'autorité qui a signé le certificat client est parfaitement digne de confiance (`highly trusted`) ou simplement digne de confiance (`trusted`). Pour plus d'informations sur les certificats d'autorités de certification accessibles, voir «Validation de l'autorité de certification (CA)», à la page 173.

---

## Vérification du port d'entrée (POE)

Developer for System z prend en charge la vérification du port d'entrée, ce qui permet à l'hôte d'accéder uniquement aux adresses TCP/IP sécurisées. Cette fonction est désactivée par défaut et requiert la définition du profil de sécurité `BPX.POE`, comme le montrent les exemples de commandes RACF :

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

### Remarque :

- RSE doit être configuré pour utiliser le port d'entrée. Pour cela, il est nécessaire de supprimer la mise en commentaire de l'option `"enable.port.of.entry=true"` dans `rsed.envvars`, comme indiqué à la section «Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JAVAOPTS`», à la page 42.
- L'ID utilisateur RSE `STCRSE` requiert `UID(0)` lorsque ce profil n'est pas défini et exige que la vérification du port d'entrée soit activée dans `rsed.envvars`.

- La définition de BPX.POE a un impact sur d'autres applications TC/PIP prenant en charge la vérification du port d'entrée (INETD, par exemple).
- Des zones de sécurité (profils EZB.NETACCESS.\*\*, qui sont des zones d'adresses IP) doivent être définies dans la classe SERVAUTH pour bénéficier pleinement de la vérification du port d'entrée.

Pour plus d'informations sur le contrôle d'accès au réseau via la vérification du port d'entrée, voir *Communications Server IP Configuration Guide* (SC31-8775).

---

## Sécurité CICSTS

Developer for System z permet, via le gestionnaire de déploiement d'application, aux administrateurs CICS de contrôler les définitions de ressource CICS qui peuvent être modifiées par le développeur, leurs valeurs par défaut ainsi que l'affichage d'une définition de ressource CICS à l'aide du serveur de définition de ressource CICS. Pour plus d'informations sur les définitions de sécurité TS CICS, voir Chapitre 15, «Remarques à propos de CICSTS», à la page 257.

## Référentiel de CRD

Le fichier VSAM du référentiel du serveur CRD contient toutes les définitions de ressource par défaut ; il doit par conséquent être protégé contre les mises à jour tout en autorisant les développeurs à consulter les valeurs qui y sont conservées.

## Transactions CICS

Developer for System z met à disposition plusieurs transactions qui sont utilisées par le serveur CRD lors de la définition et de la consultation des ressources CICS. Quand la transaction est rattachée, la vérification de la sécurité de la ressource CICS, si elle est activée, garantit que l'ID utilisateur est autorisé à exécuter l'ID de transaction.

## Communication chiffrée à l'aide du protocole SSL

Le client du gestionnaire de déploiement d'application utilise les services Web TS CICS ou l'interface RESTful pour appeler le serveur CRD. L'utilisation de SSL pour ces communications est contrôlée par la définition TCPIP SERVICE TS CICS, comme indiqué dans le document *RACF Security Guide for CICS TS*.

---

## Sécurité SCLM

SCLM Developer Toolkit offre des fonctionnalités de sécurité facultatives pour les fonctions de génération, de promotion et de déploiement.

Si l'administrateur SCLM a activé la sécurité pour une fonction, des appels SAF sont effectués afin de vérifier l'autorité qui exécute la fonction protégée avec l'ID de l'appelant ou d'un utilisateur de substitution.

Pour de plus amples informations sur les définitions de sécurité SCLM requises, voir le document *SCLM Developer Toolkit - Guide d'administration* (SC11-6464).



---

## Fichiers de configuration Developer for System z

Plusieurs fichiers de configuration Developer for System z contiennent des directives qui ont une incidence sur la configuration de la sécurité. En fonction des informations de ce chapitre, l'administrateur de sécurité et le programmeur système peuvent déterminer les paramètres à définir pour les directives ci-dessous.

### Moniteur de travaux JES - FEJJCNFG

- `LIMIT_COMMANDS={USERID | LIMITED | NOLIMIT }`  
Définit les travaux auxquels les actions peuvent être appliquées (à l'exception de la consultation et la soumission). Pour plus d'informations, voir «Actions sur les travaux - Limitations sur les cibles», à la page 168.
- `LIMIT_VIEW={USERID | NOLIMIT}`  
Définissez les fichiers de spoule qui peuvent être consultés. Pour plus d'informations, voir «Accès aux fichiers spoule», à la page 171.
- `APPLID={FEKAPPL | *}`  
ID application utilisé pour la création et la validation de mots de passe PassTicket. Pour plus d'informations, voir «Utilisation de PassTickets», à la page 166.

**Remarque :** Des informations sur ces directives et d'autres directives FEJJCNFG sont disponibles dans la section «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27.

### RSE - rsed.envvars

- `( _RSE_JAVAOPTS ) -DDENY_PASSWORD_SAVE={true | false}`  
Empêche les utilisateurs de sauvegarder leur mot de passe hôte sur le client. Pour plus d'informations, voir «Définition de paramètres de démarrage Java supplémentaires avec \_RSE\_JAVAOPTS», à la page 42.
- `( _RSE_JAVAOPTS ) -DDSTORE_IDLE_SHUTDOWN_TIMEOUT=value`  
Délai de déconnexion des clients inactifs. Pour plus d'informations, voir «Définition de paramètres de démarrage Java supplémentaires avec \_RSE\_JAVAOPTS», à la page 42.
- `( _RSE_JAVAOPTS ) -DAPPLID={FEKAPPL | *}`  
ID application utilisé pour la création et la validation de mots de passe PassTicket. Pour plus d'informations, voir «Utilisation de PassTickets», à la page 166.
- `( _RSE_JAVAOPTS ) -Denable.port.of.entry={true | false}`  
Active la vérification du port d'entrée. Pour plus d'informations, voir «Vérification du port d'entrée (POE)», à la page 176.
- `( _RSE_JAVAOPTS ) -Denable.certificate.mapping={true | false}`  
Utilisez le produit de sécurité pour authentifier des utilisateurs avec un certificat X.509. Pour plus d'informations, voir «Authentification du client à l'aide de certificats X.509», à la page 173.
- `( _RSE_JAVAOPTS ) -Ddaemon.log={/var/rdz/logs | *}`  
Emplacement des fichiers d'audit. Pour plus d'informations, voir «Consignation dans le journal d'audit», à la page 167.

**Remarque :** Des informations sur ces directives et d'autres directives rsed.envvars sont disponibles dans la section «rsed.envvars - Fichier de configuration RSE», à la page 32.



## RSE - ssl.properties

- `daemon_keydb_file`={nom du fichier de clés SAF | nom de la base de données de clés gskkyman}  
Emplacement du certificat du démon RSE. Pour plus d'informations, voir «Communication chiffrée à l'aide du protocole SSL», à la page 172.
- `daemon_key_label`=libellé du certificat  
Nom du certificat du démon RSE. Pour plus d'informations, voir «Communication chiffrée à l'aide du protocole SSL», à la page 172.
- `server_keystore_file`={nom du fichier de clés SAF | nom du magasin de clés Java}  
Emplacement du certificat du serveur RSE. Pour plus d'informations, voir «Communication chiffrée à l'aide du protocole SSL», à la page 172.
- `server_keystore_label`=certificate label  
Nom du certificat du serveur RSE. Pour plus d'informations, voir «Communication chiffrée à l'aide du protocole SSL», à la page 172.
- `server_keystore_type`={JKS | JCRACFKS | JCECCARACFKS}  
Type de fichier de clés utilisé (Java ou SAF). Pour plus d'informations, voir «Communication chiffrée à l'aide du protocole SSL», à la page 172.

**Remarque :** Des informations sur ces directives et d'autres directives `ssl.properties` sont disponibles dans la section «(Facultatif) Chiffrement SSL RSE», à la page 95.

---

## Définitions de sécurité

Personnalisez et soumettez l'exemple de membre FEKRACF, comportant les commandes RACF et z/OS UNIX permettant de créer les définitions de sécurité de base de Developer for System z.

FEKRACF se trouve dans `FEK.#CUST.JCL`, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Pour plus d'informations sur les commandes RACF, voir *RACF Command Language Reference* (SA22-7687).

### Remarque :

- Pour les sites qui utilisent CA ACF2™ for z/OS, veuillez consulter le lien <https://support.ca.com/irj/portal/kbtech?ipLogNrow=0&docid=492389&searchID=TEC492389> pour obtenir des détails sur les commandes de sécurité nécessaires à la configuration correcte de Developer for System z.
- Pour les sites qui utilisent CA Top Secret for z/OS, veuillez consulter la page de votre produit du site de support CA (<https://support.ca.com>), puis le document Developer for System z connexe. Ce document présente les commandes de sécurité nécessaires à la configuration correcte de Developer for System z.

Les sections ci-après décrivent les étapes requises, la configuration facultative et les alternatives possibles.

## Configuration requise et liste de contrôle

Pour procéder à la configuration de la sécurité, l'administrateur de la sécurité doit connaître les valeurs figurant dans le tableau 26. Ces valeurs ont été définies dans les étapes précédentes d'installation et de personnalisation de Developer for System z.

Tableau 26. Variables de configuration de la sécurité

Description	<ul style="list-style-type: none"> <li>• Valeur par défaut</li> <li>• Emplacement de la réponse</li> </ul>	Valeur
Qualifiant de haut niveau du produit Developer for System z	<ul style="list-style-type: none"> <li>• FEK</li> <li>• Installation SMP/E</li> </ul>	
Qualifiant de haut niveau de personnalisation de Developer for System z	<ul style="list-style-type: none"> <li>• FEK.#CUST</li> <li>• FEK.SFEKSAMP(FEKSETUP) (voir «Configuration personnalisée», à la page 15).</li> </ul>	
Nom de tâche démarrée du moniteur de travaux JES	<ul style="list-style-type: none"> <li>• JMON</li> <li>• FEK.#CUST.PROCLIB(JMON) (voir «Modifications de PROCLIB», à la page 21).</li> </ul>	
Nom de tâche démarrée du démon RSE	<ul style="list-style-type: none"> <li>• RSED</li> <li>• FEK.#CUST.PROCLIB(RSED) (voir «Modifications de PROCLIB», à la page 21).</li> </ul>	
Nom de tâche démarrée du démon Lock	<ul style="list-style-type: none"> <li>• LOCKD</li> <li>• FEK.#CUST.PROCLIB(LOCKD) (voir «Modifications de PROCLIB», à la page 21).</li> </ul>	
ID application	<ul style="list-style-type: none"> <li>• FEKAPPL</li> <li>• /etc/rdz/rsed.envvars, comme décrit dans «Définition de paramètres de démarrage Java supplémentaires avec _RSE_JAVAOPTS», à la page 42</li> </ul>	

La liste ci-après présente les actions à effectuer pour obtenir une configuration de sécurité de base de Developer for System z. Comme indiqué dans les sections ci-après, différentes méthodes peuvent répondre à vos exigences, en fonction du niveau de sécurité recherché. Pour plus d'informations sur la configuration de sécurité des services facultatifs de Developer for System z, reportez-vous aux sections précédentes.

- «Activation des paramètres et des classes de sécurité», à la page 181
- «Définition d'un segment OMVS pour les utilisateurs de Developer for System z», à la page 182
- «Définition des profils de fichier», à la page 182
- «Définition des tâches démarrées Developer for System z», à la page 185

- «Définition de la sécurité des commandes JES», à la page 186
- «Définition de RSE comme serveur z/OS UNIX sécurisé», à la page 188
- «Définition de bibliothèques contrôlées par programme MVS pour RSE», à la page 188
- «Définition de la protection d'application pour RSE», à la page 189
- «Définition du support de mots de passe PassTicket pour RSE», à la page 189
- «Définition de fichiers contrôlés par programme z/OS UNIX pour RSE», à la page 190
- «Vérification des paramètres de sécurité», à la page 191

## Activation des paramètres et des classes de sécurité

Developer for System z utilise différents mécanismes de sécurité pour fournir au client un environnement hôte sécurisé et contrôlé. Pour cela, plusieurs classes et paramètres de sécurité doivent être actifs, comme illustré par les exemples de commandes RACF ci-dessous :

- Affichage des paramètres courants
  - SETROPTS LIST
- Activation de la classe de fonction pour z/OS UNIX et les profils de certificats numériques
  - SETROPTS GENERIC(FACILITY)
  - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- Activation des définitions de tâche démarrée
  - SETROPTS GENERIC(STARTED)
  - RDEFINE STARTED \*\* STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
  - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- Activation de la sécurité de la console du moniteur de travaux JES
  - SETROPTS GENERIC(CONSOLE)
  - SETROPTS CLASSACT(CONSOLE) RACLIST(CONSOLE)
- Activation de la protection des commandes de l'opérateur du moniteur de travaux JES
  - SETROPTS GENERIC(OPERCMDS)
  - SETROPTS CLASSACT(OPERCMDS) RACLIST(OPERCMDS)
- Activation de la protection des applications pour RSE
  - SETROPTS GENERIC(APPL)
  - SETROPTS CLASSACT(APPL) RACLIST(APPL)
- Activation de l'ouverture de session sécurisée pour RSE à l'aide de mots de passe PassTicket
  - SETROPTS GENERIC(PTKTDATA)
  - SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
- Activation du contrôle de programme pour que seul le code sécurisé puisse être chargé par RSE
  - RDEFINE PROGRAM \*\* ADDMEM('SYS1.CMDLIB'//NOPADCHK) UACC(READ)
  - SETROPTS WHEN(PROGRAM)

**Remarque :** Ne créez pas le profil \*\* si le profil \* existe déjà dans la classe PROGRAM. Cela occulterait et compliquerait le chemin de recherche utilisé par votre logiciel de sécurité. Dans ce cas de figure, vous devez fusionner la définition \* existante et la nouvelle définition \*\*. IBM vous recommande d'utiliser le profil \*\*, comme indiqué dans le document *Security Server RACF Security Administrator's Guide* (SA22-7683).

**Attention :** Certains produits (FTP, par exemple) doivent être contrôlés par programme si "WHEN PROGRAM" est actif. Vous devez les essayer avant de les activer sur un système de production.

- (Facultatif) Activation du support X.509 HostIdMappings et du port d'entrée étendu
  - SETROPTS GENERIC(SERVAUTH)
  - SETROPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)

## Définition d'un segment OMVS pour les utilisateurs de Developer for System z

Un segment OMVS RACF (ou équivalent) indiquant un ID utilisateur z/OS UNIX différent de zéro valide, un répertoire principal et une commande shell doivent être définis pour chaque utilisateur de Developer for System z. Leur groupe par défaut requiert également un segment OMVS avec un ID groupe.

Dans l'exemple de commandes RACF ci-dessous, remplacez les marques de réservation #userid, #user-identifiant, #group-name et #group-identifiant par les valeurs réelles :

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

Bien que cela soit déconseillé, vous pouvez utiliser le segment OMVS défini dans le profil BPX.DEFAULT.USER de la classe FACILITY afin de répondre aux exigences de segment OMVS pour Developer for System z.

## Définition des profils de fichier

Un accès en lecture pour les utilisateurs et en modification pour les programmeurs système suffit pour la plupart des fichiers Developer for System z. Remplacez la marque de réservation #sysprog par des ID utilisateur ou des noms de groupes RACF. Demandez également au programmeur système qui a installé et configuré le produit de vous fournir les noms de fichier corrects. FEK est le qualificatif de haut niveau par défaut utilisé pendant l'installation, et FEK.#CUST celui relatif aux fichiers créés pendant le processus de personnalisation.

- 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

### Remarque :

- Il est vivement recommandé de protéger FEK.SFEKAUTH contre les mises à jour, car ce fichier est autorisé par APF. Il en va de même pour FEK.SFEKLOAD et FEK.SFEKLPA, mais ici la raison est que ces fichiers sont contrôlés par un programme.
- Les exemples de commande utilisés dans la présente publication et dans le travail FEKRACF supposent que l'EGN (Enhanced Generic Naming) est activé. Le qualificatif \*\* peut ainsi être utilisé pour représenter n'importe quel nombre de qualificatifs dans la classe DATASET. Remplacez \*\* par \* si l'EGN n'est pas activé dans votre système. Pour plus d'informations sur l'EGN, voir le document *Security Server RACF Security Administrator's Guide* (SA22-7683).

Certains des composants Developer for System z facultatifs requièrent des profils de fichier de sécurité supplémentaires. Remplacez les marques de réservation #sysprog, #ram-developer et #cicsadmin par des ID utilisateur ou par des noms de groupes RACF valides :

- Si la traduction des noms longs/abrégés de SCLM Developer Toolkit est utilisée, les utilisateurs doivent disposer d'un accès en mise à jour (UPDATE) au mappage VSAM, FEK.#CUST.LSTRANS.FILE.
  - ADDSD 'FEK.#CUST.LSTRANS.\*.\*' UACC(UPDATE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
  - PERMIT 'FEK.#CUST.LSTRANS.\*.\*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - SETROPTS GENERIC(DATASET) REFRESH
- Les développeurs CARMA RAM (Repository Access Manager) requièrent un accès UPDATE aux VSAM CARMA, FEK.#CUST.CRA\*.
  - ADDSD 'FEK.#CUST.CRA\*.\*' UACC(READ)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
  - PERMIT 'FEK.#CUST.CRA\*.\*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.#CUST.CRA\*.\*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
  - SETROPTS GENERIC(DATASET) REFRESH
- Si le serveur CRD du gestionnaire de déploiement d'application (définition de ressource CICS) est utilisé, l'administrateur CICS doit détenir un accès UPDATE à la méthode VSAM du référentiel CRD.
  - 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
- Si le référentiel de manifestes du gestionnaire de déploiement d'application est défini, tous les utilisateurs de CICS Transaction Server requièrent un accès en mise à jour (UPDATE) au VSAM du référentiel de manifestes.
  - 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

Utilisez les exemples de commande RACF suivants pour obtenir une configuration encore plus sécurisée dans laquelle l'accès READ est également contrôlé.

- Protection uacc(none) des fichiers
  - ADDGROUP (FEK)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - HLQ STUB')
  - OWNER(IBMUSER) SUPGROUP(SYS1)"
  - ADDSD 'FEK.\*.\*' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
  - ADDSD 'FEK.SFEKAUTH' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
  - ADDSD 'FEK.SFEKLOAD' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
  - ADDSD 'FEK.SFEKPROC' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
  - ADDSD 'FEK.#CUST.PARMLIB' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
  - ADDSD 'FEK.#CUST.CNTL' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
  - ADDSD 'FEK.#CUST.LSTRANS.\*.\*' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - SCLMDT')
  - ADDSD 'FEK.#CUST.CRA\*.\*' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - CARMA')
  - ADDSD 'FEK.#CUST.ADNREP\*.\*' UACC(NONE)
  - DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')

- ADDSD 'FEK.#CUST.ADNMAN\*.\*' UACC(NONE)  
DATA('RATIONAL DEVELOPER FOR SYSTEM Z - ADN')
- Autorisation de la gestion de toutes les bibliothèques par le programmeur système
  - PERMIT 'FEK.\*.\*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.SFEKAUTH CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.SFEKLOAD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.SFEKLOAD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.SFEKLOAD CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.SFEKPROC CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.#CUST.PARMLIB CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.#CUST.CNTL CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.#CUST.LSTRANS\*.\*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.#CUST.CRA\*.\*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.#CUST.ADNREP\*.\*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
  - PERMIT 'FEK.#CUST.ADNMAN\*.\*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
- Autorisation d'accès des clients aux bibliothèques de chargement et aux bibliothèques exec
  - 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(\*)

**Remarque :** Aucune autorisation n'est nécessaire pour FEK.SFEKLPA, car tout le code qui se trouve dans LPA est accessible à tous les utilisateurs.

- Autorisation d'accès du moniteur de travaux JES à la bibliothèque de chargement et de paramètres
  - PERMIT 'FEK.SFEKAUTH' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
  - PERMIT 'FEK.#CUST.PARMLIB' CLASS(DATASET) ACCESS(READ) ID(STCJMON)
- (Facultatif) Autorisation pour les clients de mettre à jour la conversion de nom long/abrégé VSAM pour SCLMDT
  - PERMIT 'FEK.#CUST.LSTRANS\*.\*' CLASS(DATASET) ACCESS(UPDATE) ID(\*)
- (Facultatif) Autorisation pour les développeurs de gestionnaire RAM de mettre à jour la méthode d'accès VSAM CARMA pour CARMA
  - PERMIT 'FEK.#CUST.CRA\*.\*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
- (Facultatif) Autorisation pour les utilisateurs CICS de lire le VSAM du référentiel CRD pour le gestionnaire de déploiement d'application
  - PERMIT 'FEK.#CUST.ADNREP\*.\*' CLASS(DATASET) ACCESS(READ) ID(\*)
- (Facultatif) Autorisation pour les administrateurs CICS de mettre à jour le VSAM du référentiel CRD pour le gestionnaire de déploiement d'application
  - PERMIT 'FEK.#CUST.ADNREP\*.\*' CLASS(DATASET) ACCESS(UPDATE) ID(#cicsadmin)
- (Facultatif) Autorisation pour les utilisateurs CICS de mettre à jour le VSAM du référentiel de manifestes pour le gestionnaire de déploiement d'application
  - PERMIT 'FEK.#CUST.ADNMAN\*.\*' CLASS(DATASET) ACCESS(UPDATE) ID(\*)
- (Facultatif) Autorisation d'accès du serveur TS CICS à la bibliothèque de chargement pour les options bidirectionnelles et le Gestionnaire de déploiement d'application
  - PERMIT 'FEK.SFEKLOAD' CLASS(DATASET) ACCESS(READ) ID(#cicsts)
- (Facultatif) Autorisation d'accès du serveur DB2 à la bibliothèque exec pour le générateur de procédures mémorisées DB2
  - PERMIT 'FEK.SFEKPROC' CLASS(DATASET) ACCESS(READ) ID(#db2)
- Activation des profils de sécurité
  - SETROPTS GENERIC(DATASET) REFRESH

Lorsque vous définissez le contrôle des droits d'accès en lecture (READ) aux fichiers système, vous devez fournir aux serveurs et aux utilisateurs de Developer for System z les droits d'accès en lecture (READ) aux fichiers suivants :

- CEE.SCEERUN
- CEE.SCEERUN2
- CBC.SCLBDLL
- ISP.SISPLD
- ISP.SISPLPA
- SYS1.LINKLIB
- SYS1.SIEALNKE
- REXX.V1R4M0.SEAGLPA

**Remarque :** Lorsque vous utilisez le module Alternate Library for REXX, le nom de la bibliothèque d'exécution REXX par défaut est REXX.\*.SEAGALT au lieu de REXX.\*.SEAGLPA, comme indiqué dans l'exemple ci-dessus.

## Définition des tâches démarrées Developer for System z

Les exemples de commande RACF ci-dessous créent les tâches démarrées JMON, RSED et LOCKD en leur affectant les ID utilisateur protégés (respectivement STCJMON, STCRSE et STCLOCK) et le groupe STCGROUP. Remplacez les marques de réservation #group-id et #user-id-\* par des ID OMVS valides.

- ADDGROUP STCGROUP OMVS(GID(#group-id))  
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
- ADDUSER STCJMON DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - JES JOBMONITOR')  
OMVS(UID(#user-id-jmon) HOME(/tmp) PROGRAM(/bin/sh) NOASSIZEMAX  
NOTHREADSMAX)  
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCRSE DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - RSE DAEMON')  
OMVS(UID(#user-id-rse) HOME(/tmp) PROGRAM(/bin/sh) ASSIZEMAX(2147483647)  
NOTHREADSMAX)  
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- ADDUSER STCLOCK DFLTGROUP(STCGROUP) NOPASSWORD NAME('RDZ - LOCK DAEMON')  
OMVS(UID(#user-id-lock) HOME(/tmp) PROGRAM(/bin/sh) NOASSIZEMAX)  
NOTHREADSMAX)  
DATA('RATIONAL DEVELOPER FOR SYSTEM Z')
- RDEFINE STARTED JMON.\* DATA('RDZ - JES JOBMONITOR')  
STDATA(USER(STCJMON) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED RSED.\* DATA('RDZ - RSE DAEMON')  
STDATA(USER(STCRSE) GROUP(STCGROUP) TRUSTED(NO))
- RDEFINE STARTED LOCKD.\* DATA('RDZ - LOCK DAEMON')  
STDATA(USER(STCLOCK) GROUP(STCGROUP) TRUSTED(NO))
- SETROPTS RACLIST(STARTED) REFRESH

### Remarques :

1. Assurez-vous que les ID utilisateur des tâches démarrées sont protégés en indiquant le mot clé NOPASSWORD.
2. Vérifiez que le serveur RSE possède un uid OMVS unique en raison des autorisations liées à z/OS UNIX accordées à cet uid.
3. Le démon RSE requiert une taille d'espace adresse importante (2 Go) pour fonctionner correctement. Il est recommandé d'attribuer cette valeur à la variable ASSIZEMAX du segment OMVS de l'ID utilisateur STCRSE. Il s'agit de faire en sorte que le démon RSE soit doté de la taille de région requise, quelles que soient les modifications apportées à MAXASSIZE dans SYS1.PARMLIB(BPXPRMxx).



4. RSE requiert également un grand nombre d'unités d'exécution pour fonctionner correctement. Vous pouvez définir la limite dans la variable THREADSMAX du segment OMVS de l'ID utilisateur STCRSE. Il s'agit de faire en sorte que RSE soit doté de la limite d'unité d'exécution requise, quelles que soient les modifications apportées à MAXTHREADS ou MAXTHREADTASKS dans SYS1.PARMLIB(BPXPRMxx). Voir Chapitre 13, «Remarques relatives à l'optimisation», à la page 215 pour déterminer la valeur correcte de la limite d'unité d'exécution.
5. L'ID utilisateur STCJMON est un autre bon moyen de définir THREADSMAX dans le segment OMVS, le moniteur de travaux JES utilisant une unité d'exécution par connexion client.

Vous pouvez être amené à restreindre l'ID utilisateur STCRSE. Les utilisateurs possédant l'attribut RESTRICTED ne peuvent pas accéder aux ressources protégées (MVS) auxquelles ils ne sont pas autorisés à accéder de manière spécifique.

ALTUSER STCRSE RESTRICTED

Pour vous assurer que les utilisateurs restreints n'ont pas accès aux ressources du système de fichiers z/OS UNIX via d'"autres" bits d'autorisation, vous devez définir le profil RESTRICTED.FILESYS.ACCESS dans la classe UNIXPRIV avec UACC(NONE). Pour plus d'informations sur la restriction des ID utilisateur, voir *Security Server RACF Security Administrator's Guide (SA22-7683)*.

**Avertissement :** Si vous utilisez des ID utilisateur restreints, vous devez ajouter de manière explicite le droit d'accès à une ressource avec les commandes **PERMIT TSO** ou **setfacl** z/OS UNIX. Cela inclut les ressources dans lesquelles la documentation Developer for System z utilise UACC (tel que le profil \*\* dans la classe PROGRAM) ou qui se fondent sur les conventions z/OS UNIX (lorsque tous les utilisateurs possèdent les droits d'accès en lecture et en exécution aux bibliothèques Java). Vous devez les essayer avant de les activer sur un système de production.

## Définition de la sécurité des commandes JES

Le moniteur de travaux JES émet toutes les commandes opérateur JES demandées par un utilisateur via une console EMCS dont le nom est contrôlé avec la directive `CONSOLE_NAME`, comme indiqué à la section «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27.

Dans l'exemple suivant, les commandes RACF donnent aux utilisateurs de Developer for System z un accès conditionnel à un jeu limité de commandes JES (Mettre en attente, Libérer, Annuler et Purger). Les utilisateurs possèdent des droits d'exécution uniquement s'ils lancent les commandes via le moniteur de travaux JES. Remplacez la marque de réservation `#console` par le nom réel de la console.

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

### Remarque :

- L'utilisation de la console est autorisée si aucun profil `MVS.MCSOPER.#console` n'est défini.
- La classe `CONSOLE` doit être active pour permettre le fonctionnement de `WHEN(CONSOLE(JMON))` mais il n'y a pas de vérification réelle du profil dans la classe `CONSOLE` pour les consoles EMCS.



- Ne remplacez pas JMON par le nom réel de la console dans la clause WHEN(CONSOLE(JMON)). Le mot clé JMON représente l'application de point d'entrée, pas le nom de la console.

**Avertissement :** La définition des commandes JES à l'aide de l'accès universel NONE dans votre logiciel de sécurité peut avoir une incidence sur les autres applications et opérations. Vous devez les essayer avant de les activer sur un système de production.

Le tableau 27 et le tableau 28 présentent des commandes opérateur soumises pour JES2 et JES3, et les profils de sécurité discrets qui peuvent être utilisés pour les protéger.

Tableau 27. Commandes opérateur du moniteur de travaux JES2

Action	Commande	Profil OPERCMDS	Droit d'accès requis
Mettre en attente	\$Hx(jobid) avec x = {J, S ou T}	jesname.MODIFYHOLD.BAT jesname.MODIFYHOLD.STC jesname.MODIFYHOLD.TSU	UPDATE
Libérer	\$Ax(jobid) avec x = {J, S ou T}	jesname.MODIFYRELEASE.BAT jesname.MODIFYRELEASE.STC jesname.MODIFYRELEASE.TSU	UPDATE
Annuler	\$Cx(jobid) avec x = {J, S ou T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE
Purger	\$Cx(jobid),P avec x = {J, S ou T}	jesname.CANCEL.BAT jesname.CANCEL.STC jesname.CANCEL.TSU	UPDATE

Tableau 28. Commandes opérateur du moniteur de travaux JES3

Action	Commande	Profil OPERCMDS	Droit d'accès requis
Mettre en attente	*F,J=jobid,H	jesname.MODIFY.JOB	UPDATE
Libérer	*F,J=jobid,R	jesname.MODIFY.JOB	UPDATE
Annuler	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE
Purger	*F,J=jobid,C	jesname.MODIFY.JOB	UPDATE

**Remarque :**

- Les commandes opérateur JES Mettre en attente, Publier, Annuler et Purger et Afficher le JCL peuvent être exécutées uniquement sur des fichiers spoule dont l'ID utilisateur client est le propriétaire, sauf si LIMIT\_COMMANDS= est indiqué avec la valeur LIMITED ou NOLIMIT dans le fichier de configuration du moniteur de travaux JES. Pour de plus amples informations sur cette fonction, voir «Actions sur les travaux - Limitations sur les cibles», à la page 168.
- Les utilisateurs peuvent parcourir n'importe quel fichier spoule, sauf si LIMIT\_VIEW=USERID est défini dans le fichier de configuration du moniteur de travaux JES. Pour de plus amples informations sur cette fonction, voir «Accès aux fichiers spoule», à la page 171.
- Sans autorisation pour ces commandes opérateur, les utilisateurs peuvent toujours soumettre des travaux et lire les sorties de travaux via le moniteur de travaux JES s'ils disposent de droits d'accès

suffisants à des profils qui protègent ces ressources (comme celles des classes JESINPUT, JESJOBS et JESSPOOL).

Supposons que l'accès à l'identité du serveur du moniteur de travaux JES lors de la création d'une console JMON à partir d'une session TSO est empêché par votre logiciel de sécurité. Même si la console peut être créée, le point d'entrée est différent (moniteur de travaux JES/TSO). Les commandes JES exécutées par cette console échouent lors du contrôle de sécurité si la sécurité est configurée comme indiqué dans cette publication et que l'utilisateur ne dispose pas de droits d'accès aux commandes JES via d'autres procédures.

## Définition de RSE comme serveur z/OS UNIX sécurisé

RSE requiert un accès UPDATE au profil BPX.SERVER pour créer/supprimer l'environnement de sécurité de l'unité d'exécution du client. Si ce profil n'est pas défini, UID(0) est nécessaire pour RSE.

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

**Avertissement :** La définition du profil BPX.SERVER permet de configurer z/OS UNIX comme un commutateur global qui bascule de la sécurité de niveau UNIX à la sécurité plus étendue de z/OS UNIX. Cette configuration peut avoir une incidence sur d'autres applications et opérations z/OS UNIX. Vous devez les essayer avant de les activer sur un système de production. Pour plus d'informations sur les différents niveaux de sécurité, voir *UNIX System Services Planning* (GA22-7800).

## Définition de bibliothèques contrôlées par programme MVS pour RSE

Les serveurs disposant des droits BPX.SERVER doivent être exécutés dans un environnement propre, contrôlé par un programme. Cela signifie que tous les programmes appelés par RSE doivent également être contrôlés par programme. Pour les bibliothèques de chargement MVS, le contrôle par programme est géré par votre logiciel de sécurité.

RSE utilise la bibliothèque système (SYS1.LINKLIB), l'environnement d'exécution de Language Environment' (CEE.SCEERUN\*) et la bibliothèque de chargement de la passerelle client TSO/ISPF d'ISPF (ISP.SISPLOAD).

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

**Remarque :** N'utilisez pas le profil \*\* si le profil \* existe déjà dans la classe PROGRAM. Cela occulterait et compliquerait le chemin de recherche utilisé par votre logiciel de sécurité. Dans ce cas de figure, vous devez fusionner la définition \* existante et la nouvelle définition \*\*. IBM vous recommande d'utiliser le profil \*\*, comme spécifié dans le document *Security Server RACF Security Administrator's Guide* (SA22-7683).

Les bibliothèques (prérequis) suivantes doivent être contrôlées par un programme pour la prise en charge des services facultatifs. Cette liste n'inclut pas les fichiers spécifiques d'un produit avec lequel interagit Developer for System z (IBM DebugTool, par exemple).

- Autre bibliothèque d'exécution REXX (pour SCLM Developer Toolkit)
  - REXX.\*.SEAGALT
- Bibliothèque de chargement système (pour le chiffrement SSL)
  - SYS1.SIEALNKE
- Bibliothèque de chargement du port d'écoute de File Manager (pour l'intégration de File Manager)
  - FMN.SFMNMODA

**Remarque :** Les bibliothèques qui sont conçues pour le positionnement LSA requièrent également des autorisations de contrôle de programme si l'utilisateur y accède via LINKLIST ou STEPLIB. La présente publication concerne l'utilisation des bibliothèques LPA suivantes :

- ISPF (pour la passerelle client TSO/ISPF)
  - ISP.SISPLPA
- Bibliothèque d'exécution REXX (pour SCLM Developer Toolkit)
  - REXX.\*.SEAGLPA
- Developer for System z (pour CARMA)
  - FEK.SFEKLPA

## Définition de la protection d'application pour RSE

Lors de la connexion du client, le démon RSE vérifie que l'utilisateur est autorisé à utiliser l'application.

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

**Remarque :**

Comme indiqué plus en détails dans «Définition du support de mots de passe PassTicket pour RSE», RSE prend en charge l'utilisation d'un ID application autre que FEKAPPL. La définition de classe APPL doit correspondre à l'ID application réel utilisé par RSE.

**Avertissement :** La demande de connexion du client n'aboutit pas si le profil de l'application n'est pas défini ou si l'utilisateur n'a pas l'accès en lecture au profil.

## Définition du support de mots de passe PassTicket pour RSE

Le mot de passe du client (ou toute autre méthode d'identification, telle que les certificats X.509) est utilisé uniquement pour vérifier son identité lors de la connexion. Par la suite, les mots de passe PassTicket permettent de gérer la sécurité des unités d'exécution.

Les PassTickets sont des mots de passe générés par le système pour une durée d'environ 10 minutes. Ils s'appuient sur une clé confidentielle. Cette clé est un nombre de 64 bits (16 caractères hexadécimaux). Dans les exemples de commande RACF, remplacez la marque de réservation key16 ci-dessous par une chaîne hexadécimale de 16 caractères fournie par l'utilisateur (les caractères 0-9 et A-F).

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

RSE prend en charge l'utilisation d'un ID application autre que FEKAPPL. Supprimez la mise en commentaire et personnalisez l'option APPLID=FEKAPPL dans `rsed.envvars` pour l'activer, comme indiqué à la section «Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JAVAOPTS`», à la page 42. Les définitions de classe PTKTDATA doivent correspondre à l'ID application réel utilisé par RSE.

Vous ne devez pas utiliser OMVSAPPL comme ID d'application, car il ouvrira la clé confidentielle de la plupart des applications z/OS UNIX. De la même manière, vous ne devez pas utiliser l'ID d'application par défaut MVS, qui est MVS suivi par l'ID SMF du système, car il ouvrira la clé confidentielle de la plupart des applications MVS (y compris les travaux par lots des utilisateurs).

**Remarque :**

- Si la classe PTKTDATA est déjà définie, vérifiez qu'elle est définie en tant que classe générique avant de créer les profils indiqués ci-dessus. La prise en charge de caractères génériques dans la classe PTKTDATA est une nouveauté disponible à partir de z/OS édition 1.7, avec l'introduction d'une interface Java pour les mots de passe PassTicket.
- Remplacez le caractère générique (\*) dans la définition IRRPTAUTH.FEKAPPL.\* par un masque d'ID utilisateur valide afin de limiter les ID utilisateur pour lesquels RSE peut générer un mot de passe PassTicket.
- En fonction des paramètres RACF configurés, l'utilisateur qui définit un profil peut également figurer dans la liste d'accès du profil. Il est conseillé de supprimer ce droit pour les profils PTKTDATA.
- Le moniteur de travaux JES et RSE doivent posséder le même ID application pour permettre au gestionnaire d'évaluer les mots de passe PassTicket présentés par RSE.
- Si un produit cryptographique est installé et disponible sur le système, vous pouvez chiffrer la clé de l'application de connexion sécurisée pour renforcer la protection. Pour ce faire, utilisez le mot clé KEYENCRYPTED au lieu du mot clé KEYMASKED. Pour plus d'informations, voir *Security Server RACF Security Administrator's Guide* (SA22-7683).

<p><b>Avertissement :</b> La demande de connexion du client n'aboutit pas si les mots de passe PassTickets ne sont pas correctement configurés.</p>
---

## Définition de fichiers contrôlés par programme z/OS UNIX pour RSE

Les serveurs disposant des droits BPX.SERVER doivent être exécutés dans un environnement propre, contrôlé par un programme. Cela signifie que tous les programmes appelés par RSE doivent également être contrôlés par programme. Pour les fichiers z/OS UNIX, le contrôle par programme est géré par la commande

**extattr.** Pour exécuter cette commande vous devez disposer du droit d'accès en lecture (READ) sur BPX.FILEATTR.PROGCTL dans la classe FACILITY ou avoir l'ID utilisateur UID(0).

Le serveur RSE utilise la bibliothèque partagée Java RACF(/usr/lib/libIRRracf.so).

- extattr +p /usr/lib/libIRRracf.so

**Remarque :**

- z/OS 1.9, /usr/lib/libIRRracf.so étant installé sous le contrôle de programme au cours de l'installation RACF SMP/E.
- Comme z/OS 1.10, /usr/lib/libIRRracf.so est un composant de SAF fourni avec le z/OS de base, il est également disponible aux clients non RACF.
- La configuration peut varier si vous utilisez un produit de sécurité autre que RACF. Consultez la documentation de votre produit de sécurité pour de plus amples informations.
- L'installation SMP/E de Developer for System z définit le bit de contrôle par programme pour les programmes RSE internes.
- Utilisez la commande z/OS UNIX **ls -Eog** pour afficher l'état en cours du bit de contrôle par programme (le fichier est contrôlé par programme si la lettre **p** apparaît dans la deuxième chaîne).  

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

## Vérification des paramètres de sécurité

Utilisez les exemples de commande ci-dessous pour afficher les résultats de vos personnalisations de la sécurité.

- Paramètres et classes de sécurité
  - SETROPTS LIST
- Segment OMVS pour les utilisateurs
- LISTUSER #userid NORACF OMVS
- LISTGRP #group-name NORACF OMVS
- Profils de fichier
- LISTGRP FEK
- LISTDSD PREFIX(FEK) ALL
- Tâches démarrées
  - LISTGRP STCGROUP OMVS
  - LISTUSER STCJMON OMVS
  - LISTUSER STCRSE OMVS
  - LISTUSER STCLOCK OMVS
  - RLIST STARTED JMON.\* ALL STDATA
  - RLIST STARTED RSED.\* ALL STDATA
  - RLIST STARTED LOCKD.\* ALL STDATA
- Sécurité de commande JES
  - RLIST CONSOLE JMON ALL
  - RLIST OPERCMDS MVS.MCSOPER.JMON ALL
  - RLIST OPERCMDS JES%,\*\* ALL
- RSE comme serveur z/OS UNIX sécurisé

- RLIST FACILITY BPX.SERVER ALL
- Bibliothèques contrôlées par programme MVS pour RSE
  - RLIST PROGRAM \*\* ALL
- Protection des applications pour RSE
  - RLIST APPL FEKAPPL ALL
- Prise en charge de PassTicket pour RSE
  - RLIST PTKTDATA FEKAPPL ALL SSIGNON
  - RLIST PTKTDATA IRRPTAUTH.FEKAPPL.\* ALL
- Fichiers contrôlés par programme z/OS UNIX pour RSE
  - ls -E /usr/lib/libIRRracf.so

---

## Chapitre 11. Compréhension de Developer for System z

Le système hôte Developer for System z est composé de plusieurs composants qui interagissent pour permettre au client d'accéder aux services et données de l'hôte. En comprenant bien la conception de ces composants, vous pouvez prendre les bonnes décisions de configuration.

Les rubriques suivantes sont traitées dans le présent chapitre :

- «Présentation du composant»
- «RSE comme application Java», à la page 195
- «Propriétaires de tâches», à la page 196
- «Flux de connexion», à la page 198
- «Démon lock», à la page 200
- «Structure de répertoire z/OS UNIX», à la page 202

---

### Présentation du composant

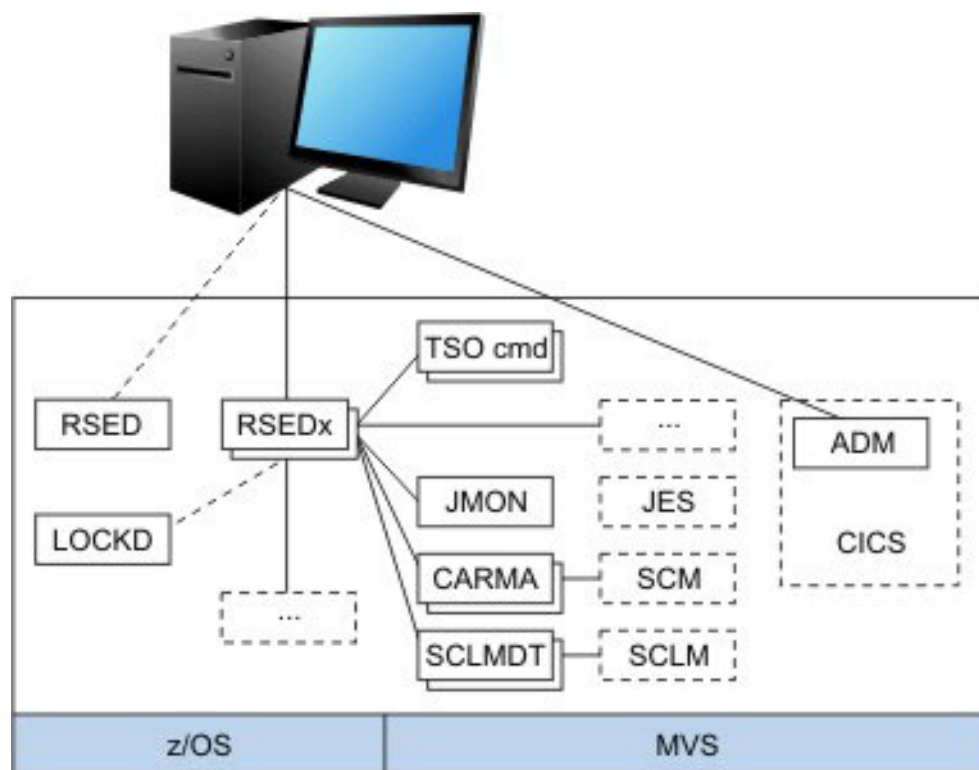


Figure 41. Présentation du composant

La figure 41 illustre une présentation générale de Developer for System z sur votre système hôte.

- L'Explorateur de systèmes distants (RSE) fournit des services de base, comme la connexion du client à l'hôte et le démarrage d'autres serveurs pour des services spécifiques. RSE se compose de deux entités logiques :

- Le démon RSE (RSED), qui gère la configuration de la connexion. Il est également en charge de l'exécution en mode serveur unique. Pour se faire, le démon RSE crée un ou plusieurs processus enfants appelés pools d'unités d'exécution RSE (RSEDx).
- Le serveur RSE qui gère les demandes client individuelles. Un serveur RSE est actif comme une unité d'exécution à l'intérieur d'un pool d'unités d'exécution RSE.
- Le démon lock (LOCKD) fournit des services de suivi pour les verrous de fichiers.
- Le service de Commandes TSO (TSO cmd) offre une interface de type par lots pour les commandes TSO et ISPF.
- Le moniteur de travaux JES (JMON) fournit tous les services relatifs à JES.
- Common Access Repository Manager (CARMA) offre une interface permettant d'interagir avec les gestionnaires d'accès au référentiel (SCM), comme CA Endevor.
- SCLM Developer Toolkit (SCLMDT) offre une interface permettant d'améliorer et d'interagir avec SCLM.
- Le gestionnaire de déploiement d'application (ADM) propose différents services liés à CICS.
- Plusieurs services sont disponibles. Ils peuvent être fournis par Developer for System z ou par d'autres logiciels corequis.

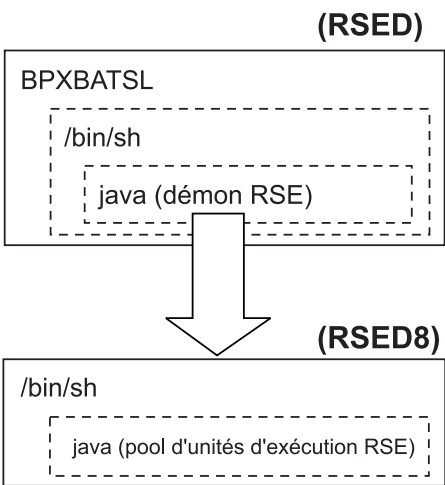
La description du paragraphe et de la liste précédent illustre le rôle central attribué à RSE. A quelques exceptions près, toute la communication du client passe par RSE. Cela permet de faciliter la configuration du réseau liée à la sécurité, étant donné que seul un ensemble limité de ports est utilisé pour la communication hôte-client.

Pour gérer les connexions et charges de travail provenant des clients, RSE est composé d'un espace adresse de démon, qui permet de contrôler les espaces adresse du groupe d'unités d'exécution. Le démon agit comme un point focal pour la connexion et la gestion, alors que les pools d'unités d'exécution traitent les charges de travail du client. Selon les valeurs définies dans le fichier de configuration `rsed.envvars` et la quantité de connexions client réelles, le démon peut démarrer plusieurs espaces adresse de pool d'unités d'exécution.



# RSE comme application Java

## Processus z/OS UNIX



## Utilisation du stockage Java

Système - partagé
Système - privé
Code (z/OS UNIX, Java, RSE)
Taille de pile Java
Inutilisé

JOBNAME	Etat	PID	PPID	Commande
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...

Figure 42. RSE comme application Java

La figure 42 présente une vue de base de l'utilisation des ressources (processus et stockage) par RSE.

RSE est une application Java, ce qui signifie qu'il est actif dans l'environnement z/OS UNIX. Cela permet de faciliter l'accès à différentes plateformes hôte et de simplifier la communication avec le client Developer for System z, qui est également une application Java (reposant sur Eclipse). Par conséquent, il est très utile d'avoir des connaissances de base du fonctionnement de z/OS UNIX et Java pour comprendre Developer for System z.

Dans z/OS UNIX, un programme s'exécute dans un processus, qui est identifié par un PID (ID processus). Chaque programme est actif dans son propre processus. Par conséquent, l'appel d'un autre programme crée un processus. Le processus qui en a démarré un autre est référencé par un PPID (PID parent), le nouveau processus étant appelé processus enfant. Ce dernier peut s'exécuter dans le même espace adresse ou être généré (créé) dans un nouvel espace adresse. L'exécution d'un nouveau processus dans le même espace adresse est comparable à l'exécution d'une commande dans TSO, la génération d'un processus dans un nouvel espace adresse s'apparentant à la soumission d'un nouveau travail.

Notez qu'un processus peut être à unité d'exécution unique ou à unités d'exécution multiples. Dans une application à unités d'exécution multiples (RSE, par exemple), les différentes unités d'exécution rivalisent pour des ressources système comme si elles se trouvaient dans des espaces adresse séparés (avec moins de temps système).

La mise en correspondance de ces informations de processus avec l'exemple RSE de la figure 42, à la page 195 permet d'obtenir le flux suivant :

1. Lorsque la tâche RSED est démarrée, elle exécute BPXBATSL, qui appelle z/OS UNIX et crée un environnement shell – PID 50331904.
2. Dans ce processus, le script de shell `rsed.sh` est exécuté, ce qui permet de lancer un processus distinct (`/bin/sh`) – PID 67109114.
3. Le script de shell définit les variables d'environnement définies dans `rsed.envvars` et exécute Java avec les paramètres requis afin de démarrer le démon RSE – PID 50331949.
4. Le démon RSE génère un shell dans un processus enfant (RSED8) – PID 307.
5. Dans ce shell, une valeur est attribuée aux variables d'environnement définies dans `rsed.envvars` et Java est exécuté avec les paramètres obligatoires afin de démarrer le pool d'unités d'exécution RSE – PID 308.

Les applications Java (RSE, par exemple) n'allouent pas de mémoire directement. Elles utilisent les services de gestion de mémoire Java. Ces services (l'allocation de mémoire, la libération de mémoire et la récupération de place, par exemple) fonctionnent dans les limites du segment de mémoire Java. Les tailles minimale et maximale du segment de mémoire sont définies (de manière implicite ou explicite) au démarrage de Java.

Ainsi, l'occupation de la plus grande partie de l'espace adresse disponible consiste à trouver le juste équilibre entre une taille de segment de mémoire importante et une place suffisante permettant à z/OS de stocker une quantité variable de blocs de contrôle du système (selon le nombre d'unités d'exécution actives).

---

## Propriétaires de tâches

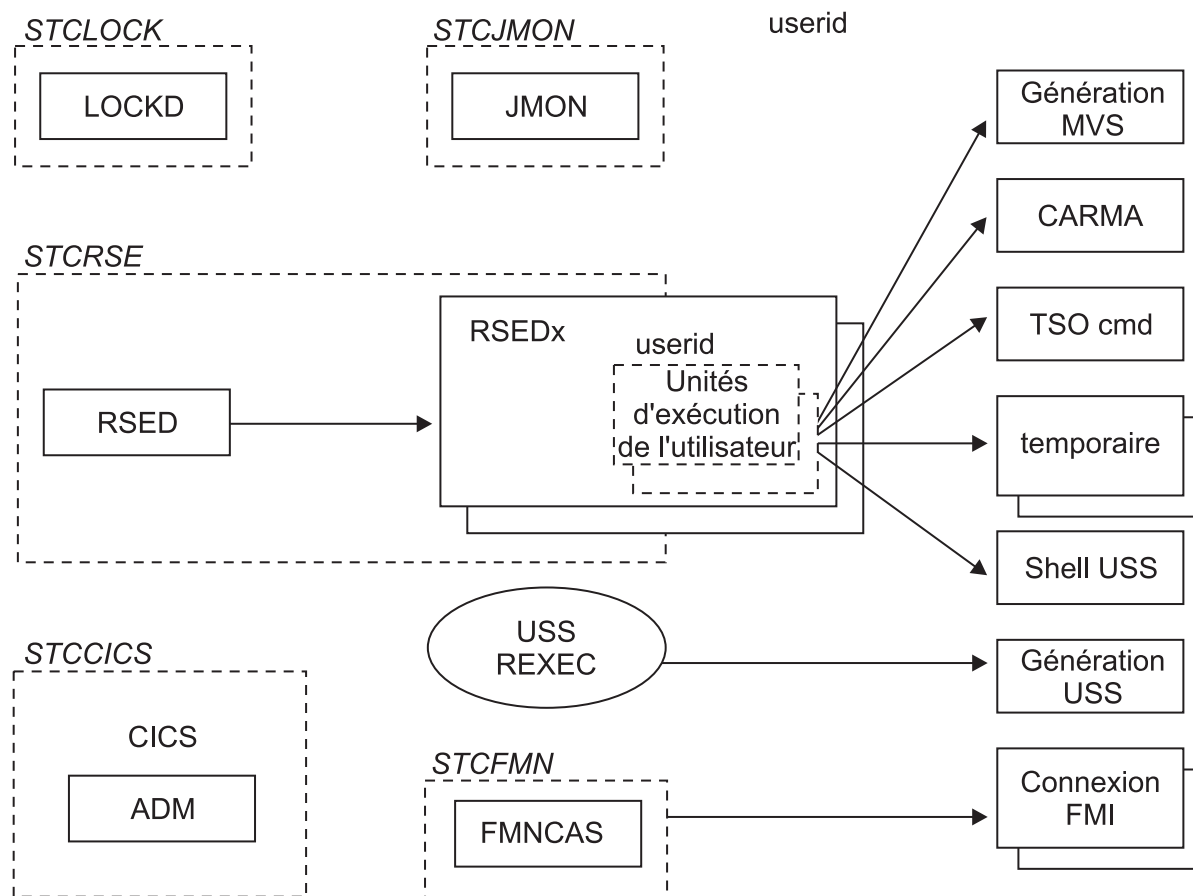


Figure 43. Propriétaires de tâches

La figure 43 offre une présentation de base du propriétaire des données d'identification utilisées par différentes tâches de Developer for System z.

La propriété d'une tâche peut être divisée en deux sections. Les tâches démarrées appartiennent à l'ID utilisateur qui est attribué à la tâche démarrée dans le logiciel de sécurité. Toutes les autres tâches, avec les pools d'unités d'exécution RSE (RSEDx) comme exception, appartiennent à l'ID utilisateur du client.

La figure 43 présente les tâches démarrées de Developer for system z (LOCKD, JMON et RSED) ainsi que des exemples de tâches démarrées et des services système avec qui Developer for System z communique. Le gestionnaire de déploiement d'application (ADM) est actif au sein d'une région CICS. FMNCAS est la tâche démarrée de File Manager (gestionnaire de fichiers). La balise USS REXEC représente le service REXEC (ou SSH) z/OS UNIX.

Le démon RSE (RSED) crée un ou plusieurs espaces adresse de pools d'unités d'exécution (RSEDx) dédiés aux demandes des clients. Chaque pool d'unités d'exécution RSE prend en charge plusieurs clients et appartient au même utilisateur que celui du démon RSE. Chaque client possède sa propre unité d'exécution au sein d'un pool d'unités d'exécution et ces unités d'exécution appartiennent à l'ID utilisateur client.

Selon les actions menées par le client, un ou plusieurs espaces adresse supplémentaires peuvent être démarrés, tous appartenant à l'ID utilisateur du client, pour exécuter l'action demandée. Ces espaces adresse peuvent être un

travail par lots MVS, une transaction APPC ou un processus enfant z/OS UNIX. Notez qu'un processus enfant est actif dans un initiateur z/OS UNIX (BPXAS), et le processus enfant apparaît comme une tâche démarrée dans JES.

La création de ces espaces adresse est le plus souvent déclenchée par une unité d'exécution d'utilisateur dans un pool d'unités d'exécution, soit directement soit à l'aide d'un service système comme ISPF. L'espace adresse pourrait très bien être aussi créé par un tiers. Par exemple, File Manager démarrera un nouvel espace adresse pour chaque fichier (ou membre) qu'il a à traiter en faveur de Developer for System z. REXEC ou SSH z/OS UNIX est impliqué lorsque un démarrage est généré dans z/OS UNIX.

Les espaces adresse spécifiques de l'utilisateur prennent fin à l'achèvement de la tâche ou à l'expiration d'un temps d'inactivité. Les tâches démarrées restent actives. Les espaces adresse répertoriés dans la figure 43, à la page 197 restent dans le système suffisamment longtemps pour être visibles. Toutefois, vous devez être conscient qu'en raison de la conception de z/OS UNIX, il existe aussi des espaces adresses temporaires de durée de vie courte.

## Flux de connexion

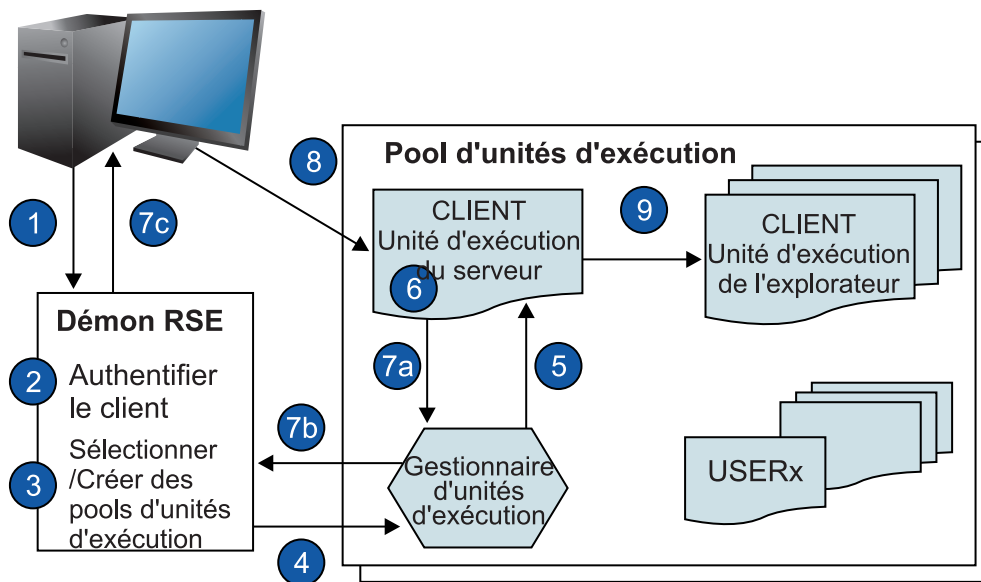


Figure 44. flux de connexion

La figure 44 présente une vue schématique d'une connexion du client à l'hôte grâce à Developer for System z. De même, elle explique brièvement comment les mots de passe PassTickets sont utilisés.

1. Le client se connecte au démon (port 4035).
2. Le démon RSE authentifie le client en utilisant les données d'identification présentées par le client.
3. Le démon RSE sélectionne un pool d'unités d'exécution existant ou en démarre un s'ils sont tous saturés.
4. Le démon RSE transmet l'ID utilisateur du client au pool d'unités d'exécution.

5. Le pool d'unités d'exécution crée une unité d'exécution de serveur RSE propre au client, en utilisant l'ID utilisateur du client et un mot de passe PassTicket pour l'authentification.
6. L'unité d'exécution de serveur du client associe un port pour les futures communications du client.
7. L'unité d'exécution de serveur renvoie le numéro de port pour permettre au client de s'y connecter.
8. Le client se déconnecte du démon RSE et se connecte au numéro de port fourni.
9. L'unité d'exécution de serveur du client démarre d'autres unités d'exécution propres à l'utilisateur (mineur), en utilisant toujours l'ID utilisateur du client et un mot de passe PassTicket pour l'authentification. Ces unités d'exécution offrent des services propres à l'utilisateur demandés par le client.

La description ci-dessus illustre la conception orientée unité d'exécution de RSE. Au lieu de démarrer un espace adresse par utilisateur, plusieurs utilisateurs sont gérés par un seul espace adresse du pool d'unités d'exécution. Dans le pool d'unités d'exécution, chaque version mineure (un service propre à l'utilisateur) est active dans sa propre unité d'exécution dans le contexte de sécurité de l'utilisateur qui lui est attribué, garantissant la sécurité de la configuration. Cette conception permet de gérer un grand nombre d'utilisateurs avec une quantité de ressources limitée, ce qui n'implique pas que chaque client va utiliser plusieurs unités d'exécution (au moins 16, selon les tâches réalisées).

du point de vue du réseau, Developer for system z agit comme FTP en mode passif. Le client se connecte à un point focal (le démon RSE), supprime la connexion, puis se connecte de nouveau à un numéro de port fourni par le point focal. La logique ci-dessous permet de contrôler la sélection du port utilisé pour la deuxième connexion :

1. Si le client a indiqué un numéro de port différent de zéro dans l'onglet des propriétés du sous-système, le serveur RSE utilise ce port pour assurer la liaison. Si ce port n'est pas disponible, la connexion n'aboutit pas.
2. Si `_RSE_PORTRANGE` est spécifié dans `rsed.envvars`, le serveur RSE établit une liaison avec un port à partir de cette plage. Si aucun port n'est disponible, la connexion n'aboutit pas. Notez que ce serveur RSE n'a pas exclusivement besoin du port pendant la durée de la connexion client. Aucun autre serveur RSE ne peut établir une liaison avec le port que dans l'intervalle de temps entre la liaison (du serveur) et la connexion (du client).
3. Si aucune limite n'est définie, le serveur RSE établit une liaison avec le port 0. Il en résulte que TCP/IP choisit le numéro de port.

L'utilisation du mot de passe PassTickets pour tous les services z/OS impliquant une authentification permet à Developer for System z d'appeler ces services à volonté, sans stocker le mot de passe ni inviter continuellement l'utilisateur à l'indiquer. L'utilisation des mots de passe PassTickets pour tous les services z/OS permet également d'utiliser d'autres méthodes d'authentification (mots de passe à usage unique et certificats X.509, par exemple).

## Démon lock

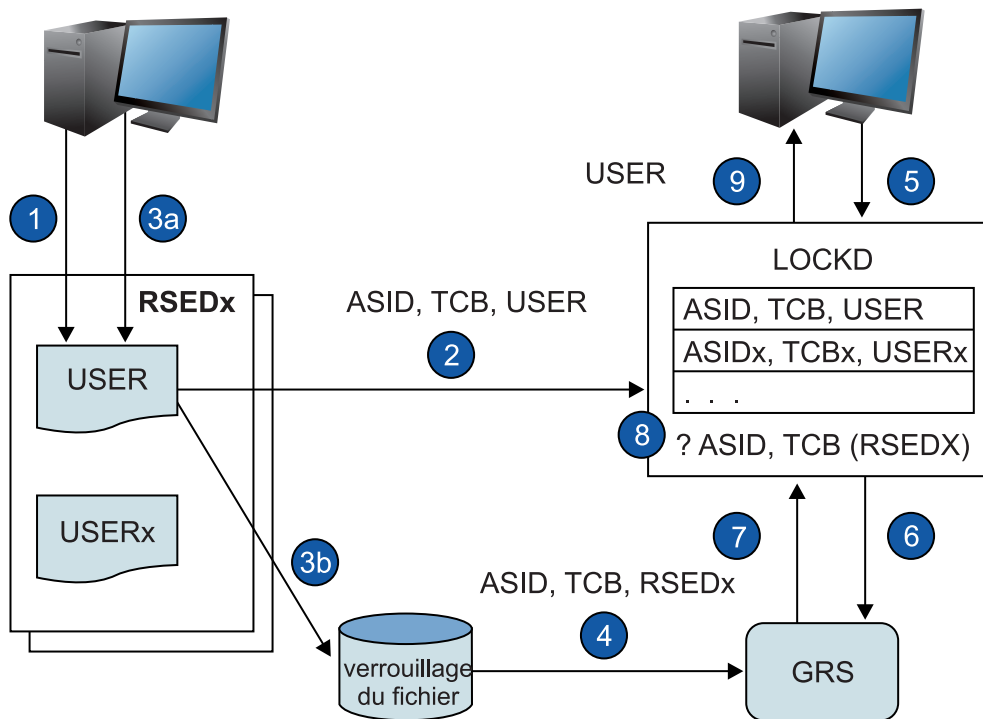


Figure 45. Flux du démon lock

La figure 45 explique schématiquement comment le démon lock détermine le client Developer for System z qui détient un verrou de fichier.

1. Le client se connecte, ce qui permet de créer une unité d'exécution de serveur RSE propre à l'utilisateur (USER) à l'intérieur d'un pool d'unités d'exécution (RSEDx).
2. Le serveur RSE enregistre un utilisateur qui vient de se connecter avec le démon lock. Les informations d'enregistrement contiennent l'identificateur d'espace adresse (qui est celui du pool d'unités d'exécution), l'identificateur de bloc de contrôle des tâches (propre à l'utilisateur) et l'ID utilisateur.
3. Le client ouvre un fichier en mode d'édition, qui informe le serveur RSE d'appliquer un verrou exclusif au fichier.
4. Le système enregistre l'identificateur d'espace adresse, le bloc de contrôle des tâches et le nom de tâche (RSEDx) du demandeur dans le cadre du processus de verrouillage. Ces informations sont stockées dans les files d'attente de sérialisation d'accès des ressources partagées (GRS).
5. Un opérateur (ou le serveur RSE pour le compte d'un client) demande au démon lock l'état de verrouillage du fichier.
6. Le démon lock analyse les files d'attente GRS pour savoir si le fichier est verrouillé.
7. Le démon extrait l'identificateur d'espace adresse, le bloc de contrôle des tâches et le nom de la tâche du propriétaire du verrou.
8. L'identificateur d'espace adresse et le bloc de contrôle des tâches extraits sont comparés à ceux des clients enregistrés.

9. L'ID utilisateur client associé est renvoyé au demandeur en cas de correspondance. Sinon, le nom de tâche extrait de la sérialisation d'accès des ressources partagées est renvoyé.

Avec la configuration à un seul serveur de Developer for System z, où plusieurs utilisateurs sont attribués à un seul espace adresse de pool d'unités d'exécution, z/OS n'a plus la possibilité de savoir qui possède un verrou sur un fichier ou un membre. Les commandes système s'arrêtent au niveau de l'espace adresse, qui correspond au pool d'unités d'exécution.

Pour remédier à cette situation, Developer for System z propose le démon lock, qui peut suivre tous les verrous de fichier/membre placés par les utilisateurs RSE, et ceux placés par d'autres produits (ISPF, par exemple).

Le serveur RSE enregistre un utilisateur qui vient de se connecter avec le démon lock. Les informations d'enregistrement contiennent l'identificateur d'espace adresse (qui est celui du pool d'unités d'exécution), l'identificateur de bloc de contrôle des tâches (propre à l'utilisateur) et l'ID utilisateur.

Notez que l'enregistrement n'a lieu qu'au moment de la connexion. Par conséquent, tous les utilisateurs RSE actifs avant le démarrage (ou le redémarrage) du démon lock ne sont pas enregistrés.

Lorsque le démon lock reçoit une requête de fichier (au moyen d'une commande d'opérateur modify query ou de la part du client grâce au serveur RSE), il analyse les files d'attente de sérialisation d'accès des ressources partagées (GRS) du système. Si l'identificateur d'espace adresse et le bloc de contrôle des tâches correspondent à ceux d'un utilisateur enregistré, l'ID utilisateur est renvoyé en tant que propriétaire du verrou. Sinon, le nom de travail/l'ID utilisateur associé à l'identificateur d'espace adresse est renvoyé en tant que propriétaire du verrou.

Un message de console (FEK513W) avec les informations d'enregistrement s'affichent si l'enregistrement n'aboutit pas. Cela permet à un opérateur de mettre les valeurs en correspondance en fonction de la sortie d'une commande opérateur **DISPLAY GRS,RES=(\*,dataset\*)** afin de trouver le propriétaire du verrou.

**Remarque :** Les enregistrements qui ont abouti figurent également dans la définition de données STDOUT du serveur si la valeur 2 est attribuée à log\_level. Il est utile de procéder à la mise en correspondance manuelle des enregistrements ayant abouti supprimés après le redémarrage du démon lock.

## Libération d'un verrou

Dans des circonstances normales, un fichier ou un membre est verrouillé lorsque le client l'ouvre en mode édition, et libéré lorsque le client ferme la session d'édition.

Certaines conditions d'erreur peuvent gêner le fonctionnement prévu de ce mécanisme. Dans ce cas, l'utilisateur qui détient le verrou peut être annulé à l'aide de la commande d'opérateur **modify cancel** de RSE (voir le Chapitre 8, «Commandes de l'opérateur», à la page 127). Les verrous du fichier actif appartenant à cet utilisateur sont libérés lors du processus.

## Structure de répertoire z/OS UNIX

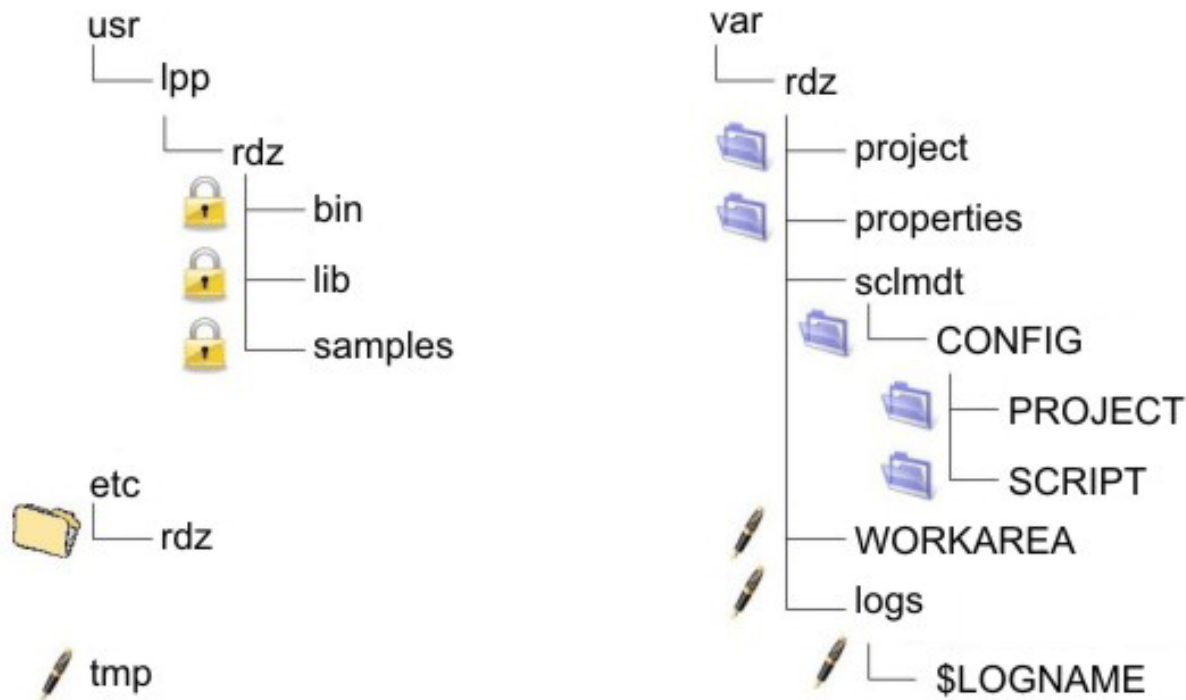


Figure 46. structure de répertoire z/OS UNIX

La figure 46 présente les répertoires de z/OS UNIX utilisés par Developer for System z. La liste suivante décrit chaque répertoire en contact avec Developer for System z, son mode de changement et qui gère les données qu'il contient.

- /usr/lpp/rdz/ est la racine du code produit Developer for System z. L'emplacement réel est spécifié dans les tâches démarrées RSED et LOCKD (variable HOME). Les fichiers sont gérés par SMP/E.
- /etc/rdz/ détient le RSE et la version mineure associée aux fichiers de configuration. L'emplacement réel est spécifié dans les tâches démarrées RSED et LOCKD (variable CNFG). Les fichiers sont gérés par le programmeur système.
- /var/rdz/sclmdt/CONFIG/ comporte les fichiers de configuration SCLMDT généraux. L'emplacement réel est spécifié dans rsed.envvars (variable SCLMDT\_CONF\_HOME). Les fichiers sont gérés par l'administrateur SCLM.
- /var/rdz/sclmdt/CONFIG/PROJECT/ comporte les fichiers de configuration du projet SCLMDT. L'emplacement réel est spécifié dans rsed.envvars (variable SCLMDT\_CONF\_HOME). Les fichiers sont gérés par l'administrateur SCLM.
- /var/rdz/sclmdt/CONFIG/script/ comporte les scripts associés à SCLMDT qui peuvent être utilisés par d'autres produits. L'emplacement réel n'est indiqué nulle part. Les fichiers sont gérés par l'administrateur SCLM.
- /var/rdz/projects/ comporte les fichiers de définition de projet résidant sur l'hôte. L'emplacement réel est spécifié dans projectcfg.properties (variable PROJECT-HOME). Les fichiers sont gérés par un responsable de projet ou un développeur.



- `/var/rdz/properties/` comporte les groupes de propriétés résidant sur l'hôte. L'emplacement réel est indiqué dans `propertiescfg.properties` (variables `PROPERTY-GROUP` et `DEFAULT-VALUES`). Les fichiers sont gérés par un responsable de projet ou un développeur.
- `/var/rdz/logs/` comporte les fichiers journaux du démon RSE et des serveurs de pool d'unités d'exécution. L'emplacement réel est indiqué dans `rsed.envvars` (variable `daemon.log`). Les fichiers sont gérés par RSE.
- `/var/rdz/logs/$LOGNAME/` comporte les fichiers journaux propres à l'utilisateur du serveur et versions mineures de RSE. L'emplacement réel est indiqué dans `rsed.envvars` (variable `user.log` et `DSTORE_LOG_DIRECTORY`). Les fichiers sont gérés par RSE et les versions mineures.

**Remarque :** `/var/rdz/logs/` requiert le masque de contrôle des données de droits `777` permettant à chaque client de créer son répertoire `$LOGNAME` et d'enregistrer les fichiers journaux propres à l'utilisateur.

- `/var/rdz/WORKAREA/` est utilisé par la passerelle de client TSO/ISPF d'ISPF et par SCLMDT pour transférer des données entre z/OS UNIX et les espaces adresses MVS. L'emplacement réel est spécifié dans `location rsed.envvars` (variable `_CMDSERV_WORK_HOME`). Les fichiers sont gérés par ISPF et SCLMDT.

**Remarque :** `/var/rdz/WORKAREA/` requiert le masque de contrôle des données de droits `777` permettant à chaque client de créer des fichiers temporaires.

- `/tmp/` est utilisé par la passerelle de client TSO/ISPF d'ISPF et les différentes versions mineures pour enregistrer les données temporaires. L'emplacement ne peut pas être personnalisé. Les fichiers sont gérés par ISPF et les versions mineures. Il s'agit également de l'emplacement par défaut des fichiers de vidage Java, qui peuvent être personnalisés avec la variable `_CEE_DUMPTARG` de `rsed.envvars`.

**Remarque :** `/tmp/` requiert le masque de contrôle des données de droits `777` permettant à chaque client de créer des fichiers temporaires.

## Droits de mise à jour des administrateurs non système

Les données de certains répertoires (`/var/rdz/projects/`, par exemple) sont gérées par des administrateurs non système (des responsables de projet, par exemple) susceptibles de ne pas détenir les droits de mise à jour dans z/OS UNIX. Si un seul ID utilisateur gère les fichiers, il n'y a aucun problème lorsque l'ID utilisateur devient le propriétaire du répertoire et de tous les éléments qu'il contient.

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

Lorsque plusieurs ID utilisateur ont besoin des droits de mise à jour du répertoire, vous pouvez gérer les bits groupe-droits.

1. Créez un groupe dans votre logiciel de sécurité comportant un segment OMVS valide et connectez tous les ID utilisateur demandant un accès UPDATE. De préférence, il s'agit de leur groupe par défaut.

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

2. Utilisez la commande z/OS UNIX **chgrp** pour attribuer le groupe au répertoire et tous les fichiers qu'il contient. Cette commande doit être répétée à chaque fois qu'un nouveau fichier est ajouté et que l'ID groupe souhaité n'est pas le groupe par défaut correspondant à l'ID utilisateur qui a ajouté le fichier.

```
chgrp -R IBMUSER /var/rdz/projects/
```

3. Utilisez la commande z/OS UNIX **chmod** pour donner à l'ensemble du groupe les droits de mise à jour du répertoire et de tous les fichiers qu'il contient.  
`chmod -R 775 /var/rdz/projects/`

## Chapitre 12. Remarques à propos de WLM

Contrairement à des applications z/OS traditionnelles, Developer for System z n'est pas une application monolithique qui peut être identifiée facilement au niveau du Workload Manager (WLM). Les différents composants de Developer for System z interagissent pour offrir au client un accès à des services et des données d'hôte. Comme décrit dans Chapitre 11, «Compréhension de Developer for System z», à la page 193, certains de ces services sont actifs dans différents espaces adresse; ce qui se traduit par différentes classifications WLM.

Les rubriques suivantes sont traitées dans le présent chapitre :

- «Classification des charges de travail»
- «Définition des objectifs», à la page 207

### Classification des charges de travail

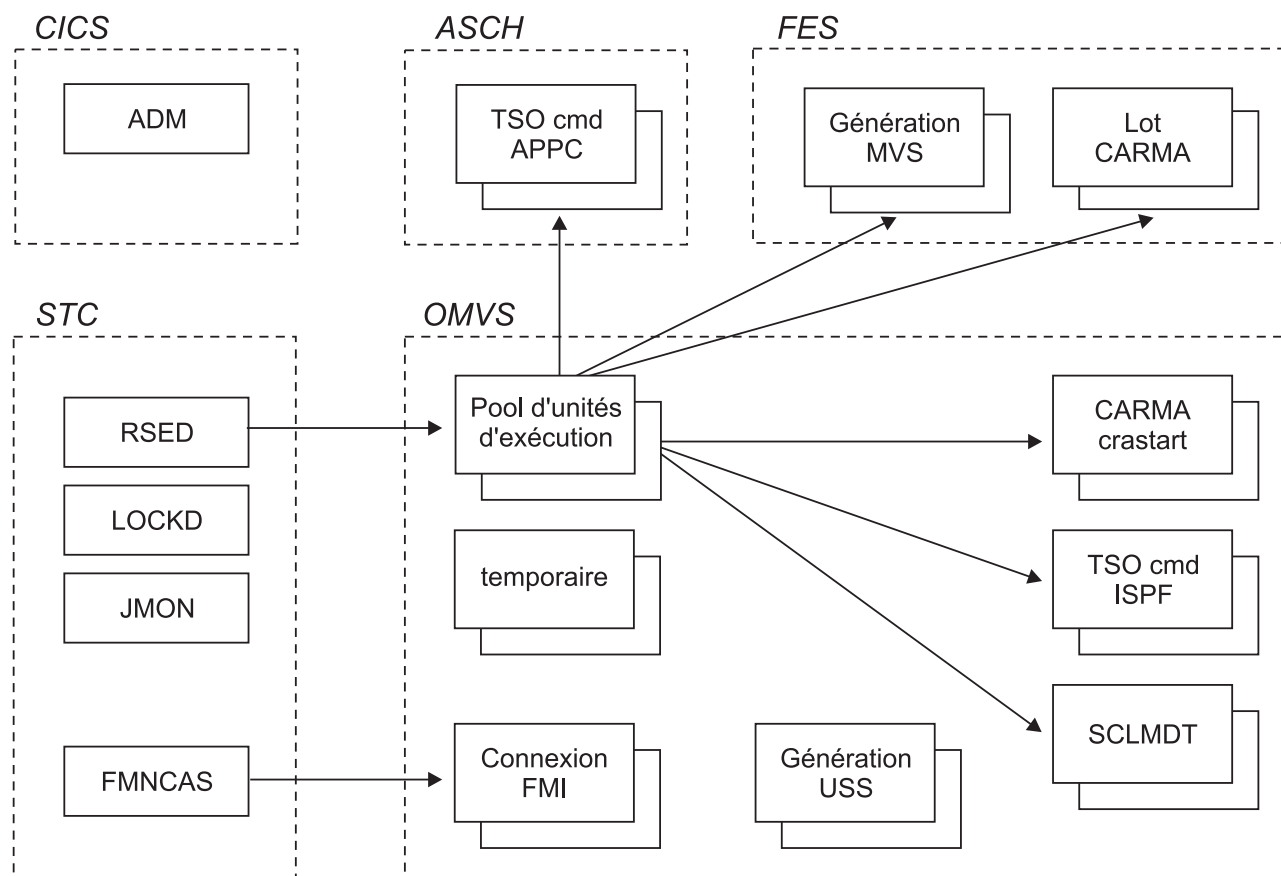


Figure 47. Classification WLM

La figure 47 affiche une présentation de base des sous-systèmes au moyen desquels des charges de travail de Developer for System z sont présentées à WLM.

Le gestionnaire de déploiement d'application (ADM) est actif au sein d'une région CICS, et il suivra donc suivra les règles de classification CICS dans WLM.

Le démon RSE (RSED), le démon Lock (LOCKD) et le moniteur de travaux JES (JMON) sont des tâches démarrées de Developer for System z (ou des travaux par lots à à exécution longue), chacun avec leur espace adresse individuel.

Comme nous l'avons documenté dans «RSE comme application Java», à la page 195, le démon RSE génère un processus enfant pour chaque serveur de pools d'unités d'exécution RSE (qui prend en charge un nombre variable de clients). Chaque pool d'unités d'exécution est actif dans un espace adresse distinct (à l'aide d'un initiateur z/OS UNIX, BPXAS). Puisqu'il s'agit de processus générés, leur classification s'effectue d'après les règles de classification WLM OMVS, mais pas selon les règles de classification des tâches démarrées.

Les clients qui sont actifs dans un pool d'unités d'exécution peuvent créer une multitude d'autres espaces adresse, selon les actions menées par les utilisateurs. Selon la configuration de Developer for System z, certaines charges de travail, comme un service de Commandes TSO (TSO cmd) ou CARMA, peuvent s'exécuter dans des sous-systèmes différents.

Les espaces adresse répertoriés dans la figure 47, à la page 205 restent dans le système suffisamment longtemps pour être visibles, vous devez cependant être conscient qu'en raison de la conception de z/OS UNIX, il existe aussi des espaces adresses temporaires de durée de vie courte. Ces espaces adresse temporaires sont actifs dans le sous-système OMVS.

Notez que tandis que les pools d'unités d'exécution utilisent le même ID utilisateur et un nom de travail similaire au démon RSE, tous les espaces adresse démarrés par un pool d'unités d'exécution appartiennent à l'ID utilisateur du client ayant demandé l'action. L'ID utilisateur du client est aussi utilisé comme (une partie de) le nom de travail pour tous les espaces adresse basés sur OMVS et déclarés par le pool d'unités d'exécution.

D'autres services utilisés par Developer for System z, comme File Manager (FMNCAS) ou z/OS UNIX REXEC (génération USS) créent des espaces adresse.

## Règles de classification

WLM utilise des règles de classification pour mapper un travail entrant le système en une classe de service. Cette classification repose sur des qualificateurs de travaux. Le premier qualificateur (obligatoire) est le type de sous-système qui reçoit la demande de travail. Le tableau 29 répertorie les types de sous-systèmes qui peuvent recevoir des charges de travail de Developer for System z.

*Tableau 29. Sous-système du point d'entrée WLM*

Type de sous-système	Description du travail
ASCH	Les demandes de travaux incluent tous les programmes de transactions APPC planifiés par le planificateur de transactions APPC/MVS fourni par IBM, ASCH.
CICS	Les demandes de travaux incluent toutes les transactions traitées par CICS.
JES	Les demandes de travaux incluent tous les travaux initiés par JES2 ou JES3.
OMVS	Les demandes de travaux incluent un travail traité dans des espaces adresse enfant en parallèle à des services système z/OS UNIX.

Tableau 29. Sous-système du point d'entrée WLM (suite)

Type de sous-système	Description du travail
STC	Les demandes de travaux incluent tous les travaux initiés par les commandes START et MOUNT. STC inclut aussi des espaces adresse de composants système.

Le tableau 30 Le Tableau 2 répertorie des qualificateurs supplémentaires que vous pouvez utiliser pour attribuer un charge de travail à une classe de service spécifique. Pour plus d'informations sur les qualificateurs répertoriés, voir MVS Planning: Workload Management (SA22-7602).

Tableau 30. Qualificateurs de travaux WLM

		ASCH	CICS	JES	OMVS	STC
AI	Comptabilité des informations	x		x	x	x
LU	Nom de l'unité logique (*)		x			
PF	Effectuer (*)			x		x
PRI	Priorité			x		
SE	Nom de l'environnement de planification			x		
SSC	Nom de collection du sous-système			x		
SI	Instance du sous-système (*)		x	x		
SPM	Paramètre du sous-système					x
PX	Nom Sysplex	x	x	x	x	x
SY	Nom du système (*)	x			x	x
TC	Classe Transaction/travail (*)	x		x		
TN	Nom Transaction/travail (*)	x	x	x	x	x
UI	ID utilisateur (*)	x	x	x	x	x

**Remarque :** S'agissant des qualificateurs marqués avec (\*), vous pouvez indiquer des groupes de classification en ajoutant un G à l'abréviation du type. Par exemple, un groupe de nom de transaction doit être TNG.

## Définition des objectifs

Comme nous l'avons documenté dans «Classification des charges de travail», à la page 205, Developer for System z crée différents types de charges de travail sur votre système. Ces différentes tâches communiquent entre elles, ce qui implique que le temps écoulé réel devienne important pour éviter des problèmes de délai d'attente lors des connexions entre les tâches. En conséquence, une tâche Developer for System z doit être placée dans des classes de services de hautes performances avec une priorité élevée.

Une révision, et probablement une mise à jour, de vos objectifs WLM actuels est donc recommandée, notamment s'il agit de charges de travail OMVS critique en temps ou nouvelles des magasins MVS traditionnels.

**Remarque :**

- Les informations d'objectif qui figurent dans cette section sont délibérément maintenues à un niveau descriptif, car les objectifs de performances réels sont très spécifiques du site.
- Pour mieux comprendre l'impact d'une tâche spécifique sur votre système, nous employons des termes comme utilisation de ressources minimale, modérée et substantielle. Tous ces termes sont relatifs à l'utilisation de la totalité des ressources de Developer for System z proprement dit, et non du système dans son intégralité.

Le tableau 31 répertorie les espaces adresse utilisés par Developer for System z. z/OS UNIX substituera le "x" dans la colonne "Nom de la tâche" par un nombre aléatoire de 1 caractère.

*Tableau 31. Charges de travail WLM*

Description	Nom de la tâche	Charge de travail
Moniteur de travaux JES	JMON	STC
Démon lock	LOCKD	STC
Démon RSE	RSED	STC
Pool d'unités d'exécution RSE	RSEDx	OMVS
Passerelle client ISPF (service Commandes TSO et SCLMDT)	<ID utilisateur>x	OMVS
Service Commandes TSO (APPC)	FEKFRSRV	ASCH
CARMA (lot)	CRA<port>	JES
CARMA (crastart)	<ID utilisateur>x	OMVS
CARMA (passerelle client ISPF)	<userid> et <userid>x	OMVS
Génération MVS (travail par lots)	*	JES
Génération z/OS UNIX (commandes shell)	<ID utilisateur>x	OMVS
Shell z/OS UNIX	<ID utilisateur>	OMVS
Tâche File Manager	<ID utilisateur>x	OMVS
Gestionnaire de déploiement d'application	CICSTS	CICS

## Remarques relatives à la sélection des objectifs

Les remarques générales suivantes relatives à WLM peuvent vous aider à bien définir les définitions d'objectifs correctes pour Developer for System z :

- Vous devez baser des objectifs sur ce qui peut être réellement obtenu, et non sur vos souhaits concernant ce qui pourrait arriver. Si vous définissez des supérieurs à ce qui est nécessaire, WLM déplace des ressources d'un travail de moindre importance vers un travail d'importance plus élevée qui pourrait ne pas avoir véritablement besoin des ressources.
- Limite le volume de travail attribué aux classes de service SYSTEM et SYSSTC, car ces classes bénéficient d'une priorité de distribution supérieure à n'importe quelle classe gérée WLM. Utilisez ces classes pour un travail qui est d'importance élevée, bien qu'utilisant peu d'unité centrale.
- Un travail qui n'entre pas dans les règles de classification finit par aboutir dans la classe SYSOTHER, qui a un objectif discrétionnaire. Un objectif discrétionnaire recommande à WLM d'agir au mieux lorsque le système a des ressources disponibles.

Lors de l'utilisation des objectifs de temps de réponse :

- Il doit exister un taux d'arrivée stable de tâches (au moins 10 tâches en 20 minutes) pour permettre à WLM de gérer correctement un objectif de temps de réponse.
- Utilisez des objectifs de temps de réponse moyen uniquement pour bien contrôler des charges de travail, car une transaction longue et unique a un impact énorme sur le temps de réponse moyen et peut contraindre WLM à réagir de façon excessive.

Lors de l'utilisation des objectifs de vitesse :

- D'une manière générale, vous ne pouvez pas obtenir d'objectif de vitesse au-delà de 90 % et ce pour différentes raisons. Par exemple, tous les espaces adresse SYSTEM et SYSSTC bénéficient d'une priorité de distribution supérieure à tout objectif de type vitesse.
- WLM utilise un nombre minimum de modèles (utilisation et délai) sur lesquels se fondent ses décisions en termes d'objectifs de vitesse. Ainsi, moins il y aura de travaux exécutés dans une classe de service, plus cela prendra de temps pour collecter le nombre requis de modèles et ajuster la règle de répartition.
- Réévaluez les objectifs de vitesse lors du changement de votre matériel. Notamment, vers une diminution, des processeurs plus rapides imposent des changements dans les objectifs de vitesse.

## STC

Toutes les tâches démarrées de Developer for System z, démon RSE, démon Lock et moniteur de travaux JES, répondent à des demandes client en temps réel.

Tableau 32. Charges de travail et STC WLM

Description	Nom de la tâche	Charge de travail
Moniteur de travaux JES	JMON	STC
Démon lock	LOCKD	STC
Démon RSE	RSED	STC

- Moniteur de travaux JES

Le moniteur de travaux JES fournit tous les services liés à JES comme des soumissions de travaux, la consultation de fichiers spoule et l'exécution de commande de l'opérateur JES. Vous devez indiquer un objectif de vitesse et de hautes performances sur une période, car la tâche ne signale pas les transactions individuelles à WLM. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type minimal à modéré.

- Démon lock

Le démon Lock interroge les tables de mise en file d'attente GRS (sérialisation d'accès des ressources partagées) lors d'une demande du client ou de l'opérateur et fait correspondre le résultat par rapport à des utilisateurs Developer for System z connus. Vous devez indiquer un objectif de vitesse et de hautes performances sur une période, car la tâche ne signale pas les transactions individuelles à WLM. On s'attend à une utilisation des ressources de type minimal.

- Démon RSE

Le démon RSE gère les connexions et authentification des clients ainsi que les différents pools d'unités d'exécution RSE. Vous devez indiquer un objectif de vitesse et de hautes performances sur une période, car la tâche ne signale pas les

transactions individuelles à WLM. On s'attend à une utilisation des ressources de type modéré, avec un pic au début de la journée de travail.

## OMVS

Les charges de travail OMVS peuvent être réparties en deux groupes, les pool d'unités d'exécution RSE et tout le reste. Ceci parce qu'à l'exception des pools d'unités d'exécution, toutes les charges de travail utilisent l'ID utilisateur du client comme base pour le nom de l'espace adresse. (z/OS UNIX substituera le "x" de la colonne "Nom de la tâche" par un nombre aléatoire de 1 caractère.)

*Tableau 33. Charges de travail - OMVS WLM*

Description	Nom de la tâche	Charge de travail
Pool d'unités d'exécution RSE	RSEDx	OMVS
Passerelle client ISPF (service Commandes TSO et SCLMDT)	<ID utilisateur>x	OMVS
CARMA (crastart)	<ID utilisateur>x	OMVS
CARMA (passerelle client ISPF)	<userid> et <userid>x	OMVS
Génération z/OS UNIX (commandes shell)	<ID utilisateur>x	OMVS
Shell z/OS UNIX	<ID utilisateur>	OMVS
Tâche File Manager	<ID utilisateur>x	OMVS

- Pool d'unités d'exécution RSE

Un pool d'unités d'exécution RSE est comme le coeur et le cerveau de Developer for System z. Presque toutes les données passent par là, tandis que les mineures (unités d'exécution spécifiques de l'utilisateur) à l'intérieur du pool d'unités d'exécution contrôlent les actions de la plupart des autres tâches liées à Developer for System z. Vous devez indiquer un objectif de vitesse et de hautes performances sur une période, car la tâche ne signale pas les transactions individuelles à WLM. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type substantiel.

Le charges de travail restantes finiront toutes par aboutir dans la même classe de service en raison d'une convention commune d'attribution de nom d'espace adresse. Vous devez indiquer un objectif à périodes multiples pour cette classe de service. Les premières périodes doivent être des objectifs de temps de réponse percentiles à hautes performances, tandis que la dernière période doit avoir un objectif de vitesse à performances modérées. Certaines charges de travail, comme une passerelle client ISPF, signaleront des transactions individuelles et d'autres non.

- Passerelle client ISPF

La passerelle client ISPF est un service ISPF appelé par Developer for System z pour exécuter des commandes TSO et ISPF non-interactives. Ceci inclut des commandes explicites émises par le client ainsi que des commandes implicites émises par Developer for System z, comme l'obtention d'une liste de membres PDS. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type minimal.

- CARMA



CARMA est un serveur Developer for System z facultatif qui permet d'interagir avec des gestionnaires de configuration logicielle (SCM) basés sur l'hôte, comme CA Endevor SCM. Developer for System z autorise différentes méthodes de démarrage pour un serveur CARMA, dont certaines deviennent une charge de travail OMVS. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type minimal.

- Génération z/OS UNIX

Lorsqu'un client initie une génération pour un projet z/OS UNIX, z/OS UNIX REXEC (ou SSH) doit démarrer une tâche qui exécute plusieurs commandes shell z/OS UNIX shell pour effectuer la génération. L'utilisation des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de modéré à substantiel, selon la taille du projet.

- Shell z/OS UNIX

Cette charge de travail traite des commande shell z/OS UNIX émises par le client. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type minimal.

- IBM File Manager

Bien qu'il ne s'agisse pas d'espace adresse dans Developer for System z, les processus enfant File Manager générés sont répertoriés ici car ils peuvent être démarrés sur la demande d'un client Developer for System z client ; ces tâches utilisent la même convention d'attribution de nom que des tâches Developer for System z. Ces tâches File Manager traitent des actions de fichiers MVS complexes, comme une édition formatée d'un fichier VSAM. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type minimal à modéré.

## JES

Developer for System z utilise les processus de traitement par lots gérés par JES de différentes manières. L'usage le plus classique concerne les générations MVS dans lesquelles un travail est soumis et contrôlé pour déterminer quand il prend fin. Toutefois, Developer for System z pourrait aussi démarrer un serveur CARMA dans un traitement par lots et communiquer avec celui-ci via TCP/IP.

Tableau 34. Charge de travail - JES WLM

Description	Nom de la tâche	Charge de travail
CARMA (lot)	CRA<port>	JES
Génération MVS (travail par lots)	*	JES

- CARMA

CARMA est un serveur Developer for System z facultatif qui permet d'interagir avec des gestionnaires de configuration logicielle (SCM) basés sur l'hôte, comme CA Endevor SCM. Developer for System z autorise différentes méthodes de démarrage pour un serveur CARMA, dont certaines deviennent une charge de travail JES. Vous devez indiquer un objectif de vitesse et de hautes performances sur une période, car la tâche ne signale pas les transactions individuelles à WLM. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type minimal.

- Génération MVS

Lorsqu'un client initie une génération pour un projet MVS, Developer for System z doit démarrer une tâche en traitement par lots pour effectuer la génération. L'utilisation des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de modéré à substantiel, selon la taille du

projet. Différentes stratégies d'objectifs à performances modérées peuvent être recommandées, selon des circonstances locales.

- Vous pourriez indiquer un objectif à périodes multiples avec une période à objectif de temps de réponse percentile et une période à objectif de vitesse secondaire. Dans ce cas, vos développeurs doivent utiliser pour la plupart la même procédure de génération et des fichiers d'entrée de tailles similaires pour créer des travaux ayant des temps de réponse uniformes. Il doit aussi exister un taux d'arrivée stable de travaux (au moins 10 travaux en 20 minutes) pour permettre à WLM de gérer correctement un objectif de temps de réponse.
- Un objectif de vitesse est plus approprié à la plupart des travaux en traitement par lots, car ces objectifs peuvent gérer des temps d'exécution et des taux d'arrivée extrêmement variables.

## ASCH

Dans les versions actuelles de Developer for System z, la passerelle client ISPF permet l'exécution de commandes TSO et ISPF non-interactive. Pour des raisons historiques, Developer for System z prend également en charge l'exécution de ces commandes via une transaction APPC.

Tableau 35. Charges de travail - ASCH WLM

Description	Nom de la tâche	Charge de travail
Service Commandes TSO (APPC)	FEKFRSRV	ASCH

- Service Commandes TSO

Developer for System z peut démarrer le service Commandes TSO peut être démarré comme une transaction APPC pour exécuter des commandes TSO et ISPF non-interactive. Ceci inclut des commandes explicites émises par le client ainsi que des commandes implicites émises par Developer for System z, comme l'obtention d'une liste de membres PDS. Vous devez indiquer un objectif à périodes multiples pour cette classe de service. Pour les premières périodes, vous devez indiquer des objectifs de temps de réponse percentiles de hautes performances. Pour la dernière période, vous devez indiquer un objectif de vitesse à performances modérées. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais on s'attend à être de type minimal.

## CICS

Le gestionnaire de déploiement d'application est un serveur Developer for System z facultatif qui est actif au sein d'une région CICS Transaction Server.

Tableau 36. Charges de travail - CICS WLM

Description	Nom de la tâche	Charge de travail
Gestionnaire de déploiement d'application	CICSTS	CICS

- Gestionnaire de déploiement d'application

Le serveur facultatif du gestionnaire de déploiement d'application qui est actif au sein d'une région CICSTS, vous permet de décharger en toute sécurité des tâches de gestion CICSTS pour les développeurs de logiciel. L'usage des ressources dépend fortement des actions des utilisateurs et donc fluctuera, mais

on s'attend à être de type minimal. Le type de classe de service que vous devez utiliser dépend des autres transactions qui sont actives dans cette région CICS, et ne sont donc pas abordées en détails.

WLM prend en charge plusieurs type de gestion que vous pouvez utiliser pour CICS :

- Gestion de CICS avec un objectif de région

L'objectif est défini sur une classe de service qui gère des espaces adresse CICS. Vous ne pouvez utiliser qu'un objectif de vitesse d'exécution pour cette classe de service. WLM utilise les règles de classification JES ou STC pour les espaces adresse, mais n'utilise pas les règles de classification des sous-systèmes pour des transactions.

- Gestion de CICS avec un objectif de temps de réponse de transaction

Vous pouvez définir un objectif de temps de réponse dans une classe de service attribuée à une transaction unique ou à un groupe de transactions. WLM utilise les règles de classification JES ou STC pour les espaces adresse et les règles de classification des sous-systèmes CICS pour des transactions.



---

## Chapitre 13. Remarques relatives à l'optimisation

Comme indiqué dans le Chapitre 11, «Compréhension de Developer for System z», à la page 193, RSE (Remote Systems Explorer) est le coeur de Developer for System z. Pour gérer les connexions et charges de travail provenant des clients, RSE est composé d'un espace adresse de démon, qui permet de contrôler les espaces adresse du groupe d'unités d'exécution. Le démon agit comme un point focal pour la connexion et la gestion, alors que les pools d'unités d'exécution traitent les charges de travail du client.

RSE devient une cible privilégiée d'optimisation de la configuration de Developer for System z. Toutefois, la gestion de centaines d'utilisateurs, chacun utilisant au moins 16 unités d'exécution, d'une certaine quantité de mémoire et éventuellement d'un ou de plusieurs espaces adresses implique de configurer correctement Developer for System z et z/OS.

Les rubriques suivantes sont traitées dans le présent chapitre :

- «Utilisation des ressources»
- «Utilisation de l'espace de stockage», à la page 226
- «Utilisation de l'espace du système de fichiers z/OS UNIX», à la page 232
- «Définitions de ressources essentielles», à la page 235
- «Définitions de ressource différentes», à la page 239
- «Contrôle», à la page 240
- «Exemple de configuration», à la page 244

---

### Utilisation des ressources

Utilisez les informations présentées dans cette section pour estimer l'utilisation normale et optimale des ressources par Developer for System z, de manière à pouvoir planifier la configuration du système en conséquence.

Lors de l'utilisation des nombres et formules présentés dans cette section pour définir les valeurs des limites du système, n'oubliez pas que vous utilisez des estimations assez précises. Lors de la définition des limites du système, prévoyez une marge suffisante afin de permettre aux tâches temporaires, aux autres tâches ou aux utilisateurs se connectant plusieurs fois à l'hôte simultanément d'utiliser les ressources (au moyen, par exemple, de RSE et TN3270).

#### Remarque :

- Les informations sont limitées aux services accessibles par l'intermédiaire de RSE qui sont fournis par Developer for System z lui-même. Par exemple, l'utilisation des ressources de TN3270 (inaccessibles par l'intermédiaire de RSE) et des programmes appelés lors des générations à distance (basées sur l'hôte) des projets MVS ou z/OS UNIX (non fournis par Developer for System z) n'est pas documentée.
- Tout ajout d'extensions tierces à Developer for System z peut augmenter les compteurs d'utilisation des ressources.
- Tous les services comportent des tâches de "nettoyage" de courte durée, qui utilisent les ressources pendant leur exécution, et qui

peuvent s'exécuter de manière séquentielle ou parallèle les unes par rapport aux autres. Les ressources utilisées par ces tâches ne sont pas documentées.

- Lorsque cela s'avère utile, l'utilisation des ressources propres à l'utilisateur des logiciels requis (ISPF Client Gateway, par exemple) est documentée.
- Les nombres indiqués ici peuvent changer sans préavis.

## Présentation

Les tableaux ci-dessous présentent les nombres d'espaces adresses, de processus et d'unités d'exécution utilisés par Developer for System z. Plus d'informations sur les nombres présentés ici sont disponibles dans les sections suivantes :

- «Nombre d'espaces adresses», à la page 217
- «Nombre de processus», à la page 220
- «Nombre d'unités d'exécution», à la page 223

Le tableau 37 donne une présentation générale des ressources essentielles utilisées par les tâches démarrées Developer for System z. Ces ressources ne sont attribuées qu'une seule fois. Elles sont partagées par tous les clients Developer for System z.

*Tableau 37. Utilisation des ressources communes*

Tâche démarrée	Espaces adresses	Processus	Unités d'exécution
JMON	1	1	3
LOCKD	1	3	10
RSED	1	3	11
RSEDx	(a)	2	10

**Remarque :** (a) Il existe au moins 1 espace adresse de pool d'unités d'exécution RSE actif. Voir «Nombre d'espaces adresses», à la page 217 pour déterminer le nombre réel d'espaces adresses de pool d'unités d'exécution RSE.

Le tableau 38 donne une présentation générale des ressources essentielles utilisées par les logiciels requis. Ces ressources sont attribuées pour chaque client Developer for System z qui appelle la fonction associée.

*Tableau 38. Utilisation des ressources requises propres à l'utilisateur*

Logiciels requis	Espaces adresses	Processus	Unités d'exécution
Passerelle client ISPF	1	2	4
APPC	1	1	2
Gestionnaire de fichiers	1	1	2

Le tableau 39, à la page 217 donne une présentation générale des ressources essentielles utilisées par chaque client Developer for System z lors de l'exécution de la fonction spécifiée. Les valeurs non numériques (ISPF, par exemple) font référence à la valeur correspondante du tableau 38.

Tableau 39. Utilisation des ressources propres à l'utilisateur

Action de l'utilisateur	Espaces adresses	Processus	Unités d'exécution		
	ID utilisateur	ID utilisateur	ID utilisateur	RSEDx	JMON
Connexion	-	-	-	16	1
Temporisateur pour le délai d'inactivité	-	-	-	1	-
Développement de PDS(E)	ISPF	ISPF	ISPF	-	-
Ouverture du fichier	ISPF	ISPF	ISPF	-	-
Commande TSO	ISPF	ISPF	ISPF	-	-
Interpréteur de commandes de z/OS UNIX	1	1	1	6	-
Génération MVS	1	-	-	-	-
Génération z/OS UNIX	3	3	3	-	-
CARMA (lot)	1	1	2	1	-
CARMA (crastart)	1	1	2	4	-
CARMA (ispf)	4	4	7	5	-
SCLMDT	ISPF	ISPF	ISPF	-	-
File Manager Integration	ISPF + FM	ISPF + FM	ISPF + FM	-	-
Fault Analyzer Integration	-	-	-	-	-

**Remarque :** ISPF peut être remplacé par APPC, sauf pour SCLM Developer Toolkit.

## Nombre d'espaces adresses

Le tableau 40 répertorie les espaces adresses utilisés par Developer for System z, où "u" de la colonne "Nombre" indique que la quantité doit être multipliée par le nombre d'utilisateurs actifs simultanés de la fonction. z/OS UNIX remplace "x" de la colonne "Nom de la tâche" par un numéro à un seul chiffre aléatoire.

Tableau 40. Nombre d'espaces adresses

Nombre	Description	Nom de la tâche	Partagé	Se termine après
1	Moniteur de travaux JES	JMON	Oui	Jamais
1	Démon lock	LOCKD	Oui	Jamais

Tableau 40. Nombre d'espaces adresses (suite)

Nombre	Description	Nom de la tâche	Partagé	Se termine après
1	Démon RSE	RSED	Oui	Jamais
(a)	Pool d'unités d'exécution RSE	RSEDx	Oui	Jamais
1u	Passerelle client ISPF (service Commandes TSO et SCLMDT)	<ID utilisateur>x	Non	15 minutes ou déconnexion de l'utilisateur
1u	Service Commandes TSO (APPC)	FEKFRSRV	Non	60 minutes ou déconnexion de l'utilisateur
1u	CARMA (lot)	CRA<port>	Non	7 minutes ou déconnexion de l'utilisateur
1u	CARMA (crastart)	<ID utilisateur>x	Non	7 minutes ou déconnexion de l'utilisateur
4u	CARMA (ispf)	(1)<ID utilisateur> ou (3)<ID utilisateur>x	Non	7 minutes ou déconnexion de l'utilisateur
(b)	Utilisation de la passerelle client ISPF simultanée par 1 utilisateur	<ID utilisateur>x	Non	Fin de la tâche
1u	Génération MVS (travail par lots)	*	Non	Fin de la tâche
3u	Génération z/OS UNIX (commandes shell)	<ID utilisateur>x	Non	Fin de la tâche
1u	Interpréteur de commandes de z/OS UNIX	<ID utilisateur>	Non	Déconnexion de l'utilisateur
(c)	Gestionnaire de fichiers	<ID utilisateur>x	Non	Fin de la tâche

**Remarque :**

- (a) Il existe au moins un espace adresse de pool d'unités d'exécution RSE actif. Le nombre réel dépend :
  - de la directive `minimum.threadpool.process` de `rsed.envvars`. La valeur par défaut est 1.
  - du nombre d'utilisateurs qu'un pool d'unités d'exécution peut gérer. Les paramètres par défaut autorisent 60 utilisateurs par pool d'unités d'exécution.
  - de la cote d'alerte haute des utilisateurs actifs simultanés, car les pools d'unités d'exécution en veille ne s'arrêtent pas automatiquement.
- (b) Developer for System z comporte plusieurs unités d'exécution actives par utilisateur. Si l'espace adresse de la passerelle client ISPF n'a pas terminé de gérer la demande d'une unité d'exécution lorsqu'une autre unité d'exécution envoie une requête, ISPF demande à une nouvelle passerelle client de traiter la nouvelle requête. Cet espace adresse se termine à l'issue de la tâche.
- (c) Le programme d'écoute File Manager démarre une espace adresse par objet à manipuler (la méthode d'accès VSAM, par exemple). Cet espace adresse reste actif tant que Developer for System z ne signale pas que l'objet n'est plus nécessaire (en fermant la méthode d'accès VSAM, par exemple).
- SCLMDT requiert un espace adresse de passerelle client ISPF. SCLMDT partage l'espace adresse avec le service Commandes TSO.



- La plupart des actions liées au fichier MVS utilisent le service Commandes TSO, qui peut être actif dans la passerelle de client ISPF ou une transaction APPC, respectivement.

Utilisez la formule de la figure 48 pour estimer le nombre maximal d'espaces adresses utilisés par Developer for System z.

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

Figure 48. Nombre maximal d'espaces adresses

Où

- “3” correspond au nombre d'espaces adresses de serveur actif permanent.
- “A” représente le nombre d'espaces adresses de pool d'unités d'exécution RSE.
- “N” représente le nombre maximal d'utilisateurs simultanés.
- “x” est l'une des valeurs suivantes, selon les options de configuration sélectionnées.

X	SCLMDT	TSO au moyen de la passerelle client	TSO au moyen d'APPC
1	Non	Non	Oui
1	Non	Oui	Non
1	Oui	Oui	Non

- “y” est l'une des valeurs suivantes, selon les options de configuration sélectionnées.

Y	
0	Pas de CARMA
1	CARMA (lot)
1	CARMA (crastart)
4	CARMA (ispf)

- Par défaut, la valeur de “z” est 0, mais elle peut augmenter selon les actions de l'utilisateur :
  - Ajoutez 1 lorsqu'une génération MVS est réalisée. Ces espaces adresses se terminent lorsque la tâche de génération connexe (un travail par lots) se termine.
  - Ajoutez 3 lorsqu'une génération z/OS UNIX est réalisée. Notez que le nombre réel peut être plus élevé, en fonction des besoins des programmes appelés. Ces espaces adresses se terminent à l'issue de la tâche de génération connexe.
  - Ajoutez 1 pour chaque interaction simultanée avec IBM File Manager. Ces espaces adresses se terminent lorsque l'objet demandé n'est plus nécessaire.
- “2 + N\*0.01” permet d'ajouter une mémoire tampon aux espaces adresses temporaires. La taille de mémoire tampon requise peut différer en fonction du site.

Utilisez la formule de la figure 49, à la page 220 pour estimer le nombre maximal d'espaces adresses utilisés par le client Developer for System z (sans compter les espaces adresses temporaires non documentés).

$$x + y + z$$

Figure 49. Nombre d'espaces adresses par client

Où

- "x" dépend des options de configuration sélectionnées et est intégré dans la formule afin de calculer le nombre maximal d'espaces adresses (figure 48, à la page 219).
- "y" dépend des options de configuration sélectionnées et est intégré dans la formule afin de calculer le nombre maximal d'espaces adresses (figure 48, à la page 219).
- Par défaut, la valeur de "z" est 0, mais elle peut augmenter selon les actions de l'utilisateur, comme indiqué dans la formule permettant de calculer le nombre maximal d'espaces adresses (figure 48, à la page 219).

Les définitions du tableau 41 peuvent limiter le nombre réel d'espaces adresses.

Tableau 41. Limites d'espace adresse

Adresse	Limite	Ressources affectées
rsed.envvars	maximum.threadpool.process	Limite le nombre de pools d'unités d'exécution RSE
IEASYMxx	MAXUSER	Limite le nombre d'espaces adresses
ASCHPMxx	MAX	Limite le nombre de demandeurs APPC pour le service Commandes TSO (APPC)

## Nombre de processus

Le tableau 42 répertorie le nombre de processus par espace adresse utilisés par Developer for System z. où "u" de la colonne "Espaces adresses" indique que la quantité doit être multipliée par le nombre d'utilisateurs actifs simultanés de la fonction.

Tableau 42. Nombre de processus

Processus	Espaces adresses	Description	ID utilisateur
1	1	Moniteur de travaux JES	STCJMON
3	1	Démon lock	STCLOCK
3	1	Démon RSE	STCRSE
2	(a)	Pool d'unités d'exécution RSE	STCRSE
2	(b)	Passerelle client ISPF (service Commandes TSO et SCLMDT)	<ID utilisateur>
1	1u	Service Commandes TSO (APPC)	<ID utilisateur>
1	1u	CARMA (lot)	<ID utilisateur>
1	1u	CARMA (crastart)	<ID utilisateur>
1	1u	CARMA (ispf)	<ID utilisateur>

Tableau 42. Nombre de processus (suite)

Processus	Espaces adresses	Description	ID utilisateur
1	3u	Génération z/OS UNIX (commandes shell)	<ID utilisateur>
1	1u	Interpréteur de commandes de z/OS UNIX	<ID utilisateur>
1	(c)	Gestionnaire de fichiers	<ID utilisateur>
(5)	(u)	SCLM Developer Toolkit	<ID utilisateur>

**Remarque :**

- (a) Il existe au moins 1 espace adresse de pool d'unités d'exécution RSE actif. Voir «Nombre d'espaces adresses», à la page 217 pour déterminer le nombre réel d'espaces adresses de pool d'unités d'exécution RSE.
- Le démon RSE et tous les pools d'unités d'exécution RSE utilisent le même ID utilisateur.
- (b) Dans des situations normales, et lorsque les options de configuration par défaut sont utilisées, il y a une passerelle client ISPF active par utilisateur. Le nombre réel peut varier (voir «Nombre d'espaces adresses», à la page 217).
- (c) Le programme d'écoute File Manager utilise un processus par objet à manipuler (la méthode d'accès VSAM, par exemple). Ce processus reste actif tant que Developer for System z ne signale pas que l'objet n'est plus nécessaire (en fermant la méthode d'accès VSAM, par exemple).
- SCLMDT requiert un espace adresse de passerelle client ISPF. SCLMDT partage l'espace adresse avec le service Commandes TSO.
- (u) Les processus SCLMDT s'exécutent dans l'espace adresse de la passerelle client ISPF. Par conséquent, elle ne porte aucune valeur de comptage d'espace adresse.
- Les processus SCLMDT sont temporaires et se terminent à l'issue de la tâche, mais plusieurs processus peuvent être actifs simultanément pour un seul utilisateur. Le tableau 42, à la page 220 répertorie le nombre maximal de processus SCLMDT simultanés.
- La plupart des actions liées au fichier MVS utilisent le service Commandes TSO, qui peut être actif dans la passerelle de client ISPF ou une transaction APPC, respectivement.
- Une génération z/OS UNIX utilise trois processus au total, chacun s'exécutant dans leur propre espace adresse.
- Tous les processus répertoriés restent actifs tant que l'espace adresse associé n'est pas terminé, sauf indication contraire.

Utilisez la formule de la figure 50, à la page 222 pour estimer le nombre maximal de processus utilisés par Developer for System z.

$$7 + 2 * A + N * (x + y + z) + (10 + N * 0.05)$$

Figure 50. Nombre maximal de processus

Où

- “7” correspond au nombre de processus utilisés par les espaces adresses de serveur actif permanent.
- “A” représente le nombre d'espaces adresses de pool d'unités d'exécution RSE.
- “N” représente le nombre maximal d'utilisateurs simultanés.
- “x” est l'une des valeurs suivantes, selon les options de configuration sélectionnées.

X	SCLMDT	TSO au moyen de la passerelle client	TSO au moyen d'APPC
1	Non	Non	Oui
2	Non	Oui	Non
7	Oui	Oui	Non

- “y” est l'une des valeurs suivantes, selon les options de configuration sélectionnées.

Y	
0	Pas de CARMA
1	CARMA (lot)
1	CARMA (crastart)
4	CARMA (ispf)

- Par défaut, la valeur de “z” est 0, mais elle peut augmenter selon les actions de l'utilisateur :
  - Ajoutez 1 lorsqu'un interpréteur de commandes z/OS UNIX est ouvert. Ce processus reste actif tant que l'utilisateur ne se déconnecte pas.
  - Ajoutez 3 lorsqu'une génération z/OS UNIX est réalisée. Notez que le nombre réel peut être plus élevé, en fonction des besoins des programmes appelés. Ces processus se terminent à l'issue de la tâche de génération connexe.
  - Ajoutez 1 pour chaque interaction simultanée avec IBM File Manager. Ces processus se terminent lorsque l'objet demandé n'est plus nécessaire.
- “10 + N\*0.05” permet d'ajouter une mémoire tampon pour les processus temporaires. La taille de mémoire tampon requise peut différer en fonction du site.

Utilisez la formule de la figure 51 pour estimer le nombre maximal de processus utilisés par le client Developer for System z (sans compter les espaces adresses temporaires non documentés).

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

Figure 51. Nombre de processus par client

Où

- "x" dépend des options de configuration sélectionnées et est intégré dans la formule afin de calculer le nombre maximal de processus (figure 50, à la page 222).
- "y" dépend des options de configuration sélectionnées et est intégré dans la formule afin de calculer le nombre maximal de processus (figure 50, à la page 222).
- Par défaut, la valeur de "z" est 0, mais elle peut augmenter selon les actions de l'utilisateur, comme indiqué dans la formule permettant de calculer le nombre maximal de processus (figure 50, à la page 222).
- "s" est égal à 1 si SCLM Developer Toolkit est utilisé. Il est égal à 0 dans le cas contraire.

Les définitions du tableau 43 peuvent limiter le nombre réel de processus.

*Tableau 43. Limites de processus*

Adresse	Limite	Ressources affectées
BPXPRMxx	MAXPROCSYS	Limite le nombre de processus
BPXPRMxx	MAXPROCUSER	Limite le nombre de processus par UID z/OS UNIX

Remarque :

- Le démon RSE et les pools d'unités d'exécution RSE utilisent le même ID utilisateur. Etant donné que le démon RSE démarre un nouveau pool d'unités d'exécution à chaque fois que cela s'avère nécessaire, le nombre de processus associés à cet ID utilisateur peut augmenter. Par conséquent, la valeur attribuée à MAXPROCUSER doit permettre de s'adapter à cette augmentation, qui peut être formulée sous la forme "3 + 2\*A".
- La limite MAXPROCUSER est par ID utilisateur z/OS UNIX unique. Multipliez le nombre de processus par utilisateur estimé par le nombre de clients actifs simultanément si vos utilisateurs partagent le même UID.

## Nombre d'unités d'exécution

Le tableau 44, à la page 224 répertorie le nombre d'unités d'exécution utilisées par les fonctions sélectionnées de Developer for System z. La lettre "u" des colonnes "Unités d'exécution" indique que la quantité doit être multipliée par le nombre d'utilisateurs actifs simultanés de la fonction. Le nombre d'unités d'exécution est indiqué par processus, étant donné que les limites sont définies à ce niveau.

- RSEDx : ces unités d'exécution sont créées dans le pool d'unités d'exécution RSE, que plusieurs clients se partagent. Toutes les unités d'exécution se retrouvant dans le même pool d'unités d'exécution doivent être ajoutés pour obtenir le nombre total.
- Actif : ces unités d'exécution font partie intégrante du processus qui exécute la fonction demandée. Chaque processus est une unité autonome. Par conséquent, il n'est pas utile de faire la somme des unités d'exécution, même si elles sont associées au même ID utilisateur, sauf indication contraire.
- Amorce : les processus d'amorce sont indispensables au démarrage du processus réel. Chacun d'eux dispose d'une unité d'exécution, et il peut exister plusieurs amorces consécutives. Il n'est pas utile de faire la somme des unités d'exécution.

Tableau 44. Nombre d'unités d'exécution

Unités d'exécution			ID utilisateur	Description
RSEDx	Actif	Amorce		
-	3 + 1u	-	STCJMON	Moniteur de travaux JES
-	10	2	STCLOCK	Démon lock
-	11	2	STCRSE	Démon RSE
10 (a) + 16u	-	1 (a)	STCRSE	Pool d'unités d'exécution RSE
-	4u (b)	1u (b)	<ID utilisateur>	Passerelle client ISPF (service Commandes TSO et SCLMDT)
-	2u	-	<ID utilisateur>	Service Commandes TSO (APPC)
1u	2u	-	STCRSE et <ID utilisateur>	CARMA (lot)
4u	2u	-	STCRSE et <ID utilisateur>	CARMA (crastart)
5u	4u	3u	STCRSE et <ID utilisateur>	CARMA (ispf)
-	1u (d)	2u	<ID utilisateur>	Génération z/OS UNIX (commandes shell)
6u	1u	-	STCRSE et <ID utilisateur>	Interpréteur de commandes de z/OS UNIX
-	2u (c)	-	<ID utilisateur>	Gestionnaire de fichiers
-	(5)	-	<ID utilisateur>	SCLM Developer Toolkit
1u	-	-	STCRSE	Temporisateur pour le délai d'inactivité

**Remarque :**

- (a) Il existe au moins 1 espace adresse de pool d'unités d'exécution RSE actif. Voir «Nombre d'espaces adresses», à la page 217 pour déterminer le nombre réel d'espaces adresses de pool d'unités d'exécution RSE.
- (b) Dans des situations normales, et lorsque les options de configuration par défaut sont utilisées, il y a une passerelle client ISPF active par utilisateur. Le nombre réel peut varier (voir «Nombre d'espaces adresses», à la page 217).

- (c) Il existe un processus propre à l'utilisateur (avec le nombre d'unités d'exécution indiqué) par interaction avec IBM File Manager. Ces processus se terminent lorsque l'objet demandé n'est plus nécessaire.
- SCLMDT requiert un espace adresse de passerelle client ISPF. SCLMDT partage l'espace adresse avec le service Commandes TSO.
- Selon l'action sélectionnée, SCLMDT peut utiliser plusieurs processus à une seule unité d'exécution se terminant à l'issue de la tâche. Le tableau 44, à la page 224 répertorie le nombre maximal d'unités d'exécution SCLMDT simultanées.
- La plupart des actions liées au fichier MVS utilisent le service Commandes TSO, qui peut être actif dans la passerelle de client ISPF ou une transaction APPC, respectivement.
- (d) Une génération z/OS UNIX appelle différents utilitaires de génération, qui peuvent comporter plusieurs unités d'exécutions. Le tableau 44, à la page 224 répertorie le nombre minimal d'unités d'exécution de génération z/OS UNIX simultanées.
- Toutes les unités d'exécution répertoriées restent actives tant que le processus associé n'est pas terminé, sauf indication contraire.

Utilisez la formule de la figure 52 pour estimer le nombre maximal d'unités d'exécution utilisées par un pool d'unités d'exécution RSE. Utilisez la formule de la figure 53 pour estimer le nombre maximal d'unités d'exécution utilisées par le moniteur de travaux JES.

$$9 + N*(16 + x + y + z) + (20 + N*0.1)$$

Figure 52. Nombre maximal d'unités d'exécution du pool d'unités d'exécution RSE

$$3 + N$$

Figure 53. Nombre maximal d'unités d'exécution du moniteur de travaux JES

Où

- "N" représente le nombre maximum d'utilisateurs concurrents dans ce pool d'unités d'exécution ou ce moniteur de travaux JES. Les paramètres par défaut autorisent 60 utilisateurs par pool d'unités d'exécution.
- "x" est l'une des valeurs suivantes, selon les options de configuration sélectionnées.

X	SCLMDT	TSO au moyen de la passerelle client	TSO au moyen d'APPC	Délai d'attente
0	Non	Non	Oui	Non
0	Non	Oui	Non	Non
0	Oui	Oui	Non	Non
1	Non	Non	Oui	Oui
1	Non	Oui	Non	Oui
1	Oui	Oui	Non	Oui

- “y” est l'une des valeurs suivantes, selon les options de configuration sélectionnées.

Y	
0	Pas de CARMA
1	CARMA (lot)
4	CARMA (crastart)
5	CARMA (ispf)

- Par défaut, la valeur de “z” est 0, mais elle peut augmenter selon les actions de l'utilisateur :
  - Ajoutez 6 lorsqu'un interpréteur de commandes z/OS UNIX est ouvert. Ces unités d'exécution restent actives tant que l'utilisateur ne se déconnecte pas.
- "20 + N\*0.1" permet d'ajouter une mémoire tampon pour des unités d'exécution temporaires. La taille de mémoire tampon requise peut différer en fonction du site.

Les définitions du tableau 45 peuvent limiter le nombre réel d'unités d'exécution d'un processus, qui est en général important pour les pools d'unités d'exécution RSE.

Tableau 45. Limites d'unités d'exécution

Adresse	Limite	Ressources affectées
BPXPRMxx	MAXTHREADS	Limite le nombre d'unités d'exécution d'un processus.
BPXPRMxx	MAXTHREADTASKS	Limite le nombre de tâches MVS d'un processus.
BPXPRMxx	MAXASSIZE	Limite la taille d'espace adresse, et donc la mémoire disponible pour les blocs de contrôle liés à l'unité d'exécution.
rsed.envvars	Xmx	Définit la taille de pile Java maximale. Cette mémoire est réservée. Elle n'est donc plus disponible pour les blocs de contrôle liés à l'unité d'exécution.
rsed.envvars	maximum.clients	Limite le nombre de clients (et donc leurs unités d'exécution) dans un pool d'unités d'exécution RSE.
rsed.envvars	maximum.threads	Limite le nombre d'unités d'exécution client dans un pool d'unités d'exécution RSE.
FEJCNFG	MAX_THREADS	Limite le nombre d'unités d'exécution dans le moniteur de travaux JES.

**Remarque :** La valeur de maximum.threads dans rsed.envvars doit être inférieure à celle de MAXTHREADS et MAXTHREADTASKS dans BPXPRMxx.

## Utilisation de l'espace de stockage

RSE est une application Java, ce qui signifie que l'utilisation de l'espace de stockage (mémoire) de Developer for System z doit s'appuyer sur deux limites d'allocation de mémoire : la taille de pile Java et la taille de l'espace adresse.



## Limite de la taille de pile Java

Java offre de nombreux services visant à faciliter le codage des applications Java. L'un de ces services est la gestion de l'espace de stockage.

La gestion de l'espace de stockage Java alloue des blocs d'espace de stockage volumineux et les utilise pour satisfaire les demandes d'espace de stockage de l'application. Cet espace de stockage géré par Java est appelé pile Java. La récupération régulière de place (défragmentation) permet de récupérer l'espace inutilisé dans la pile et de réduire sa taille.

La taille de pile Java maximale est définie dans `rsed.envvars` avec la directive `Xmx`. Si cette directive n'est pas spécifiée, Java utilise une taille par défaut de 64 Mo.

Chaque pool d'unités d'exécution RSE (qui gère les actions du client) est une application Java distincte et possède donc une pile Java personnelle. Notez que tous les pools d'unités d'exécution utilisent le même fichier de configuration `rsed.envvars` et donc la même limite de taille de pile Java.

L'utilisation du pool d'unités d'exécution de la pile Java dépend fortement des actions des clients connectés. Il est nécessaire de surveiller régulièrement l'utilisation de la pile pour définir la limite de taille de pile optimale. Utilisez la commande de l'opérateur **modify display process** pour surveiller l'utilisation de la pile Java par les pools d'unités d'exécution RSE.

## Limite de la taille d'espace adresse

Toutes les applications z/OS, y compris les applications Java, sont actives dans un espace adresse et sont donc liées par les limites de taille de l'espace adresse.

La taille d'espace adresse souhaitée est spécifiée au démarrage (avec le paramètre `REGION` de JCL, par exemple). Toutefois, les caractéristiques du système peuvent limiter la taille d'espace adresse réelle. Voir «Taille d'espace adresse», à la page 155 pour en savoir plus sur ces limites.

- `MAXASSIZE` de `SYS1.PARMLIB(BPXPRMxx)`
- `ASSIZEMAX` du segment `OMVS` de l'ID utilisateur attribué à la tâche démarrée
- sorties du système `IEFUSI` et `IEALIMIT`

Les pools d'unités d'exécution RSE héritent des limites de taille d'espace adresse provenant du démon RSE. La taille d'espace adresse doit être suffisante pour héberger la pile Java, Java lui-même, les zones de mémoire communes et tous les blocs de contrôle que le système crée pour prendre en charge l'activité du pool d'unités d'exécution (un bloc de contrôle des tâches par unité d'exécution, par exemple). Notez que certaines de ces utilisations de l'espace de stockage usage est inférieure à la ligne 16 Mo.

Il est recommandé de surveiller la taille réelle de l'espace adresse avant de modifier les paramètres qui l'influencent (la modification de la taille de la pile Java ou le nombre d'utilisateurs pris en charge par un seul pool d'unités d'exécution, par exemple). Utilisez les logiciels de surveillance du système pour suivre l'utilisation réelle de l'espace de stockage par Developer for system z. Si vous ne disposez pas de ce type d'outil, vous pouvez utiliser des outils comme la vue `SDSF DA` ou `TASID` (un outil d'informations système en l'état disponible sur la page Web ISPF "Support and downloads") afin de rassembler des informations de base.

## Instructions relatives à l'évaluation de la taille

Comme indiqué précédemment, l'utilisation réelle de l'espace de stockage par Developer for system z est fortement influencée par l'activité de l'utilisateur. Certaines actions utilisent une quantité fixe d'espace de stockage (connexion, par exemple), d'autres étant variables (liste des fichiers avec un qualificatif de haut niveau spécifié, par exemple).

- Utilisez un espace adresse de 2 Go pour RSE afin d'attribuer de l'espace pour la pile Java et tous les blocs de contrôle du système.
- L'exemple de configuration `rsed.envvars` autorise 60 utilisateurs par pool d'unités d'exécution.
  - `maximum.clients=60`
  - `maximum.threads=1000` ( $10+16*60 = 970$ , 1000 pour 61 clients)
- L'exemple de configuration `rsed.envvars` permet à la pile Java d'augmenter jusqu'à 256 Mo (60 clients utilisant en moyenne 4 Mo par client ( $60*4 = 240$ )).

Notez que le message de console FEK004I de RSE affiche la limite en cours de la taille de la pile Java et de l'espace adresse lors du démarrage.

Utilisez l'un des scénarios suivants si la surveillance montre que la taille de pile Java est insuffisante comparée à la charge de travail réelle :

- Augmentez la taille maximale de la pile Java avec la directive `Xmx` dans `rsed.envvars`. Mais auparavant, vérifiez que l'espace adresse est suffisant pour l'augmentation de la taille.
- Diminuez le nombre de clients par pool d'unités d'exécution avec la directive `maximum.clients` dans `rsed.envvars`. RSE prend toujours en charge le même nombre de clients, lesquels sont répartis entre plusieurs pools d'unités d'exécution.

## Exemple d'analyse de l'utilisation de l'espace de stockage

Les affichages des figures ci-dessous présentent des exemples de cas d'utilisation des ressources pour une configuration Developer for system z par défaut avec une modification. La taille maximale de la pile Java est de 10 Mo, une petite valeur maximale donnant lieu à un centile plus important d'utilisation et à des limites de taille de la pile atteintes plus tôt.

Max Heap Size=10MB and private AS Size=1,959MB

startup

BPXM023I (STCRSE)  
ProcessId(268 ) Memory Usage(7%) Clients(0)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2740	72
LOCKD	1.60	28.7M	14183
RSED	4.47	32.8M	15910
RSED8	1.15	27.4M	12612

logon 1

BPXM023I (STCRSE)  
ProcessId(268 ) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2864	81
LOCKD	1.64	28.8M	14259
RSED	4.55	32.8M	15980
RSED8	3.72	55.9M	24128

logon 2

BPXM023I (STCRSE)  
ProcessId(268 ) Memory Usage(23%) Clients(2)

Jobname	Cpu time	Storage	EXCP
JMON	0.02	2944	86
LOCKD	1.66	28.9M	14268
RSED	4.58	32.9M	16027
RSED8	4.20	57.8M	25205

logon 3

BPXM023I (STCRSE)  
ProcessId(268 ) Memory Usage(37%) Clients(3)

Jobname	Cpu time	Storage	EXCP
JMON	0.02	3020	91
LOCKD	1.67	29.0M	14277
RSED	4.60	32.9M	16076
RSED8	4.51	59.6M	26327

logon 4

BPXM023I (STCRSE)  
ProcessId(268 ) Memory Usage(41%) Clients(4)

Jobname	Cpu time	Storage	EXCP
JMON	0.02	3108	96
LOCKD	1.68	29.0M	14286
RSED	4.61	32.9M	16125
RSED8	4.77	62.3M	27404

Figure 54. Utilisation des ressources avec 5 connexions

logon 5

```
BPXM023I (STCRSE)
ProcessId(268      ) Memory Usage(41%) Clients(4)
ProcessId(33554706) Memory Usage(13%) Clients(1)
```

Jobname	Cpu time	Storage	EXCP
JMON	0.03	3184	101
LOCKD	1.69	29.1M	14295
RSED	4.64	32.9M	16229
RSED8	4.78	62.4M	27413
RSED9	4.60	56.6M	24065

Figure 55. Utilisation des ressources avec 5 connexions (suite)

La figure 54, à la page 229 et la figure 55 illustrent un scénario dans lequel 5 clients se connectent à un démon RSE avec un pile Java de 10 Mo.

- Un pool d'unités d'exécution (RSED8) est un état dormant au démarrage, utilisant environ 27 Mo, dont 0,7 Mo se trouvent dans la pile Java (7 % de 10 Mo).
- Le pool d'unités d'exécution devient actif lorsque le premier client se connecte, utilisant 27 Mo plus 2 Mo pour chaque client qui se connecte.
- Une partie de ces 2 Mo par connexion se trouve dans la pile Java, comme l'illustre l'augmentation de l'utilisation de la pile.
- Toutefois, il n'existe pas de modèle d'utilisation de la pile, car cela dépend des mécanismes Java qui évaluent l'espace requis et en allouent plus que nécessaire. La récupération intermittente de place permet de libérer de l'espace de stockage, ce qui rend les tendances plus difficiles à détecter.
- Les mécanismes internes qui limitent le nombre de connexions par pool d'unités d'exécution pour garantir une taille de pile suffisante pour les unités d'exécutions actives obligent la cinquième connexion à être créée dans un nouveau pool d'unités d'exécution (RSED9). En principe, ces réseaux de sécurité interne ne sont pas appelés lorsqu'une installation est correctement configurée, car d'autres limites seraient atteintes en premier (la plus probable étant `maximum.clients` de `rsed.envvars`).

Max Heap Size=10MB and private AS Size=1,959MB

startup

BPXM023I (STCRSE)  
ProcessId(212 ) Memory Usage(7%) Clients(0)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2736	71
LOCKD	1.73	30.5M	14179
RSED	4.35	32.9M	15117
RSED8	1.43	27.4M	12609

logon

BPXM023I (STCRSE)  
ProcessId(212 ) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2864	80
LOCKD	1.76	30.6M	14255
RSED	4.48	33.0M	15187
RSED8	3.53	53.9M	24125

expand large MVS tree (195 data sets)

BPXM023I (STCRSE)  
ProcessId(212 ) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
JMON	0.01	2864	80
LOCKD	1.78	30.6M	14255
RSED	4.58	33.1M	16094
RSED8	4.28	56.1M	24740

expand small PDS (21 members)

BPXM023I (STCRSE)  
ProcessId(212 ) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
IBMUSER2	0.22	2644	870
JMON	0.01	2864	80
LOCKD	1.78	30.6M	14255
RSED	4.61	33.1M	16108
RSED8	4.40	56.2M	24937

open medium sized member (86 lines)

BPXM023I (STCRSE)  
ProcessId(212 ) Memory Usage(13%) Clients(1)

Jobname	Cpu time	Storage	EXCP
IBMUSER2	0.22	2644	870
JMON	0.01	2864	80
RSED	4.61	33.1M	16108
RSED8	8.12	62.7M	27044

Figure 56. Utilisation des ressources lors de l'édition d'un membre PDS

La figure 56 illustre un scénario dans lequel un client se connecte au démon RSE avec un pile Java de 10 Mo, puis édite un membre PDS.

- La recherche de catalogue qui a généré 195 noms de fichier a utilisé environ 2 Mo d'espace de stockage, en raison de l'activité du système, car l'utilisation de la pile Java n'augmente pas.
- L'ouverture d'un fichier PDS de 21 membres utilise à peine la mémoire du pool d'unités d'exécution, mais l'affichage montre que le service Commandes TSO a été appelé. Un nouvel espace adresse est actif (IBMUUSER2) et utilise la taille de région attribuée à cet ID utilisateur dans TSO. Cet espace adresse reste actif pendant une durée spécifiée de manière à pouvoir être de nouveau utilisé pour de futures requêtes par le service Commandes TSO.
- L'ouverture d'un membre montre des numéros analogues au fur et à mesure du développement d'un qualificatif de haut niveau. L'utilisation de la pile Java ne change pas, mais l'espace de stockage augmente de 6,5 Mo en raison de l'activité du système.

---

## Utilisation de l'espace du système de fichiers z/OS UNIX

La plupart de données liées à Developer for System z qui ne sont pas écrites dans une instruction de définition de données sont placées dans un fichier z/OS UNIX. Le programmeur système peut décider des données écrites et de leur destination. Toutefois, il ne contrôle pas la quantité de données écrites.

Les données peuvent être regroupées dans les catégories suivantes :

- Analyse du problème (fichiers journaux et fichiers de vidage système), présentée en détails dans le Chapitre 9, «Identification des incidents liés à la configuration», à la page 139
- Contrôle (voir «Consignation dans le journal d'audit», à la page 167)
- Données temporaires

Developer for System z écrit les journaux de l'hôte associé à RSE dans les répertoires z/OS UNIX suivants (voir Chapitre 9, «Identification des incidents liés à la configuration», à la page 139) :

- /var/rdz/logs pour les fichiers journaux de la tâche démarrée RSE
- /var/rdz/logs/\$LOGNAME pour les journaux utilisateur

Par défaut, seuls les erreurs et messages d'avertissement sont consignés dans les fichiers journaux. Ainsi, si tout ce passe comme prévu, ces répertoires ne contiennent que des fichiers vides ou presque vides (sans compter les journaux d'audit).

Vous pouvez activer la consignation des messages d'information (de préférence avec l'aide du point de service IBM), ce qui augmente sensiblement la taille des fichiers journaux.

```

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

```

Figure 57. Utilisation de l'espace du système de fichiers z/OS UNIX

La figure 57 illustre l'utilisation minimale de l'espace du système de fichiers z/OS UNIX lors de l'utilisation du niveau de débogage 2 (messages d'information).

- Les journaux de la tâche démarrée utilisent 34 Ko après le démarrage et augmentent doucement lorsque les utilisateurs se connectent, se déconnectent ou que des commandes opérateur sont émises.
- Un répertoire de journalisation client utilise 11 Ko après la connexion et augmente régulièrement lorsque l'utilisateur commence à travailler (cette situation n'est pas illustrée dans l'exemple).
- La déconnexion ajoute 40 Ko supplémentaires dans les journaux utilisateur, soit un total de 51 Ko.

A l'exception de journaux d'audit, les fichiers journaux sont écrasés à chaque redémarrage (pour la tâche démarrée RSE) ou connexion (pour un client), ce qui permet de contrôler la taille totale. La directive `keep.last.log` de `rsed.envvars` peut changer cela, étant donné qu'elle peut demander à RSE de conserver un exemplaire des fichiers journaux précédents. Les exemplaires plus anciens sont toujours supprimés.

Un message d'avertissement est envoyé à la console lorsque l'espace disponible dans le système de fichiers qui contient les fichiers journaux d'audit commence à manquer et que le contrôle est actif. Le message de console (FEK103E) s'affiche régulièrement tant que l'incident lié au manque d'espace n'a pas été résolu. Pour obtenir la liste des messages de la console générés par RSE, voir «Messages de la console», à la page 134.

Les définitions du tableau 46 contrôlent les données écrites dans les répertoires de journalisation ainsi que l'emplacement de ces répertoires.

*Tableau 46. Répertoires de sortie de journal*

Adresse	Directive	Fonction
resecomm.properties	debug_level	Définition du niveau de détails du journal par défaut.
rsed.envvars	keep.last.log	Conservation d'un exemplaire des fichiers journaux précédents avant démarrage/connexion.
rsed.envvars	enable.audit.log	Conservation d'un trace d'audit des actions du client.
rsed.envvars	enable.standard.log	Ecriture des flux stdout et stderr du/des pool(s) d'unités d'exécution dans un fichier journal.
rsed.envvars	DSTORE_TRACING_ON	Activation du journal des actions du magasin de données.
rsed.envvars	DSTORE_MEMLOGGING_ON	Activation du journal relatif à l'utilisation de la mémoire par le magasin de données.
Commande d'opérateur	modify rsecommlog <niveau>	Modification dynamique du niveau de détail du journal de rsecomm.log
Commande d'opérateur	modify rsedaemonlog <niveau>	Modification dynamique du niveau de détail du journal de rsedaemon.log
Commande d'opérateur	modify rseserverlog <niveau>	Modification dynamique du niveau de détail du journal de rseserver.log
Commande d'opérateur	modify rsestandardlog {on   off}	Modification dynamique de la mise à jour de std*.log
rsed.envvars	daemon.log	Chemin d'accès au répertoire de base de la tâche démarrée RSE et des journaux d'audit.
rsed.envvars	user.log	Chemin d'accès au répertoire principal des journaux utilisateur.



Developer for System z, et les logiciels requis (ISPF Client Gateway, par exemple) écrit également les données temporaires dans /tmp et /var/rdz/WORKAREA. La quantité de données écrites suite aux actions de l'utilisateur n'est pas prévisible. Il est donc recommandé de prévoir un espace disponible suffisant dans les systèmes de fichiers contenant ces répertoires.

Developer for system z tente toujours de nettoyer ces fichiers temporaires, mais le nettoyage manuel (voir «(Facultatif) Nettoyage du répertoire WORKAREA», à la page 110) peut être réalisé pratiquement à tout moment.

---

## Définitions de ressources essentielles

### /etc/rdz/rsed.envvars

Les variables d'environnement définies dans rsed.envvars sont utilisées par RSE, Java et z/OS UNIX. Le fichier exemple qui accompagne Developer for System z vise les petites et moyennes installations qui n'ont pas besoin des composants facultatifs de Developer for System z. La section «rsed.envvars - Fichier de configuration RSE», à la page 32 décrit chaque variable définie dans le fichier exemple, les variables ci-dessous devant faire l'objet d'une attention particulière :

**\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Xms128m -Xmx256m"**

Définit la taille de pile initiale (Xms) et maximale (Xmx). Les valeurs par défaut sont respectivement 128M et 256M. Modifiez la valeur pour appliquer la taille de pile de votre choix. Si cette directive est mise en commentaire, les valeurs par défaut Java sont alors utilisées, 4M et 512M respectivement (1M et 64M pour Java 5.0).

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.clients=60"**

Nombre maximal de clients pris en charge par un même pool d'unités d'exécution. La valeur par défaut est 60. Supprimez la mise en commentaire et personnalisez l'option pour limiter le nombre de clients par pool d'unités d'exécution. Notez que d'autres limites risquent d'empêcher RSE d'atteindre cette limite.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.threads=1000"**

Nombre maximum d'unités d'exécution actives d'un pool d'unités d'exécution pour autoriser de nouveaux clients. La valeur par défaut est 1000. Supprimez la mise en commentaire et personnalisez pour limiter le nombre de clients par pool d'unités d'exécution en fonction du nombre d'unités d'exécution utilisées. Notez que chaque connexion client utilise plusieurs unités d'exécution (au moins 16) et que d'autres limites risquent d'empêcher RSE d'atteindre cette valeur maximale.

**Remarque :** Cette valeur doit être inférieure à celle de MAXTHREADS et MAXTHREADTASKS dans SYS1.PARMLIB(BPXPRMxx).

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dminimum.threadpool.process=10"**

Nombre minimal de pools d'unités d'exécution actifs. La valeur par défaut est 1. Supprimez la mise en commentaire de cette ligne et personnalisez-la pour lancer au moins le nombre de processus de pool d'unités d'exécution répertoriés. Les processus de pool d'unité d'exécution sont utilisés pour l'équilibrage de charge des unités d'exécution du serveur RSE. Des processus supplémentaires sont démarrés, si nécessaire. Le démarrage immédiat de nouveaux processus permet d'éviter les délais de connexion mais utilise davantage de ressources pendant les phases d'inactivité.

**#\_RSE\_JAVAOPTS="\$\_RSE\_JAVAOPTS -Dmaximum.threadpool.process=100"**

Nombre maximal de pools d'unités d'exécution actifs. La valeur par défaut est 100. Supprimez la mise en commentaire et personnalisez pour limiter le nombre de processus de pool d'unité d'exécution. Les processus de pool d'unités d'exécution sont utilisés pour l'équilibrage de charge des unités d'exécution du serveur RSE ; si vous les limitez, ils limiteront donc la quantité de connexions client actives.

## **SYS1.PARMLIB(BPXPRMxx)**

RSE est une application Java, ce qui signifie qu'il est actif dans l'environnement z/OS UNIX. BPXPRMxx peut donc aisément devenir un membre parmlib essentiel, étant donné qu'il contient les paramètres permettant de contrôler l'environnement et les systèmes de fichiers z/OS UNIX. BPXPRMxx est décrit dans le document *MVS Initialization and Tuning Reference* (SA22-7592). Les directives suivantes sont réputées avoir un impact sur Developer for System z :

### **MAXPROCSYS(nnnnn)**

Indique le nombre maximal de processus que le système autorise.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 5 et 32767.

Valeur par défaut : 900

### **MAXPROCUSER(nnnnn)**

Indique le nombre maximal de processus qu'un seul ID utilisateur z/OS UNIX peut activer simultanément, quelle que soit la manière dont les processus ont été créés.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 3 et 32767.

Valeur par défaut : 25

#### **Remarque :**

- Tous les processus RSE utilisent le même ID utilisateur z/OS UNIX (celui de l'utilisateur attribué au démon RSE), car tous les client fonctionnent comme des unités d'exécution dans les processus RSE.
- Cette valeur peut également être établie avec la variable PROCUSERMAX du segment de profil de sécurité OMVS de l'utilisateur attribué à la tâche démarrée RSED.

### **MAXTHREADS(nnnnnn)**

Indique le nombre maximal d'unités d'exécution pthread\_created, y compris celles qui sont en cours d'exécution, mises en file d'attente et arrêtées sans être libérées, qu'un seul processus peut activer simultanément. Si vous indiquez la valeur 0, les applications n'utilisent pas pthread\_create.

Gamme de valeurs : nnnnnn est une valeur décimale comprise entre 0 et 100000.

Valeur par défaut : 200

#### **Remarque :**

- Chaque client utilise au moins 16 unités d'exécution dans le processus du pool d'unités d'exécution RSE, plusieurs clients étant actifs à l'intérieur du processus.

- Cette valeur peut également être établie avec la variable THREADSMAX du segment de profil de sécurité OMVS de l'utilisateur attribué à la tâche démarrée RSED. Lorsqu'elle est définie, la valeur THREADSMAX est utilisée pour MAXTHREADS et MAXTHREADTASKS.

#### **MAXTHREADTASKS(nnnnn)**

Indique le nombre maximal de tâches MVS qu'un seul processus peut activer simultanément pour les unités d'exécution pthread\_created.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 0 et 32768.

Valeur par défaut : 1000

#### **Remarque :**

- Chaque unité d'exécution active comporte une tâche MVS (bloc de contrôle des tâches).
- Chaque tâche MVS simultanée requiert un espace de stockage supplémentaire, dont certains doivent être inférieures à la ligne de 16 Mo.
- Chaque client utilise au moins 16 unités d'exécution dans le processus du pool d'unités d'exécution RSE, plusieurs clients étant actifs à l'intérieur du processus.
- Cette valeur peut également être établie avec la variable THREADSMAX du segment de profil de sécurité OMVS de l'utilisateur attribué à la tâche démarrée RSED. Lorsqu'elle est définie, la valeur THREADSMAX est utilisée pour MAXTHREADS et MAXTHREADTASKS.

#### **MAXUIDS(nnnnn)**

Indique le nombre maximal d'ID utilisateur z/OS UNIX (UID) qui peuvent opérer simultanément.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 1 et 32767.

Valeur par défaut : 200

#### **MAXASSIZE(nnnnn)**

Indique les valeurs de ressource RLIMIT\_AS qui vont faire office de valeurs initiales pour les nouveaux processus. RLIMIT\_AS indique la taille de la région de l'espace adresse.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 10485760 (10 mégaoctets) et 2147483647 (2 gigaoctets).

Valeur par défaut : 209715200 (200 mégaoctets)

#### **Remarque :**

- Cette valeur doit être de 2G.
- Cette valeur peut également être établie avec la variable ASSIZEMAX du segment de profil de sécurité OMVS de l'utilisateur attribué à la tâche démarrée RSED.

#### **MAXFILEPROC(nnnnnn)**

Indique le nombre maximal de descripteurs pour les fichiers, sockets, répertoires et autres objets de système de fichiers qu'un seul processus peut activer ou allouer simultanément.

Gamme de valeurs : nnnnnn est une valeur décimale comprise entre 3 et 524287.  
Valeur par défaut : 64000

**Remarque :**

- Toutes les unités d'exécution client d'un pool d'unités d'exécution se trouvent dans un seul processus.
- Cette valeur peut également être établie avec la variable FILEPROCMAX du segment de profil de sécurité OMVS de l'utilisateur attribué à la tâche démarrée RSED.

**MAXMMAPAREA(nnnnn)**

Indique la quantité d'espace de stockage de l'espace de données (en pages) qui peut être allouée pour les mappages mémoire des fichiers z/OS UNIX. L'espace de stockage n'est pas alloué tant que le mappage mémoire n'est pas actif.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 1 et 16777216.  
Valeur par défaut : 40960

**Remarque :** Cette valeur peut également être établie avec la variable MMAPAREAMAX du segment de profil de sécurité OMVS de l'utilisateur attribué à la tâche démarrée RSED.

Utilisez la commande de l'opérateur **SETOMVS** ou **SET OMVS** pour augmenter ou diminuer de manière dynamique (jusqu'à l'IPL suivant) la valeur de l'une des variables BPXPRMxx précédentes. Pour apporter une modification permanente, éditez le membre BPXPRMxx qui va être utilisé pour les IPL. Voir le document *MVS System Commands* (SA22-7627) pour plus d'informations relatives à ces commandes de l'opérateur.

Les définitions suivantes sont des sous-paramètres de l'instruction NETWORK.

**MAXSOCKETS(nnnnnnnn)**

Indique le nombre maximal de sockets pris en charge par ce système de fichiers pour cette famille d'adresses. Il s'agit d'un paramètre facultatif.

Gamme de valeurs : nnnnnnnn est une valeur décimale comprise entre 0 et 16777215.  
Valeur par défaut : 100

**INADDRANYCOUNT(nnnn)**

Indique le nombre de ports que le système réserve pour une utilisation avec PORT 0 et les liaisons INADDR\_ANY, en commençant par le numéro de port spécifié dans le paramètre INADDRANYPORT. Cette valeur est uniquement nécessaire pour CINET (plusieurs piles TCP/IP).

Gamme de valeurs : nnnn est une valeur décimale comprise entre 1 et 4000.  
Valeur par défaut : si INADDRANYPORT et INADDRANYCOUNT ne sont pas spécifiés, la valeur par défaut d'INADDRANYCOUNT est 1000.  
Sinon, aucun port n'est réservé (0).

---

## Définitions de ressource différentes

### Carte EXEC dans le JCL de serveur

Il est recommandé d'ajouter les définitions suivantes à la carte EXEC dans le JCL des serveurs Developer for System z.

#### **REGION=0M**

REGION=0M est recommandé pour les tâches démarrées du démon RSE et du moniteur de travaux JES (RSED et JMON, respectivement). Se faisant, la taille de l'espace adresse est limitée uniquement par l'espace de stockage privé disponible ou par la sorti du système IEFUSI ou IEALIMIT. Notez qu'IBM recommande vivement de ne pas utiliser ces sorties pour les espaces adresses z/OS UNIX (le démon RSE, par exemple).

#### **TIME=NOLIMIT**

Il est recommandé d'utiliser TIME=NOLIMIT pour tous les serveurs Developer for System z. En effet, le temps UC de tous les clients Developer for System z s'accumulent dans les espaces adresses du serveur.

### **FEK.#CUST.PARMLIB(FEJJCNFG)**

Les variables d'environnement définies dans FEJJCNFG sont utilisées par le moniteur de travaux JES. Le fichier exemple qui accompagne Developer for System z vise les petites et moyennes installations. La section «FEJJCNFG, fichier de configuration du moniteur de travaux JES», à la page 27 décrit chaque variable définie dans le fichier exemple, les variables ci-dessous devant faire l'objet d'une attention particulière :

#### **MAX\_THREADS**

Nombre maximal d'utilisateurs qui peuvent utiliser simultanément un moniteur de travaux JES. La valeur par défaut est 200. La valeur maximale est 2147483647. Si vous augmentez cette valeur, vous devez augmenter la taille de l'espace adresse du moniteur de travaux JES.

### **SYS1.PARMLIB(IEASYSxx)**

IEASYSxx contient les paramètres système et est décrit dans le document *MVS Initialization and Tuning Reference* (SA22-7592). Les directives suivantes sont réputées avoir un impact sur Developer for System z :

#### **MAXUSER=nnnnn**

Ce paramètre indique une valeur que le système utilise, sous certaines conditions, pour limiter le nombre de travaux et de tâches démarrées qui peuvent être exécutés simultanément lors d'une IPL donnée.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 0 et 32767. Notez que la somme des valeurs spécifiées pour les paramètres système MAXUSER, RSVSTRT, et RSVNONR ne peut pas dépasser 32767.

Valeur par défaut : 255

### **SYS1.PARMLIB(IVTPRMxx)**

IVTPRMxx permet d'attribuer une valeur aux paramètres du gestionnaire de stockage des communications (CSM) et est décrit dans le document *MVS Initialization and Tuning Reference* (SA22-7592). Les directives suivantes sont réputées avoir un impact sur Developer for System z :

**FIXED MAX(maxfix)**

Définit la quantité maximale d'espace de stockage dédié aux mémoires tampons CSM fixes.

Gamme de valeurs : maxfix est une valeur comprise entre 1024 Ko et 2048 Mo.  
Valeur par défaut : 100 Mo

**ECSA MAX(maxecsa)**

Définit la quantité maximale d'espace de stockage dédié aux mémoires tampons CSM ECSA.

Gamme de valeurs : maxecsa est une valeur comprise entre 1024 Ko et 2048 Mo.  
Valeur par défaut : 100 Mo

**SYS1.PARMLIB(ASCHPMxx)**

Le membre parmlib de ASCHPMxx contient des informations de planification pour le répartiteur de transactions ASCH e est décrit dans le document *MVS Initialization and Tuning Reference* (SA22-7592). Les directives suivantes sont réputées avoir un impact sur Developer for System z :

**MAX(nnnnn)**

Paramètre facultatif de la définition CLASSADD indiquant le nombre maximal de demandeurs de transaction APPC admis pour une classe particulière de demandeurs de transaction. Lorsque cette limite est atteinte, plus aucun espace adresse n'est créé et les demandes entrantes sont placées dans la file d'attente tant que les espaces adresses existants du demandeur ne sont pas disponibles. La valeur ne doit pas dépasser le nombre maximal d'espaces adresses admis par votre installation. N'oubliez pas de comparer les produits sur le système qui va également avoir besoin d'espaces adresses.

Gamme de valeurs : nnnnn est une valeur décimale comprise entre 1 et 64000.  
Valeur par défaut : 1

**Remarque :** Si vous utilisez APPC pour démarrer le service Commandes TSO, la classe de transaction utilisée doit comporter suffisamment de demandeurs de transaction pour en autoriser un par utilisateur simultané de Developer for System z.

---

**Contrôle**

Etant donné que les charges de travail de l'utilisateur peuvent modifier les besoins en ressources système, il est recommandé de contrôler le système régulièrement pour mesurer l'utilisation des ressources, de manière à pouvoir ajuster Rational Developer for System z et les configurations système en fonction des exigences de l'utilisateur. Les commandes suivantes peuvent être utilisées pour faciliter ce processus de contrôle.

**Contrôle de RSE**

Les pools d'unités d'exécution RSE sont le point focal de l'activité d'utilisateur dans Developer for System z et doivent donc être contrôlés pour assurer une utilisation optimale. Le démon RSE peut s'avérer nécessaire pour les informations qui ne peuvent pas être rassemblés avec les outils de contrôle du système habituels.

- Utilisez vos outils de contrôle habituels (RMF, par exemple) pour rassembler des données spécifiques de l'espace adresse (l'espace de stockage réel utilisé et le temps UC, par exemple. Si vous ne disposez pas de ce type d'outil, vous pouvez utiliser des outils comme la vue SDSF DA ou TASID (un outil d'informations système en l'état disponible sur la page Web ISPF "Support and downloads") afin de rassembler des informations de base.
- Lors du démarrage, le démon RSE rapporte la taille de l'espace adresse disponible et la taille de pile Java avec le message de console FEK004I.  
FEK004I RseDaemon: Max Heap Size=65MB and private AS Size=1,959MB
- La commande de l'opérateur **MODIFY RSED,APPL=DISPLAY PROCESS** affiche les processus du pool d'unités d'exécution RSE. La zone "Memory Usage" affiche la quantité de pile Java définie réellement utilisée. Voir Chapitre 8, «Commandes de l'opérateur», à la page 127 pour plus d'informations sur cette commande.

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

Des informations supplémentaires sont fournies lorsque vous utilisez l'option **DETAIL** de la commande de modification **DISPLAY PROCESS** :

```
f rsed,appl=d p,detail
BPXM023I (STCRSE)
ProcessId(33555087) ASId(002E) JobName(RSED8) Order(1)
PROCESS LIMITS:  CURRENT  HIGHWATER  LIMIT
  JAVA HEAP USAGE(%)  10      56      100
    CLIENTS           0       25      60
  MAXFILEPROC        83     103    64000
  MAXPROCUSER        97      99     200
  MAXTHREADS         9       14    1500
  MAXTHREADTASKS     9       14    1500
```

## Contrôle de z/OS UNIX

La plupart des limites z/OS UNIX qui présentent un intérêt pour Developer for System z peuvent être affichées à l'aide des commandes de l'opérateur. Certaines commandes affichent même l'utilisation réelle et la cote d'alerte haute associées à une limite particulière. Voir le document *MVS System Commands* (SA22-7627) pour plus d'informations relatives à ces commandes.

- La directive **LIMMSG(ALL)** de **SYS1.PARMLIB(BPXPRMxx)** demande à z/OS UNIX d'afficher les messages de console (BPXI040I) lorsque l'une des limites parmlib est sur le point d'être atteinte. La valeur par défaut de **LIMMSG** est **NONE**, ce qui désactive la fonction. Utilisez la commande de l'opérateur **SETOMVS LIMMSG=ALL** afin d'activer dynamiquement cette fonction (jusqu'à l'IPL suivante). Voir le document *MVS Initialization and Tuning Reference* (SA22-7592) pour plus d'informations relatives à cette directive.
- La commande de l'opérateur **DISPLAY OMVS,OPTIONS** affichent les valeurs en cours des directives z/OS UNIX qui peuvent être définies de manière dynamique.

```
d omvs,o
BPX0043I 13.10.16 DISPLAY OMVS 066
OMVS 000D ETC/INIT WAIT OMVS=(M7)
CURRENT UNIX CONFIGURATION SETTINGS:
MAXPROCSYS      =      256  MAXPROCUSER      =      16
MAXFILEPROC     =      256  MAXFILESIZE      = NOLIMIT
MAXCPUPTIME     =     1000  MAXUIDS         =      200
MAXPTYS         =      256  MAXASSIZE        = 209715200
MAXMMAPAREA     =      256  MAXTHREADTASKS   =     1000
MAXTHREADS      =      200  MAXSHAREPAGES    =     4096
MAXCORESIZE     = 4194304
```



```

IPCMSGQBYTES   = 2147483647   IPCMSGQMNUM     =      10000
IPCMSGNIDS     =      500     IPCSEMNIIDS     =      500
IPCSEMNIOPS    =      25     IPCSEMNISEMS    =      1000
IPCshmmpages   =      25600   IPCshmniIDS     =      500
IPCshmNSEGS    =      500     IPCshmSPAGES    =      262144
SUPERUSER      = BPXROOT     FORKCOPY         = COW
STEPLIBLIST    =
USERIDALIASTABLE=
SERV_LINKLIB   = POSIX.DYN SERV.LOADLIB  BPXLK1
SERV_LPALIB    = POSIX.DYN SERV.LOADLIB  BPXLK1
PRIORITYPG VALUES: NONE
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGs  =      1000     SHRLIBRGNSIZE   =      67108864
SHRLIBMAXPAGES =      4096     VERSION         = /
SYSCALL COUNTS = NO           TTYGROUP         = TTY
SYSPLX         = NO           BRM SERVER          = N/A
LIMMSG         = NONE         AUTOCVT          = OFF
RESOLVER PROC  = DEFAULT
AUTHPGMLIST    = NONE
SWA            = BELOW

```

- La commande de l'opérateur **DISPLAY OMVS,LIMITS** affiche les informations relatives aux limites parmlib z/OS UNIX System Services en cours, leurs cotes d'alertes haute et l'utilisation en cours du système.

```

d omvs,l
BPX0051I 14.05.52 DISPLAY OMVS 904
OMVS      0042 ACTIVE          OMVS=(69)
SYSTEM WIDE LIMITS:          LIMMSG=SYSTEM

```

	CURRENT USAGE	HIGHWATER USAGE	SYSTEM LIMIT
<b>MAXPROCSYS</b>	<b>1</b>	<b>4</b>	<b>256</b>
<b>MAXUIDS</b>	<b>0</b>	<b>0</b>	<b>200</b>
MAXPTYs	0	0	256
<b>MAXMMAPAREA</b>	<b>0</b>	<b>0</b>	<b>256</b>
MAXSHAREPAGES	0	10	4096
IPCMSGNIDS	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

La commande affiche les cotes d'alerte hautes et l'utilisation en cours d'un processus individuel lorsque le mot clé PID=processid est également spécifié.

```

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 USAGE	HIGHWATER USAGE	PROCESS LIMIT
<b>MAXFILEPROC</b>	<b>83</b>	<b>103</b>	<b>256</b>
MAXFILESIZE	---	---	NOLIMIT
<b>MAXPROCUSER</b>	<b>97</b>	<b>99</b>	<b>200</b>
MAXQUEUEDSIGs	0	1	1000
<b>MAXTHREADS</b>	<b>9</b>	<b>14</b>	<b>200</b>
<b>MAXTHREADTASKS</b>	<b>9</b>	<b>14</b>	<b>1000</b>
IPCshmNSEGS	0	0	500
MAXCORESIZE	---	---	4194304
MAXMEMLIMIT	0	0	16383P



- La commande de l'opérateur **DISPLAY OMVS,PFS** affiche des informations relatives à chaque système de fichiers physique faisant partie intégrante de la configuration z/OS UNIX, qui inclut les piles TCP/IP.

```
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)
```

- La commande de l'opérateur **DISPLAY OMVS,PID=processid** affiche les informations relatives à l'unité d'exécution pour un processus particulier.

```
d omvs,pid=16777456
BPX0040I 15.30.01 DISPLAY OMVS 637
OMVS      000E ACTIVE      OMVS=(76)
USER      JOBNAM     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
0E08F00000000000 005E6C58      .001 PTX JYNV
0E09300000000000 005E6AC0      7.368 PTX JYNV
0E0CB00000000000 005C2CF0 OMVS      1.872 SEL JFNV
0E1920000000003CE 005A0B70 OMVS      IBMUSER   14.088 POL JFNV
0E18D0000000003CF 005A1938      IBMUSER   .581 SND JYNV
```

## Contrôle du réseau

Lors de la prise en charge d'un nombre important de clients se connectant à l'hôte, Developer for System z et votre infrastructure réseau doivent être en mesure de gérer la charge de travail. La gestion du réseau est un sujet largement abordé qui entre dans le domaine d'application de la documentation Developer for System z. Par conséquent, seuls les pointeurs suivants sont fournis.

- La commande de l'opérateur **DISPLAY NET,CSM** permet au gestionnaire de stockage des communications de contrôler l'utilisation de l'espace de stockage. Vous pouvez utiliser cette commande afin de déterminer la quantité d'espace de stockage CSM en cours d'utilisation pour ECSA et les pools de stockage d'espace de données (voir *Communications Server SNA Operations* (SC31-8779)).

## Contrôle des systèmes de fichiers z/OS UNIX

Developer for System z utilise les systèmes de fichiers z/OS UNIX pour stocker les différents types de données (les journaux et fichiers temporaires, par exemple). Utilisez la commande z/OS UNIX **df** pour connaître le nombre de descripteurs de

fichier encore disponibles et la quantité d'espace libre avant la création de la prochaine étendue du fichier HFS ou zFS sous-jacent.

```
$ df
Mounted on    Filesystem      Avail/Total    Files    Status
/tmp          (OMVS.TMP)      1393432/1396800 4294967248 Available
/u/ibmuser    (OMVS.U.IBMUSER) 1248/1728      4294967281 Available
/usr/lpp/rdz  (OMVS.FEK.HHOP760) 3062/43200     4294967147 Available
/var          (OMVS.VAR)      27264/31680    4294967054 Available
```

---

## Exemple de configuration

L'exemple de configuration ci-dessous illustre la configuration requise permettant de prendre en charge ces exigences :

- 500 connexions client simultanées
- 300 générations MVS simultanées (travail par lots)
- 200 connexions CARMA simultanées (utilisation de la méthode de configuration CRASTART)
- 3 heures de délai d'attente d'inactivité
- N'autorisez pas l'utilisation de z/OS UNIX
- SCLM Developer Toolkit et File Manager Integration ne sont pas utilisés
- Prévoyez une utilisation moyenne de pile Java de 5 Mo
- Les utilisateurs disposent d'UID z/OS UNIX uniques

## Nombre de pools d'unités d'exécution

Par défaut, Developer for system z tente d'ajouter 60 utilisateurs à un seul pool d'unités d'exécution. Toutefois, nos exigences stipulent que le délai d'attente d'inactivité soit actif. Le tableau 44, à la page 224 indique qu'une unité d'exécution va être ajoutée par client connecté. Il s'agit d'une unité d'exécution de temporisation qui doit donc être active en permanence. Cela empêche RSE de placer 60 utilisateurs dans un seul pool d'unités d'exécution, étant donné que  $60 \times (16+1) = 1020$  et que, par défaut, la valeur 1000 a été attribuée à `maximum.threads`.

La valeur de `maximum.threads` pourrait être augmentée, mais compte tenu de l'obligation de prévoir une pile moyenne Java de 5 Mo par utilisateur, la valeur de `maximum.clients` a été abaissée à 50. La taille maximale de la pile de Java de 256 Mo ( $5 \times 50 = 250$ ) est respectée.

Avec 50 clients par pool d'unités d'exécution et la nécessité de prendre en charge 500 connexions, nous savons désormais que nous allons avoir besoin de 10 espaces adresses de pool d'unités d'exécution.

## Détermination des limites minimales

A l'aide des formules présentées ci-avant dans ce chapitre et les critères établis au début de la présente section, il est possible de déterminer l'utilisation des ressources.

- Nombre d'espaces adresses - maximal
$$3 + A + N \times (x + y + z) + (2 + N \times 0.01)$$
$$3 + 10 + 500 \times 1 + 200 \times 1 + 300 \times 1 + (2 + 500 \times 0.01) = 1020$$
- Nombre d'espaces adresses - par utilisateur
$$x + y + z$$
$$1 + 1 + 1 = 3$$
- Nombre de processus - maximal

$$7 + 2*A + N*(x + y + z) + (10 + N*0.05)$$

$$7 + 2*10 + 500*2 + 200*1 + 300*0 + (10 + 500*0.05) = 1562$$

- Nombre de processus - par utilisateur  
 $(x + y + z) + 5*s$   
 $(2 + 1 + 0) + 5*0 = 3$
- Nombre d'unités d'exécution - pool d'unités d'exécution RSE  
 $9 + N*(16 + x + y + z) + (20 + N*0.1)$   
 $9 + 60*(16 + 1 + 4 + 0) + (20 + 60*0.1) = 1295$
- Nombre d'unités d'exécution - moniteur de travaux JES  
 $3 + N$   
 $3 + 500 = 503$
- ID utilisateur  
 $500 + 3 = 503$

Les 3 ID utilisateur supplémentaires sont pour STCJMON, STCLOCK et STCRSE, les ID utilisateur de la tâche démarrée Developer for System z.

## Définition des limites

Maintenant que les numéros d'utilisation des ressources sont connus, il est possible de personnaliser les directives de personnalisation avec les valeurs appropriées.

- /etc/rdz/rsed.envvars
  - Xmx256m
  - non modifié
  - Dmaximum.clients=50
  - Dmaximum.threads=1000
  - non modifié
  - Dminimum.threadpool.process=10
  - Ce changement est facultatif ; RSE démarrera des nouveaux pools d'unités d'exécution, si nécessaire
  - DHIDE\_ZOS\_UNIX=true
  - DDSTORE\_IDLE\_SHUTDOWN\_TIMEOUT=10800000
- FEK.#CUST.PARMLIB(FEJJCNFG)
  - MAX\_THREADS=503
- SYS1.PARMLIB(BPXPRMxx)
  - MAXPROCSYS(2500)
  - 1562 au moins, mémoire tampon supplémentaire ajoutée pour les tâches autres que Developer for System z
  - MAXPROCUSER(25)
  - non modifié, au moins 3
  - MAXTHREADS(1500)
  - doit être de 503 au moins (pour le moniteur de travaux JES)
  - si THREADSMAX du segment OMVS de l'ID utilisateur STCRSE est utilisé pour définir la limite de RSE (1295 au moins)
  - MAXTHREADTASKS(1500)

doit être de 503 au moins (pour le moniteur de travaux JES)  
si THREADSMAX du segment OMVS de l'ID utilisateur STCRSE  
est utilisé pour définir la limite de RSE (1295 au moins)

- MAXUIDS(700)

503 au moins, mémoire tampon supplémentaire ajoutée pour les tâches  
autres que Developer for System z

- MAXASSIZE(209715200)

non modifié (valeur par défaut du système de 200 Mo),  
nous utilisons ASSIZEMAX dans le segment OMVS  
de l'ID utilisateur STCRSE

- SYS1.PARMLIB(IEASYSxx)
  - MAXUSER=2000

1020 au moins, mémoire tampon supplémentaire ajoutée  
pour les tâches autres que Developer for System z

- Segment OMVS de l'ID utilisateur STCRSE
  - ASSIZEMAX(2147483647)

2 Go

## Utilisation des ressources du moniteur

Après l'activation des limites du système comme documenté dans «Définition des limites», à la page 245, nous pouvons démarrer le contrôle de l'utilisation des ressources par Developer for System z pour voir si un ajustement de certaines variables est nécessaire. La figure 58, à la page 247 affiche l'utilisation des ressources après la connexion de 495 utilisateurs. (L'exemple présenté sur la figure montre seulement la connexion. Aucune action utilisateur n'est indiquée dans l'exemple.)

```

BPXM023I (STCRSE)
ProcessId(16779764) Memory Usage(10%) Clients(50) Order(1)
ProcessId(67108892) Memory Usage(16%) Clients(50) Order(2)
ProcessId(67108908) Memory Usage(10%) Clients(50) Order(3)
ProcessId(67108898) Memory Usage(16%) Clients(50) Order(4)
ProcessId(67108916) Memory Usage(16%) Clients(50) Order(5)
ProcessId(67108897) Memory Usage(16%) Clients(50) Order(6)
ProcessId(67108921) Memory Usage(16%) Clients(50) Order(7)
ProcessId(83886146) Memory Usage(16%) Clients(50) Order(8)
ProcessId(67108920) Memory Usage(16%) Clients(50) Order(9)
ProcessId(3622      ) Memory Usage(8%) Clients(45) Order(10)

```

Jobname	Cpu time	Storage	EXCP
-----	-----	-----	-----
JMON	1.74	43.0M	2753
LOCKD	10.05	31.9M	24621
RSED	6.65	40.1M	41780
RSED1	8.17	187.0M	76566
RSED2	13.04	184.9M	78946
RSED3	17.77	181.1M	76347
RSED4	11.63	174.9M	74638
RSED5	15.27	172.9M	72883
RSED6	13.85	180.8M	75031
RSED7	9.79	174.3M	76636
RSED8	21.59	176.1M	70583
RSED8	18.88	184.7M	76953
RSED9	9.52	189.8M	80490

Figure 58. Utilisation des ressources de la configuration modèle



---

## Chapitre 14. Remarques relatives aux performances

z/OS est un système d'exploitation hautement personnalisable, et des modifications (parfois mineures) du système peuvent présenter un impact très important sur les performances globales. Le présent chapitre met en évidence certaines modifications qui peuvent être apportées afin d'améliorer les performances de Developer for System z.

Pour plus d'informations sur l'optimisation du système, voir le document *MVS Initialization and Tuning Guide* (SA22-7591) and *UNIX System Services Planning* (GA22-7800).

---

### Utilisation du système de fichiers zFS

zFS (zSeries File System) et HFS (Hierarchical File System) sont tous deux des systèmes de fichiers UNIX qui peuvent être utilisés dans un environnement z/OS UNIX. Cependant, zFS fournit les fonctions et avantages suivants :

- Des gains de performances sont constatés dans nombre d'environnements de clients lors de l'accès à des fichiers approchant une taille de 8 Ko, fréquemment ouverts et mis à jour. Les performances d'accès aux fichiers de plus petite taille sont équivalentes à celles du système de fichiers HFS.
- Clonage en lecture seule d'un système de fichiers du même fichier. Le système de fichiers cloné peut être accessible aux utilisateurs afin de fournir une copie instantanée en lecture seule d'un système de fichiers. Cette fonction disponible uniquement dans un environnement non sysplex est facultative.
- zFS est le système de fichiers stratégique z/OS UNIX. Les fonctionnalités HFS ont été stabilisées et des optimisations apportées au système de fichiers seront uniquement appliquées à zFS.

Pour plus d'informations sur zFS, voir le document *UNIX System Services Planning* (GA22-7800).

---

### Eviter l'emploi de STEPLIB

Chaque processus z/OS UNIX qui présente un STEPLIB propagé de parent à enfant ou à travers une commande exec utilise environ 200 octets de zone de mémoire commune étendue ECSA. Si aucune variable d'environnement STEPLIB n'est définie ou qu'elle est définie comme STEPLIB=CURRENT, z/OS UNIX propage toutes les allocations TASKLIB, STEPLIB et JOBLIB actuellement actives lors d'une fonction fork(), spawn(), ou exec().

Developer for System z présente une valeur par défaut STEPLIB=NONE codée dans rsed.envvars, comme décrit dans le fichier de configuration rsed.envvars. Pour les raisons mentionnées ci-avant, il est conseillé de ne pas modifier cette directive et de placer les fichiers ciblés dans LINKLIST ou dans la zone permanente de programme LPA à la place.

---

## Amélioration de l'accès aux bibliothèques du système

Certaines bibliothèques du système ainsi que certains modules de chargement sont fortement utilisés par z/OS UNIX et par les activités de développement d'applications. En améliorant leur accès (en les ajoutant à la zone permanente de programme (LPA), par exemple), il est possible d'améliorer les performances du système. Pour plus d'informations sur la modification des membres SYS1.PARMLIB décrits ci-après, voir le document *MVS Initialization and Tuning Reference* (SA22-7592).

### Bibliothèque d'exécution Language Environment (LE)

Lorsque des programmes C (y compris le shell z/OS UNIX) sont exécutés, ils utilisent fréquemment des routines issues de la bibliothèque d'exécution Language Environment (LE). En moyenne, environ 4 mégaoctets de bibliothèque d'exécution sont chargés dans la mémoire pour chaque espace adresse qui exécute un programme LE activé, et sont copiés à chaque fourche.

CEE.SCEELPA

Le fichier CEE.SCEELPA contient un sous-ensemble de routines d'exécution LE, qui sont fortement utilisées par z/OS UNIX. Il est conseillé, pour gagner le maximum de performances, d'ajouter ce fichier à SYS1.PARMLIB(LPALSTxx). De cette manière, les modules sont lus une seule fois sur le disque, et sont stockés dans un emplacement partagé.

**Remarque :** Ajoutez les instructions suivantes à SYS1.PARMLIB(PROGxx) si vous préférez ajouter les modules de chargement à la zone permanente de programme (LPA) dynamique.

```
LPA ADD MASK(*) DSNAME(CEE.SCEELPA)
```

Il est également conseillé de placer les bibliothèques d'exécution LE CEE.SCEERUN et CEE.SCEERUN2 dans LINKLIST, en ajoutant les fichiers à SYS1.PARMLIB(LNKLISTxx) ou à SYS1.PARMLIB(PROGxx). Cela élimine le temps système de z/OS UNIX STEPLIB, et il y a moins d'entrées/sorties en raison de la gestion par LLA et VLF ou par des produits similaires.

**Remarque :** Ajoutez également la bibliothèque de classe de la bibliothèque de chargement dynamique C/C++ CBC.SCLBDLL à LINKLIST, pour les mêmes raisons.

Si vous décidez de ne pas mettre ces bibliothèques dans LINKLIST, vous devez alors configurer l'instruction STEPLIB appropriée dans rsed.envvars, comme décrit dans le fichier de configuration rsed.envvars. Même si cette méthode utilise toujours de la mémoire virtuelle supplémentaire, vous pouvez améliorer les performances en définissant les bibliothèques d'exécution LE pour LLA ou un produit similaire. Cela réduit les entrées/sorties nécessaires au chargement des modules.

### Développement d'applications

Les performances des systèmes sur lesquels le développement d'applications est l'activité principale peuvent également être améliorées si vous placez l'éditeur de liens dans la LPA dynamique, en ajoutant les lignes suivantes à SYS1.PARMLIB(PROGxx) :

```
LPA ADD MODNAME(CEEINIT,CEEELIBM,CEEV003,EDCV) DSNAME(CEE.SCEERUN)
```

```
LPA ADD MODNAME(IEFIB600,IEFXB603) DSNAME(SYS1.LINKLIB)
```



Pour le développement C/C++, vous pouvez également ajouter le fichier du compilateur CBC.SCCNCMP à SYS1.PARMLIB(LPALSTxx).

Les instructions ci-avant sont des exemples de candidats LPA possibles, mais les besoins de votre site peuvent être différents. Voir *Language Environment Customization* (SA22-7564) pour plus d'informations sur le placement d'autres modules de chargement LE dans la LPA dynamique. Pour plus d'informations sur l'insertion de modules de chargement de compilateur C/C++ dans la LPA dynamique, voir le document *UNIX System Services Planning* (GA22-7800).

---

## Amélioration des performances du contrôle d'autorisations d'accès

Afin d'améliorer les performances des contrôles d'autorisations d'accès effectués pour z/OS UNIX, définissez le profil BPX.SAFFASTPATH dans la classe FACILITY de votre logiciel de sécurité. Cela réduit le temps système des contrôles des droits d'accès de z/OS UNIX pour une large plage d'opérations. Cette gamme comprend le contrôle de l'accès aux fichiers, le contrôle de l'accès aux communications interprocessus, et le contrôle de propriété de processus. Pour plus d'informations sur ce profil, voir *UNIX System Services Planning* (GA22-7800).

**Remarque :** Les utilisateurs n'ont pas besoin d'avoir de permission pour le profil BPX.SAFFASTPATH.

---

## Gestion de la charge de travail

Chaque site a des besoins spécifiques, et il est possible de personnaliser le système d'exploitation z/OS pour tirer le meilleur parti des ressources disponibles pour répondre à ces besoins. Avec la gestion de charge de travail, vous pouvez définir des objectifs de performances et leur attribuer une importance. Vous définissez les buts en terme d'activités, et le système décide quelles ressources, comme le stockage ou l'unité centrale, doivent être allouées à la tâche pour qu'elle atteigne son but.

Les performances de Developer for System z peuvent être équilibrées en paramétrant les but adéquats pour ses processus. Vous trouverez des instructions générales ci-dessous :

- En cas d'utilisation, attribuez une transaction APPC à un groupe de performances TSO.
- Attribuez un groupe de performances de tâche démarrée (SYSSTC) pour les espaces adresses du serveur Developer for System z : moniteur de travaux JES (JMON), démon Lock (LOCKD), démon RSE (RSED) et pools d'unités d'exécution RSE (RSEDx).

Pour des informations détaillées sur le sujet, voir *MVS Planning Workload Management* (SA22-7602).

---

## Taille de pile Java fixe

Avec une pile à taille fixe, aucune expansion ou contraction de pile ne peut se produire, ce qui peut permettre des gains de performances significatifs dans certaines situations. Toutefois, l'emploi d'une pile de taille fixe n'est généralement pas une bonne idée, dans la mesure où elle retarde le démarrage de la récupération de place jusqu'au moment où la pile est pleine, ce qui est une tâche très importante. Elle augmente également le risque de fragmentation, qui nécessite une compression de la pile. Aussi, n'utilisez les piles à taille fixe qu'après avoir effectué des essais

appropriés, ou sur instructions du point service IBM. Pour plus d'informations sur les tailles de pile et la récupération de place, voir le document *Java Diagnostics Guide* (SC34-6650).

La taille de pile initiale par défaut d'une machine virtuelle z/OS Java (JVM) est 1 mégaoctet. La taille maximale est 64 mégaoctets. Les limites peuvent être définies avec les options de ligne de commande `-Xms` (initiale) et `-Xmx` (maximale) Java.

Dans Developer for System z, les options de ligne de commande Java sont définies dans la directive `_RSE_JAVAOPTS` de `rsed.envvars` (voir la section «Définition de paramètres de démarrage Java supplémentaires avec `_RSE_JAVAOPTS`», à la page 42).

```
#_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xms128m -Xmx128m"
```

---

## Option Java -Xquickstart

**Remarque :** L'option Java `-Xquickstart` est uniquement utile si vous utilisez la méthode de démarrage REXEC/SSH alternative pour le serveur RSE. Cette méthode est présentée dans «(Facultatif) Utilisation de REXEC (ou SSH)», à la page 104.

L'option `-Xquickstart` peut être utilisée pour améliorer le délai de démarrage de certaines applications Java. L'option `-Xquickstart` entraîne l'exécution du compilateur JIT (Just In Time) avec un sous-ensemble d'optimisations (il s'agit d'une compilation rapide). Cette compilation rapide permet d'améliorer le délai de démarrage.

L'option `-Xquickstart` convient aux applications avec un temps d'exécution plus court, notamment les applications pour lesquelles le délai d'exécution n'est pas concentré dans un petit nombre de méthodes. L'option `-Xquickstart` peut affecter les performances si elle est utilisée avec des applications connaissant un délai d'exécution plus long qui contiennent des méthodes à chaud.

Pour activer l'option `-Xquickstart` pour le serveur RSE, ajoutez la directive suivante à la fin de `rsed.envvars` :

```
_RSE_JAVAOPTS="$_RSE_JAVAOPTS -Xquickstart"
```

---

## Partage de classes entre JVM

IBM Java Virtual Machine (JVM) versions 5 et suivantes vous permet de partager entre JVM les amorces et les classes d'application par leur stockage dans un cache de la mémoire partagée. Le partage de classes réduit la consommation globale de mémoire virtuelle quand plusieurs JVM partagent un cache. Le partage de classes réduit également le temps de démarrage d'une JVM une fois que le cache a été créé.

Le cache de classes partagées est indépendant de toute JVM active et se conserve au-delà de la durée de vie de la JVM qui l'a créé. Dans la mesure où le cache de classes partagées se conserve au-delà de la durée de vie de toute JVM, le cache est mis à jour de façon dynamique afin de refléter toute modification qui peut avoir été apportée aux JAR ou aux classes sur le système de fichiers.

Le temps système pour créer et remplir un nouveau cache est minimal. Le coût de démarrage en temps d'une JVM unique est généralement entre 0 et 5 % plus lent

que celui d'un système qui n'utilise pas le partage de classes, en fonction du nombre de classes qui est chargé. L'amélioration du temps de démarrage de JVM avec un cache rempli est généralement entre 10 % et 40 % plus rapide par rapport à celui d'un système n'utilisant pas le partage de classe, en fonction du système d'exploitation et du nombre de classes chargées. Plusieurs JVM exécutées en même temps présenteront des bénéfices plus importants en terme de temps de démarrage.

Pour plus d'informations sur le partage des classes, voir le document *Java SDK and Runtime Environment User Guide*.

## Activation du partage de classe

Pour activer le partage de classe pour le serveur RSE, ajoutez la directive suivante à la fin de `rsed.envvars`. La première instruction définit un cache nommé RSE avec accès de groupe et permet au serveur RSE de démarrer même si le partage de classes échoue. La seconde instruction est facultative. Elle définit la taille du cache à 6 mégaoctets (la valeur par défaut du système étant 16 mégaoctets). La troisième instruction ajoute les paramètres de partage de classe aux options de démarrage Java.

```
_RSE_CLASS_OPTS=-Xshareclasses:name=RSE,groupAccess,nonFatal
# RSE_CLASS_OPTS="$ _RSE_CLASS_OPTS -Xscmx6m
_RSE_JAVA_OPTS="$ _RSE_JAVA_OPTS $ _RSE_CLASS_OPTS"
```

**Remarque :** Comme il est mentionné dans «Sécurité de la mémoire cache», tous les utilisateurs des classes partagées doivent avoir le même identificateur de groupe primaire (ID groupe). Cela signifie que les utilisateurs doivent avoir le même groupe par défaut défini dans le logiciel de sécurité, ou que des groupes par défaut différents présentent le même ID groupe dans leur segment OMVS.

## Limites de la taille de la mémoire cache

Le maximum théorique de la taille de la mémoire cache est de 2 gigaoctets. La taille de la mémoire cache que vous pouvez indiquer est limitée par la quantité de mémoire physique et par le fichier d'échange disponible pour le système. Dans la mesure où l'espace d'adresse virtuelle d'un processus est partagée entre la mémoire cache de classe partagée et la pile Java, l'augmentation de la taille maximale de la pile Java réduit la taille de la mémoire cache de classes partagées que vous pouvez créer.

## Sécurité de la mémoire cache

L'accès au cache de classes partagées est limité par les autorisation du système d'exploitation et par les autorisations de sécurité de Java.

Par défaut, les caches de classes sont créés avec une sécurité au niveau utilisateur, de sorte que seul l'utilisateur qui a créé le cache peut y avoir accès. Dans z/OS UNIX, l'option `groupAccess` donne accès à tous les utilisateurs du groupe primaire de l'utilisateur qui a créé le cache. Toutefois, quel que soit le niveau d'accès utilisé, un cache peut uniquement être détruit par l'utilisateur qui l'a créé, ou par le superutilisateur (UID 0).

Pour plus d'informations sur les options de sécurité supplémentaires à l'aide d'un gestionnaire de sécurité Java, voir le document *Java SDK and Runtime Environment User Guide*.

## SYS1.PARMLIB(BPXPRMxx)

Certains paramètres de SYS1.PARMLIB(BPXPRMxx) affectent les performances des classes partagées. L'emploi des mauvais paramètres peut arrêter le fonctionnement des classes partagées. Ces paramètres peuvent également avoir un impact sur les performances. Pour de plus amples informations sur les implications relatives aux performances et à l'utilisation de ces paramètres, voir les documents *MVS Initialization and Tuning Reference* (SA22-7592) et *UNIX System Services Planning* (GA22-7800). Les paramètres BPXPRMxx les plus significatifs qui affectent le fonctionnement des classes partagées sont :

- MAXSHAREPAGES, IPCSHMPAGES, IPCSHMPAGES et IPCSHMNSEGS

Ces paramètres affectent la quantité de pages de mémoire partagée disponible pour la machine virtuelle Java. La taille de page partagée pour un service de système z/OS UNIX 31-bit est fixée à 4 kilooctets. Les classes partagées essaient de créer par défaut une mémoire cache de 16 mégaoctets. Définissez donc IPCSHMPAGES à une valeur supérieure à 4096.

Si vous définissez une taille de cache en utilisant -Xscmx, la machine virtuelle Java arrondira la valeur au mégaoctet le plus proche. Vous devez en tenir compte lors de la définition de IPCSHMPAGES sur le système.

- IPCSEMNIIDS et IPCSEMNSEMS

Ces paramètres affectent la quantité de sémaphores disponibles pour les processus UNIX. Les classes partagées utilisent les sémaphores de communication interprocessus pour dialoguer entre les machines virtuelles Java.

## Espace disque

Le cache de classes partagées nécessite de l'espace disque pour stocker les informations d'identification concernant les caches qui existent sur le système. Ces informations sont stockées dans /tmp/javasharedresources. Si le répertoire des informations d'identification est effacé, la machine virtuelle Java ne peut plus identifier les classes partagées sur le système, et doit recréer le cache.

## Utilitaires de gestion de la mémoire cache

La ligne de commande Java -Xshareclasses peut prendre un certain nombre d'options, dont certaines sont des utilitaires de gestion de cache. Trois d'entre-elles sont présentées dans l'exemple ci-après (\$ est l'invite z/OS UNIX). Pour une présentation complète des options de ligne de commande prises en charge, voir le document *Java SDK and Runtime Environment User Guide*.

```
$ java -Xshareclasses:listAllCaches
Shared Cache      OS shmid      in use      Last detach time
RSE               401412       0           Mon Jun 18 17:23:16 2007
```

Could not create the Java virtual machine.

```
$ java -Xshareclasses:name=RSE,printStats
```

Current statistics for cache "RSE":

```
base address      = 0x0F300058
end address       = 0x0F8FFFF8
allocation pointer = 0x0F4D2E28
```

```
cache size        = 6291368
free bytes        = 4355696
ROMClass bytes    = 1912272
Metadata bytes    = 23400
Metadata % used   = 1%
```

```
# ROMClasses      = 475
# Classpaths      = 4
# URLs            = 0
# Tokens          = 0
# Stale classes   = 0
% Stale classes   = 0%
```

Cache is 30% full

Could not create the Java virtual machine.

```
$ java -Xshareclasses:name=RSE,destroy
JVMSHRC010I Shared Cache "RSE" is destroyed
Could not create the Java virtual machine.
```

**Remarque :**

- Les utilitaires de cache effectuent les opérations nécessaires sur le cache indiqué sans démarrer la machine virtuelle Java, de sorte que le message "Could not create the Java virtual machine" soit normal.
- Un cache ne peut être détruit uniquement lorsque toutes les machines virtuelles JAVA qui l'utilisent sont arrêtées et que l'utilisateur qui émet la commande dispose d'autorisations suffisantes.



---

## Chapitre 15. Remarques à propos de CICSTS

Habituellement, la définition des ressources dans CICS est gérée par un administrateur système CICS. Autoriser les développeurs d'applications à définir les ressources CICS n'est pas si simple, pour les raisons suivantes :

- La plupart des définitions de ressources CICS se composent de nombreux paramètres qui, par leur complexité, leurs liens avec d'autres définitions de ressource et les normes d'usine, nécessitent des connaissances en administration CICS pour effectuer une définition correcte. Une mauvaise définition risque d'entraîner des résultats imprévus qui peuvent avoir une incidence sur la totalité de la région CICS.
- La plupart des magasins fournissent des environnements de test et de développement CICS qui doivent être disponibles pour une utilisation partagée par plusieurs développeurs et groupes d'applications. De nombreux magasins ont mis en place des contrats de service pour ces environnements. Un contrôle strict des environnements est nécessaire pour satisfaire ces contrats.

Developer for System z aborde ces questions en autorisant les administrateurs CICS à contrôler les valeurs par défaut de la définition de ressource CICS, ainsi que les propriétés d'affichage d'un paramètre de définition de ressource CICS à l'aide du serveur de définition de ressource CICS, qui fait partie intégrante du gestionnaire de déploiement d'application.

Par exemple, l'administrateur CICS peut fournir certains paramètres de définition de ressource CICS qui risquent de ne pas avoir été mis à jour par le développeur d'applications. D'autres paramètres de définition de ressource CICS sont réactualisables, avec ou sans les valeurs fournies par défaut, ou le paramètre de définition de ressource CICS peut être masqué pour éviter toute complexité inutile.

Une fois que le développeur d'applications est satisfait des définitions de ressource CICS, il peut les installer immédiatement dans l'environnement de test CICS en cours d'exécution ou les exporter dans un manifeste pour qu'un administrateur CICS puisse les éditer et les valider. L'administrateur CICS peut utiliser l'utilitaire d'administration (utilitaire de traitement par lots) ou l'outil de traitement de manifestes pour mettre en oeuvre les modifications de définition de ressource.

**Remarque :** L'outil de traitement de manifestes est un module d'extension pour IBM CICS Explorer.

Pour plus d'informations sur les tâches requises pour configurer le gestionnaire de déploiement d'application sur votre système hôte, voir Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75.

La personnalisation du gestionnaire de déploiement d'application ajoute les services suivants à Developer for System z :

- (Sur le client) IBM CICS Explorer apporte une infrastructure Eclipse qui permet de visualiser et gérer les ressources CICS et qui assure une plus grande intégration entre les outils CICS.
- (sur le client) éditeur de définition de ressource CICS (CRD)
- (sur l'hôte) le serveur de définition de ressources (CRD), qui fonctionne comme une application CICS

Le serveur de définition de ressource CICS (CRD) du gestionnaire de déploiement d'application se compose du serveur CRD lui-même, d'un référentiel CRD, des définitions de ressource CICS associées, des fichiers de liens de service Web et d'un exemple de gestionnaire des messages de pipeline. Le serveur CRD doit être exécuté dans une région gérant le Web (WOR), référencée dans la documentation de Developer for System z sous le terme de région de connexion CICS primaire.

Voir le centre de documentation Developer for System z (<http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp>) pour en savoir plus sur les services du gestionnaire de déploiement d'application disponibles dans la version en cours de Developer for System z.

---

## RESTful par opposition à Web Service

Les versions 1.4 et ultérieures de CICS Transaction Server proposent un support pour l'interface HTTP reposant sur les principes RESTful (Representational State Transfer). Cette interface RESTful est désormais l'interface CICSTS stratégique utilisée par les applications client. L'ancienne interface Web Service a été stabilisée, les améliorations concernant uniquement l'interface RESTful.

Le gestionnaire de déploiement d'application suit cette logique et demande au serveur CRD RESTful tous les nouveaux services de Developer for System version 7.6 ou ultérieures.

Les interfaces RESTful et Web Service peuvent être activées simultanément dans une seule région CICS, le cas échéant. Dans ce cas, la région contient deux serveurs CRD actifs. Les deux serveurs partagent le même référentiel de CRD. Notez que CICS émet des avertissements relatives aux définitions en double lorsque la deuxième interface est définie dans la région.

---

## Régions de connexion primaires versus régions de connexion non primaires

Un environnement de test CICS peut se composer de plusieurs régions connectées par MRO (exploitation multirégionale). Au cours du temps, des désignations non officielles ont été utilisées pour catégoriser ces régions. Les désignations standard sont : région TOR (région gérant les terminaux), région WOR (région gérant le Web), région AOR (région gérant les applications) et région DOR (région gérant les données).

Une région gérant le Web est utilisée pour mettre en oeuvre la prise en charge des services Web CICS et le serveur de définition de ressource CICS du gestionnaire de déploiement d'application doit être exécuté dans cette région. Cette région est connue dans le Gestionnaire de déploiement d'application en tant que région de connexion CICS primaire. Le client CRD ADM implémente une connexion de service Web à la région primaire de connexion CICS.

Les régions de connexion non primaires CICS constituent toutes les autres régions que le serveur CRD peut gérer. Ce service englobe l'affichage des ressources à l'aide d'IBM CICS Explorer et des ressources de définition à l'aide de l'éditeur de définition de ressource CICS.

Si les services d'application métier (BAS) SM CICSplex sont utilisés pour gérer les définitions de ressource CICS de la région de connexion primaire CICS, toutes les autres régions CICS gérées par BAS peuvent être gérées par le serveur CRD.



Les régions CICS non gérées par BAS requièrent des modifications supplémentaires afin de pouvoir être gérées par le serveur CRD.

---

## Consignation des messages d'installation des ressources CICS

Les actions effectuées par le serveur CRD sur les ressources CICS sont consignées dans la file d'attente TD CSDL CICS qui indique généralement la définition de données MSGUSR de votre région CICS.

Si CICSplex SM Business Application Services (BAS) est utilisé pour gérer vos définitions de ressources CICS, la directive CICSplex SM EYUPARM BASLOGMSG doit être définie sur (YES) pour la consignation à créer.

---

## gestionnaire de déploiement d'application, sécurité

### Sécurité du référentiel CRD

Le fichier VSAM du référentiel du serveur CRD contient toutes les définitions de ressource par défaut ; il doit par conséquent être protégé contre les mises à jour tout en autorisant les développeurs à consulter les valeurs qui y sont conservées. Pour protéger le référentiel CRD, reportez-vous à «Définition des profils de fichier», à la page 182 afin d'obtenir des exemples de commande RACF.

### pipeline,sécurité

Quand le message SOAP est reçu par CICS par l'intermédiaire de l'interface Web Service, il est traité par un pipeline. Un pipeline désigne un ensemble de gestionnaires des messages qui sont exécutés dans l'ordre. CICS lit le fichier de configuration du pipeline pour déterminer les gestionnaires de messages à appeler dans le pipeline. Un gestionnaire des messages est un programme qui permet d'exécuter un traitement spécial des demandes et réponses de service Web.

Application Deployment Manager procure un exemple de fichier de configuration de pipeline qui spécifie l'appel d'un gestionnaire des messages et d'un programme de traitement de l'en-tête SOAP.

Le gestionnaire de message de pipeline (ADNTMSGH) est utilisé par sécurité dans le traitement de l'ID utilisateur et des mots de passe dans l'en-tête du protocole SOAP. ADNTMSGH est référencé par le fichier de configuration de pipeline et doit donc être placé dans la concaténation RPL CICS.

### Sécurité de transaction

CPIH correspond à l'ID de transaction par défaut sous lequel une application appelée par un pipeline sera exécutée. En règle générale, CPIH est défini pour un niveau minimal d'autorisation.

Developer for System z met à disposition plusieurs transactions qui sont utilisées par le serveur CRD lors de la définition et de la consultation des ressources CICS. Ces ID de transaction sont définis par le serveur CRD en fonction de l'opération demandée. Voir le Chapitre 4, «(Facultatif) Gestionnaire de déploiement d'application», à la page 75 pour plus d'informations sur la personnalisation des ID transaction.

Transaction	Description
ADMS	Pour les demandes, par l'outil de traitement des manifestes, de modification des ressources CICS. Généralement destiné aux administrateurs CICS. Cette transaction requiert des droits d'accès de haut niveau.
ADMI	Pour les demandes qui définissent, installent ou désinstallent les ressources CICS. Cette transaction peut requérir des droits d'accès de niveau intermédiaire en fonction des règles d'administration de votre site.
ADMR	Pour toutes les autres demandes qui récupèrent des informations sur les ressources ou l'environnement CICS. Cette transaction peut requérir des droits d'accès de niveau minimum en fonction des règles d'administration de votre site.

Certaines ou toutes les demandes de définition de ressource effectuées par les transactions du serveur CRD doivent être sécurisées. Au minimum, les commandes de mise à jour (mise à jour des paramètres de service Web par défaut, des paramètres de descripteur par défaut et de liaison de noms de fichiers) doivent être sécurisées pour éviter que les administrateurs CICS émettent ces commandes permettant de définir les paramètres par défaut de la ressource globale.

Quand la transaction est rattachée, la vérification de la sécurité de la ressource CICS, si elle est activée, garantit que l'ID utilisateur est autorisé à exécuter l'ID de transaction.

La vérification de la ressource est contrôlée par l'option RESSEC dans la transaction qui est en cours de fonctionnement, c'est-à-dire le paramètre d'initialisation système RESSEC, et pour le serveur CRD, le paramètre d'initialisation système XPCT.

La vérification de la ressource a lieu uniquement si la valeur du paramètre d'initialisation système XPCT est différente de NO et que l'option RESSEC de la définition TRANSACTION est définie sur YES ou que le paramètre d'initialisation système RESSEC est défini sur ALWAYS.

Les commandes RACF suivantes fournissent un exemple de procédé de protection des transactions du serveur CRD. Voir *RACF Security Guide for CICS* pour plus d'informations relatives à la définition de la sécurité CICS.

- RALTER GCICSTRN SYSADM UACC(NONE) ADDMEM(ADMS)
- PERMIT SYSADM CLASS(GCICSTRN) ID(#cicsadmin)
- RALTER GCICSTRN DEVELOPER UACC(NONE) ADDMEM(ADMI)
- PERMIT DEVELOPER CLASS(GCICSTRN) ID(#cicsdeveloper)
- RALTER GCICSTRN ALLUSER UACC(READ) ADDMEM(ADMR)
- SETROPTS RACLIST(TCICSTRN) REFRESH

## Communication chiffrée à l'aide du protocole SSL

Le chiffrement SSL du flux de données est pris en charge lorsque le client Application Deployment Manager utilise l'interface Web Services pour appeler le serveur CRD. L'utilisation de SSL pour ces communications est contrôlée par le mot clé SSL(YES) dans la définition CICS TCIPSERVICE (voir le document *RACF Security Guide for CICS*).

## Protection des ressources

CICSTS offre la possibilité de protéger les ressources et les commandes permettant de les manipuler. Certaines actions du gestionnaire de déploiement d'application risquent de ne pas aboutir si la sécurité est active mais pas intégralement configurée (autorisation de manipuler les nouveaux types de ressources, par exemple)

En cas d'incident de fonction dans le gestionnaire de déploiement d'application, recherchez dans le fichier journal CICS les messages s'apparentant au suivant, puis procédez à l'action corrective (voir *RACF Security Guide for CICSTS*).

```
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').
```

---

## Utilitaire d'administration

Developer for System z offre l'utilitaire d'administration qui permet aux administrateurs CICS de fournir des valeurs par défaut pour les définitions de ressource CICS. Ces valeurs par défaut peuvent être accessibles en lecture seulement ou peuvent être éditées par le développeur d'application.

L'utilitaire d'administration offre les fonctions suivantes :

- Il fournit le nom CICSplex pour les environnements de test gérés par CICSplex
- Il fournit le nom du groupe de transfert de CICSplex SM
- Il fournit le paramètre de la règle d'exportation des manifestes
- Il fournit les droits d'affichage et les valeurs par défaut des attributs de ressource CICS
- Il fournit la liaison logique-physique CICS utilisée pour les définitions de fichier VSAM

L'utilitaire d'administration est appelé par un modèle de travail ADNJSPAU dans le fichier FEK.#CUST.JCL. L'utilisation de cet utilitaire requiert les droits d'accès UPDATE au référentiel CRD.

ADNJSPAU se trouve dans FEK.#CUST.JCL, sauf si le programmeur système z/OS a indiqué un autre emplacement lorsqu'il a personnalisé et soumis le travail FEK.SFEKSAMP(FEKSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

**Remarque :** Le référentiel CRD doit être fermé dans CICS avant l'exécution du travail ADNJSPAU. Vous pourrez rouvrir le référentiel une fois le travail exécuté. Par exemple, après vous être connecté à CICS, entrez les commandes suivantes pour fermer et ouvrir le fichier :

- CEMT S FILE(ADNREPF0) CLOSED
- CEMT S FILE(ADNREPF0) OPEN

Les instructions de contrôle d'entrée permettent de mettre à jour le référentiel CRD pour un environnement de test CICS dans lequel les règles de syntaxe suivantes sont d'application :

- Un astérisque en position 1 indique une ligne de commentaire.
- Une commande DEFINE doit débuter en position 1, suivie par un espace unique, puis d'un mot clé valide comme TRANSACTION.

- Une valeur de mot clé doit suivre immédiatement un mot clé. Aucun espace n'est autorisé. La seule exception concerne les mots clés d'autorisation d'affichage UPDATE, PROTECT et HIDDEN qui ne possèdent pas de valeurs.
- Les valeurs des mots clés sont placées entre parenthèses.
- Un mot clé et sa valeur doivent se trouver sur la même ligne.

Les exemples de définition ci-dessous suivent la structure des commandes DFHCSDUP (voir le document *CICS Resource Definition Guide for CICSTS*). La seule différence concerne l'insertion des mots clés d'autorisation d'affichage suivants pour regrouper les valeurs d'attribut en trois ensembles de droits :

UPDATE	Les attributs qui suivent ce mot clé peuvent être mis à jour par un développeur d'applications à l'aide de Developer for System z. Il s'agit également de la valeur par défaut pour les attributs omis.
PROTECT	Les attributs qui suivent ce mot clé sont affichés, mais ils sont protégés contre toute mise à jour par un développeur d'applications qui utilise Developer for System z.
HIDDEN	Les attributs qui suivent ce mot clé ne sont pas affichés et sont protégés contre toute mise à jour par un développeur d'applications qui utilise Developer for System z.

Voir l'exemple de code ADNJSPAU .

```

//ADNJSPAU JOB <JOB PARAMETERS>
//*
//ADNSPAU EXEC PGM=ADNSPAU,REGION=1M
//STEPLIB DD DISP=SHR,DSN=FEK.SFEKLOAD
//ADMREP DD DISP=OLD,DSN=FEK.#CUST.ADNREP0
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
*
* CICSplex SM parameters
*
DEFINE CPSMNAME(          )
*DEFINE STAGINGGROUPNAME(ADMSTAGE)
*
* Manifest export rule
*
DEFINE MANIFESTEXPORTRULE(installOnly)
*
* CICS resource definition defaults
* Omitted attributes default to UPDATE.
*
* DB2TRAN default attributes
*
DEFINE DB2TRAN()
    UPDATE DESCRIPTION()
    ENTRY()
    TRANSID()
*
* DOCTEMPLATE default attributes
*
DEFINE DOCTEMPLATE()
    UPDATE DESCRIPTION()
    TEMPLATENAME()
    FILE() TSQUEUE() TDQUEUE() PROGRAM() EXITPGM()
    DDNAME(DFHHTML) MEMBERNAME()
    HFSFILE()
    APPENDCRLF(YES) TYPE(EBCDIC)
*
* File default attributes
*
DEFINE FILE()
    UPDATE DESCRIPTION()
    RECORDSIZE() KEYLENGTH()
    RECORDFORMAT(V) ADD(NO)
    BROWSE(NO) DELETE(NO) READ(YES) UPDATE(NO)
    REMOTESYSTEM() REMOTENAME()
    PROTECT DSNNAME() RLSACCESS(NO) LSRPOOLID(1) STRINGS(1)
    STATUS(ENABLED) OPENTIME(FIRSTREF)
    DISPOSITION(SHARE) DATABUFFERS(2) INDEXBUFFERS(1)
    TABLE(NO) MAXNUMRECS(NOLIMIT)
    READINTEG(UNCOMMITTED) DSNSHARING(ALLREQS)
    UPDATEMODEL(LOCKING) LOAD(NO)
    JNLREAD(NONE) JOURNAL(NO)
    JNLSYNCREAD(NO) JNLUPDATE(NO)
    JNLADD(NONE) JNLSYNCSWRITE(YES)
    RECOVERY(NONE) FWDRECOVLOG(NO)
    BACKUPTYPE(STATIC)
    PASSWORD() NSRGROUP()
    CFDTPOOL() TABLENAME()

```

Figure 59. Utilitaire d'administration ADNJSAPU - CICSSTS (partie 1 de 3)

```

*
* Mapset default attributes
*
DEFINE MAPSET()
    UPDATE  DESCRIPTION()
    PROTECT RESIDENT(NO) STATUS(ENABLED)
           USAGE(NORMAL) USELPACOPY(NO)
** Processtype default attributes
*
DEFINE PROCESSTYPE()
    UPDATE  DESCRIPTION()
           FILE(BTS)
    PROTECT STATUS(ENABLED)
           AUDITLOG() AUDITLEVEL(OFF)
*
* Program default attributes
*
DEFINE PROGRAM()
    UPDATE  DESCRIPTION()
           CEDF(YES) LANGUAGE(LE370)
           REMOTESYSTEM() REMOTENAME() TRANSID()
    PROTECT API(CICSAPI) CONCURRENCY(QUASIRENT)
           DATALOCATION(ANY) DYNAMIC(NO)
           EXECCKEY(USER) EXECUTIONSET(FULLAPI)
           RELOAD(NO) RESIDENT(NO)
           STATUS(ENABLED) USAGE(NORMAL) USELPACOPY(NO)
    HIDDEN JVM(NO) JVMCLASS() JVMPROFILE(DFHJVMPR)
*
* TDQueue default attributes
*
DEFINE TDQUEUE()
    UPDATE  DESCRIPTION()
           TYPE(INTRA)
* Extra partition parameters
    DDNAME() DSNAME()
    REMOTENAME() REMOTESYSTEM() REMOTELength(1)
    RECORDSIZE() BLOCKSIZE(0) RECORDFORMAT(UNDEFINED)
    BLOCKFORMAT() PRINTCONTROL() DISPOSITION(SHR)
* Intra partition parameters
    FACILITYID() TRANSID() TRIGERRLEVEL(1)
    USERID()
* Indirect parameters
    INDIRECTNAME()
    PROTECT WAIT(YES) WAITACTION(REJECT)
* Extra partition parameters
    DATABUFFERS(1)
    SYSOUTCLASS() ERROROPTION(IGNORE)
    OPENTIME(INITIAL) REWIND(LEAVE) TYPEFILE(INPUT)
* Intra partition parameters
    ATIFACILITY(TERMINAL) RECOVSTATUS(NO)

```

*Figure 59. Utilitaire d'administration ADNJSPAU - CICSTS (partie 2 de 3)*

```

*
* Transaction default attributes
*
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 attributes
| *
| 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()
|
| *
| * Optional file name to VSAM data set name binding
| *
| *DEFINE DSBINDING() DSNAME()
| /*

```

Figure 59. Utilitaire d'administration ADNJSPAU - CICSTS (partie 3 de 3)

## Notes de migration de l'utilitaire d'administration

Developer for System z version 7.6.1 bénéficie désormais de la prise en charge d'URIMAP pour l'utilitaire d'administration. Pour pouvoir utiliser la prise en

charge d'URIMAP, vous devez allouer au fichier VSAM du référentiel CRD la taille d'enregistrement maximale de 3000. Jusqu'à Developer for System z version 7.6.1, le modèle du travail d'allocation du référentiel CRD utilise une taille d'enregistrement maximale de 2000.

Les étapes suivantes décrivent l'activation de la prise en charge d'URIMAP, si vous utilisez un ancien référentiel CRD :

1. Créez une sauvegarde de votre référentiel CRD existant, FEK.#CUST.ADNREPF0.
2. Supprimez le référentiel CRD existant.
3. Personnalisez et soumettez le travail FEK.SFEKSAMP(ADNVCRD) pour allouer et initialiser un nouveau référentiel CRD. Pour obtenir les instructions de personnalisation, consultez la documentation qui se trouve dans le membre.
4. Personnalisez et soumettez le travail FEK.SFEKSAMP(ADNJSPAU) pour utiliser l'utilitaire d'administration pour remplir le référentiel CRD.

**Remarque :**

- La migration du référentiel existant n'est pas nécessaire car l'utilitaire d'administration remplace l'intégralité du contenu du référentiel CRD chaque fois qu'il est exécuté.
- Il n'existe aucun problème de compatibilité avec le référentiel CRD. Tous les codes client et hôte Developer for System z pris en charge fonctionneront avec l'une ou l'autre des tailles d'enregistrement maximales. Notez toutefois que la prise en charge d'URIMAP sera désactivée si la taille d'enregistrement maximale n'est pas 3000.

## Messages de l'utilitaire d'administration

Les messages ci-après sont générés par l'utilitaire d'administration dans SYSPRINT DD. Les messages CRAZ1803E, CRAZ1891E, CRAZ1892E, et CRAZ1893E contiennent les codes de statut de fichier ainsi que les codes de retour, de fonction et de commentaire VSAM. Les codes de retour, de fonction et de commentaire de VSAM sont expliqués dans le document *DFSMS Macro Instructions for Data Sets* (SC26-7408). Les codes de statut de fichier sont expliqués dans le document *Enterprise COBOL for z/OS Language Reference* (SC27-1408).

### **CRAZ1800I**

**Exécution achevée avec succès à la ligne <dernier numéro de ligne d'instruction de contrôle>**

**Explication :** L'utilitaire d'administration du programmeur système a été correctement exécuté.

**Réponse de l'utilisateur :** Aucune.

### **CRAZ1801W**

**Exécution achevée avec des avertissements à la ligne <dernier numéro de ligne d'instruction de contrôle>**

**Explication :** L'utilitaire d'administration du programmeur système a fini de traiter un ou plusieurs avertissements détectés lors du traitement des instructions de contrôle.

**Réponse de l'utilisateur :** Vérifiez les autres messages d'avertissement.

### **CRAZ1802E**

**Apparition d'une erreur à la ligne <numéro de ligne>**

**Explication :** L'utilitaire d'administration du programmeur système a rencontré une erreur grave.



**Réponse de l'utilisateur :** Vérifiez les autres messages d'avertissement.

**CRAZ1803E**

**Erreur d'ouverture de référentiel, status=<code de statut du fichier>  
RC=<code retour VSAM> FC=<code de fonction VSAM> FB=<code de  
commentaire VSAM>**

**Explication :** L'utilitaire d'administration du programmeur système a rencontré une erreur grave lors de l'ouverture du référentiel CRD.

**Réponse de l'utilitaire :** Vérifiez les codes de statut, de retour, de fonction et de commentaire VSAM.

**CRAZ1804E**

**Enregistrement d'entrée non reconnu à la ligne <numéro de ligne>**

**Explication :** L'utilitaire d'administration du programmeur système a rencontré une instruction de contrôle d'entrée non reconnue.

**Réponse de l'utilisateur :** Vérifiez qu'une commande **DEFINE** est bien suivie d'un seul espace, puis du mot clé CPSMNAME, STAGINGGROUPNAME, MANIFESTEXPORTRULE, DSBINDING, DB2TRAN, DOCTEMPLATE, FILE, MAPSET, PROCESSTYPE, PROGRAM, TDQUEUE ou TRANSACTION.

**CRAZ1805E**

**Traitement du mot clé <mot clé> à la ligne <numéro de ligne>**

**Explication :** L'utilitaire d'administration du programmeur système traite l'instruction de contrôle d'entrée du mot clé DEFINE.

**Réponse de l'utilisateur :** Aucune.

**CRAZ1806E**

**Règle d'exportation d'un manifeste non valide à la ligne <numéro de ligne>**

**Explication :** L'utilitaire d'administration du programmeur système a rencontré une règle d'exportation de manifeste non valide.

**Réponse de l'utilisateur :** Vérifiez que la valeur du mot clé MANIFESTEXPORTRULE est "installOnly", "exportOnly" ou "both".

**CRAZ1807E**

**Mot clé DSNNAME manquant à la ligne <numéro de ligne>**

**Explication :** L'utilitaire d'administration du programmeur système traitait une instruction de contrôle DEFINE DSBINDING dans laquelle il manque le mot clé DSNNAME.

**Réponse de l'utilisateur :** Vérifiez que l'instruction de contrôle DEFINE DSBINDING contient le mot clé DSNNAME.

**CRAZ1808E**

**Valeur de mot clé non valide pour le mot clé <mot clé> à la ligne  
<numéro de ligne>**

**Explication :** L'utilitaire d'administration du programmeur système traitait une instruction de contrôle DEFINE lorsqu'il a rencontré une valeur non valide pour le mot clé spécifié.

**Réponse de l'utilisateur :** Vérifiez que la longueur et la valeur du mot clé spécifié sont correctes.

**CRAZ1890W**

**Erreur de syntaxe de mot clé à la ligne <numéro de ligne>**

**Explication** : L'utilitaire d'administration du programmeur système traitait une instruction de contrôle DEFINE lorsqu'il a rencontré une erreur de syntaxe pour un mot clé ou sa valeur.

**Réponse de l'utilisateur** : Vérifiez que la valeur du mot clé est placée entre parenthèses et qu'elle est immédiatement suivie du mot clé. Le mot clé et sa valeur doivent se trouver sur la même ligne.

**CRAZ1891W**

**Erreur d'écriture de clé multiple de référentiel, status=<code de statut du fichier> RC=<code retour VSAM> FC=<code de fonction VSAM> FB=<code de commentaire VSAM>**

**Explication** : L'utilitaire d'administration du programmeur système a rencontré une erreur de clé en double lors de l'enregistrement dans le référentiel CRD.

**Réponse de l'utilitaire** : Vérifiez les codes de statut, de retour, de fonction et de commentaire VSAM.

**CRAZ1892W**

**Erreur d'écriture dans le référentiel, status=<code de statut du fichier> RC=<code retour VSAM> FC=<code de fonction VSAM> FB=<code de commentaire VSAM>**

**Explication** : L'utilitaire d'administration du programmeur système a rencontré une grave erreur lors de l'écriture dans le référentiel CRD.

**Réponse de l'utilitaire** : Vérifiez les codes de statut, de retour, de fonction et de commentaire VSAM.

**CRAZ1893W**

**Erreur de lecture du référentiel, status=<code de statut du fichier> RC=<code retour VSAM> FC=<code de fonction VSAM> FB=<code de commentaire VSAM>**

**Explication** : L'utilitaire d'administration du programmeur système a rencontré une erreur grave lors de lecture dans le référentiel CRD.

**Réponse de l'utilitaire** : Vérifiez les codes de statut, de retour, de fonction et de commentaire VSAM.

---

## Chapitre 16. Personnalisation de l'environnement TSO

Cette annexe vous aide à simuler une procédure d'ouverture de session TSO en ajoutant des instructions de définition de données et des fichiers à l'environnement TSO dans Developer for System z.

---

### Service Commandes TSO

Le service Commandes TSO est le composant Developer for System z qui exécute les commandes TSO et ISPF (par lots) et renvoie le résultat au client demandeur. Ces commandes peuvent être demandées implicitement par le produit, ou explicitement par l'utilisateur.

Les exemples de membres fournis avec Developer for System z créent un environnement TSO/ISPF minimal. Si les développeurs doivent accéder à des bibliothèques personnalisées ou tierces, le programmeur système z/OS doit ajouter les instructions de définition de données et les bibliothèques nécessaires à l'environnement du service Commandes TSO. Bien que l'implémentation soit différente dans Developer for System z, la logique sous-jacente est identique à la procédure d'ouverture de session TSO.

**Remarque :** Le service Commandes TSO est un outil de ligne de commande non interactif, par conséquent, les commandes ou les procédures d'invite de saisie de données ou d'affichage de panneaux ISPF ne fonctionnent pas. Un émulateur 3270, comme un émulateur de connexion à l'hôte, qui fait partie du client Developer for System z, est nécessaire à leur exécution.

### Méthodes d'accès

Depuis la version 7.1, Developer for System z permet de choisir le mode d'accès au service Commandes TSO.

- Service de passerelle client TSO/ISPF d'ISPF, qui nécessite un niveau service ISPF minimal. Il s'agit de la méthode par défaut utilisée dans les exemples fournis.
- Transaction APPC (comme dans les éditions précédant la version 7.1).

**Remarque :** Le service de passerelle TSO/ISPF d'ISPF remplace la fonction SCLM Developer Toolkit utilisée dans la version 7.1.

Consultez le fichier `rsed.envvars` pour déterminer la méthode d'accès utilisée pour les hôtes de la version 7.1 ou d'une version supérieure. Si les valeurs par défaut ont été utilisées pendant la procédure de configuration, `rsed.envvars` se trouve dans `/etc/rdz/`.

- Si l'instruction `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` n'est pas présente (ou mise en commentaire, ce qui est le cas par défaut), le service de passerelle TSO/ISPF d'ISPF est utilisé.
- Si l'instruction `_RSE_JAVAOPTS="$_RSE_JAVAOPTS -DTSO_SERVER=APPC"` est présente (et non mise en commentaire), APPC est utilisé.

---

## Utilisation de la méthode d'accès par passerelle client TSO/ISPF

### Personnalisation de base – ISPF.conf

Le fichier de configuration ISPF.conf (situé par défaut dans le répertoire /etc/rdz/) définit l'environnement TSO/ISPF utilisé par Developer for System z. Il existe un seul fichier de configuration ISPF.conf actif, lequel est utilisé par tous les utilisateurs Developer for System z.

La section principale du fichier de configuration définit les noms de définition de données et les concaténations de fichiers associées, comme ceux de l'exemple suivant :

```
sysproc=ISP.SISPLIB,FEK.SFEKPROC
ispm1ib=ISP.SISPMENU
ispt1ib=ISP.SISPTENU
ispp1ib=ISP.SISPPENU
isps1ib=ISP.SISPSLIB
isp11ib=ISP.SISPLOAD
myDD=HLQ1.LLQ1,HLQ2.LLQ2
```

- Chaque définition de données utilise exactement une ligne (il n'y a pas de prise en charge de plusieurs lignes) et il n'y a pas de limites de longueur.
- Il n'y a pas de distinction maj/min dans les définitions et tous les blancs sont ignorés.
- Les lignes mises en commentaire commencent par un astérisque (\*).
- Les noms de définition de données sont suivies d'un signe égale (=), lui-même suivi de la concaténation de fichiers. Les noms de fichier sont séparés par une virgule (,).
- La recherche des concaténations de fichiers s'effectue dans l'ordre où elles sont affichées.
- Les fichiers doivent avoir un nom qualifié complet, sans guillemets (') et sans variables.
- Tous les fichiers sont alloués avec DISP=SHR.
- De nouveaux noms de définition de données peuvent être ajoutés à volonté, mais ils doivent obéir aux règles applicables aux noms de définitions (JCL) et ne pas entrer en conflit avec les autres paramètres de configuration du fichier ISPF.conf. De plus, ISPPROF est alloué dynamiquement DISP=NEW,DELETE) par le service de passerelle client TSO/ISPF.

### Avancé – Utilisation des profils ISPF existants

Par défaut, la passerelle client TSO/ISPF crée un profil ISPF temporaire pour le service Commandes TSO. Cependant, vous pouvez demander à la passerelle client TSO/ISPF z d'utiliser une copie d'un profil ISPF existant. La clé est ici l'instruction `_RSE_CMDSERV_OPTS` du fichier `rsed.envvars`.

```
#_RSE_CMDSERV_OPTS="$_RSE_CMDSERV_OPTS &ISPPROF=&SYSUID..ISPPROF"
```

Supprimez la mise en commentaire de l'instruction (en retirant le signe dièse (#) initial) et fournissez le nom qualifié complet de fichier de données du profil ISPF existant pour utiliser cette fonction.

Les variables suivantes peuvent être utilisées dans le nom du fichier :

- `&SYSUID`. en remplacement de l'ID utilisateur du développeur
- `&SYSPREF`. en remplacement du préfixe TSO du développeur

**Remarque :**

- Si le nom de fichier transmis dans "ISPPROF" n'est pas valide, un profil ISPF vide temporaire est utilisé à la place.
- Le profil ISPF (temporaire et copié) est supprimé à la fin de la session. Les modifications apportées au profil ne sont pas fusionnées dans le profil ISPF existant.

## **Avancé – Utilisation d'une commande exec d'allocation**

L'instruction `allocjob` du fichier `ISPF.conf` (mise en commentaire par défaut) pointe sur une commande `exec` qui peut être utilisée pour fournir des allocations de fichiers supplémentaires par ID utilisateur.

`*allocjob = FEK.#CUST.CNTL(CRAISPRX)`

Supprimez la mise en commentaire de l'instruction (en retirant l'astérisque (\*) initial) et fournissez la référence complète à la commande `exec` d'allocation pour utiliser cette fonction.

- La commande `exec` est exécutée après l'allocation d'ISPPROF et des définitions de données définies dans `ISPF.conf`, mais avant l'initialisation d'ISPF. Vérifiez que la commande `exec` d'allocation n'annule pas ces définitions.
- Un paramètre, l'ID utilisateur du demandeur, est transmis à la commande `exec`.
- Un exemple d'instruction `exec` `CRAISPRX` est fourni dans l'exemple de bibliothèque `FEK.#CUST.CNTL`, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail `FEK.SFEKSAMP(FEKSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

**Remarque :** Du fait que la commande `exec` est appelée avant l'initialisation d'ISPF, vous ne pouvez pas utiliser **VPUT** et **VGET**. Vous pouvez cependant créer votre propre implémentation de ces fonctions à l'aide d'un fichier PDS(E) ou VSAM.

## **Avancé – Utilisation de plusieurs commandes exec d'allocation**

Bien que `ISPF.conf` prenne uniquement en charge l'appel d'une seule commande `exec`, cette dernière peut en revanche appeler une autre commande `exec` sans limite. L'ID utilisateur du client transmis comme paramètre donne la possibilité d'appeler des commandes `exec` d'allocation personnalisées. Vous pouvez, par exemple, vérifier si le membre `USERID'.EXEC(ALLOC)'` existe et l'exécuter.

Un certain nombre de variations de ce scénario permettent d'utiliser les procédures d'ouverture de session TSO :

- Lisez le fichier de configuration propre à l'utilisateur (`USERID'.FEKPROF'`, par exemple).
- Identification de la procédure d'ouverture de session mentionnée dans le fichier.
- Lecture et analyse de la procédure mentionnée à partir de `SYS1.PROCLIB` en vue d'en extraire les instructions de définition de données et les allocations de fichiers.
- Allocation du fichier selon un mode similaire à la procédure d'ouverture de session réelle.

## Avancé – Fichiers ISPF.conf multiples avec configurations Developer for System z multiples

Si les scénarios exec d'allocation décrits ci-dessus ne répondent pas à vos besoins spécifiques, vous pouvez créer différentes instances de serveur de communication RSE Developer for System z, qui utilisent chacune leur propre fichier ISPF.conf. L'inconvénient majeur de la méthode décrite ci-dessous est que les utilisateurs de Developer for System z doivent se connecter à différents serveurs sur le même système hôte pour obtenir l'environnement TSO voulu.

**Remarque :** La création d'une deuxième instance sur le serveur RSE nécessite la duplication et la mise à jour des fichiers de configuration ainsi que des définitions de JCL de démarrage et de tâche démarrées. Aucune nouvelle installation du produit n'est nécessaire, ni aucune duplication de code.

```
$ 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
-> modification : _CMDSERV_CONF_HOME=/etc/rdz/tso2
-> suppression de la mise en commentaire et modification :
-Ddaemon.log=/var/rdz/logs/tso2
-> ajout à la fin :
# -- NECESSAIRE POUR TROUVER LES FICHIERS DE CONFIGURATION RESTANTS
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
$ oedit /etc/rdz/tso2/ISPF.conf
-> modification : modifier si nécessaire
```

Les commandes de l'exemple ci-dessus copient les fichiers de configuration Developer for System z à modifier dans le répertoire qui vient d'être créé, tso2. La variable \_CMDSERV\_CONF\_HOME de rsed.envvars doit être mise à jour pour définir le nouveau répertoire de base ISPF.conf, et daemon.log doit être mis à jour pour définir le nouvel emplacement du fichier journal (créé automatiquement s'il n'existe pas). La mise à jour de la variable CLASSPATH permet de s'assurer que RSE peut localiser les fichiers de configuration qui n'ont pas été copiés dans tso2. Le fichier ISPF.conf peut lui-même être mis à jour pour que vous puissiez l'adapter à vos besoins. Notez que la zone de travail ISPF (variable \_CMDSERV\_WORK\_HOME dans rsed.envvars) peut être partagée entre les deux instances.

Les éléments restants créent une nouvelle tâche démarrée pour RSE qui utilise un nouveau numéro de port et les nouveaux fichiers de configuration /etc/rdz/tso2.

Pour plus d'informations sur les actions présentées ci-dessus, voir les sections correspondantes dans ce document.

---

## Utilisation de la méthode d'accès APPC

### Personnalisation de base – JCL de transaction APPC

La définition d'une transaction APPC est constituée de paramètres APPC et d'un langage JCL de transaction. L'exemple de JCL utilisé pour créer une transaction Developer for System z APPC, FEK.#CUST.JCL(FEKAPPCC), renferme deux options de définition du JCL de transaction, avec et sans prise en charge ISPF.

```
//SYSIN DD DDNAME=SYSINISP * use SYSINTSO or SYSINISP
```

Le client obtient l'environnement TSO/ISPF défini dans le JCL de transaction, de sorte qu'en personnalisant cette section, suivant les règles de définition de données classiques, vous pouvez personnaliser l'environnement pour le client.

```
...
//CMDSERV EXEC PGM=IKJEFT01,DYNAMNBR=50,
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
//SYSPROC DD DISP=SHR,DSN=FEK.SFEKPROC
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
// SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD DUMMY
//MYDD DD DISP=SHR,DSN=HLQ1.LLQ1
// DISP=SHR,DSN=HLQ2.LLQ2
```

**Remarque :** Une transaction APPC existante peut être modifiée à l'aide des panneaux ISPF APPC.

## Avancé – Utilisation des profils ISPF existants

Si la prise en charge d'ISPF est sélectionnée, Developer for System z crée par défaut un profil ISPF temporaire pour le service Commandes TSO. Cependant, vous pouvez demander à Developer for System z d'utiliser une copie d'un profil ISPF existant. Comme décrit dans l'exemple de travail FEK.SFEKSAMP (FEKAPPC), vous devez procéder comme suit :

- Supprimer la mise en commentaire de l'étape COPY dans le JCL de transaction (EXEC et cartes de définition de données associées).
- Modifier &SYSUID..ISPPROF pour qu'il corresponde au nom du fichier de profil ISPF de l'utilisateur.
- Supprimer la mise en commentaire de la première instruction de définition de données ISPPROF dans l'étape CMDSERV et mettez la deuxième en commentaire.

```
...
//COPY EXEC PGM=IEBCOPY * (optional) clone existing ISPF profile
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=&SYSUID..ISPROF//SYSUT2 DD DISP=(MOD,PASS),DSN=&&PROF,
// UNIT=SYSALLDA,
// LIKE=&SYSUID..ISPROF//*
//CMDSERV EXEC PGM=IKJEFT01,DYNAMNBR=50,
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
//*ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//* SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//ISPPROF DD DISP=(OLD,DELETE,DELETE),DSN=&&PROF
```

**Remarque :** Si un nom de fichier non valide est utilisé, le démarrage de la transaction APPC (et, par conséquent, du service Commandes TSO) échouera.

## Avancé – Utilisation d'une commande exec d'allocation

L'exemple de JCL de transaction appelle le service Commandes TSO directement en transmettant son nom (FEKFRSRV) en tant que paramètre au programme IKJEFT01. Vous pouvez effectuer des modifications pour appeler une autre commande exec. Cette commande exec peut effectuer des allocations en fonction de l'ID utilisateur courant, puis appeler le service Commandes TSO.



Contrairement à la méthode d'accès par passerelle client TSO/ISPF, les variables enregistrées dans le profil ISPF de l'utilisateur peuvent être utilisées par cette commande exec en vue de personnaliser l'environnement. N'oubliez pas cependant que les mises à jour de ce profil sont perdues à la fin de la session étant donné que vous utilisez une copie temporaire et non le profil réel.

Notez que l'utilisation d'une commande exec d'allocation dans la transaction APPC n'est pas prise en charge et que la description ci-dessus est conservée en l'état.

## Avancé – Transactions APPC multiples avec configurations Developer for System z multiples

Si vous avez besoin d'environnements TSO multiples, vous pouvez créer différentes instances de serveur de communication RSE Developer for System z, utilisant chacune leur propre transaction APPC. L'inconvénient majeur de la méthode décrite ci-dessous est que les utilisateurs de Developer for System z doivent se connecter à différents serveurs sur le même système hôte pour obtenir l'environnement TSO voulu.

**Remarque :** La création d'une deuxième instance sur le serveur RSE nécessite la duplication et la mise à jour des fichiers de configuration ainsi que des définitions de JCL de démarrage et de tâche démarrées. Aucune nouvelle installation du produit n'est nécessaire, ni aucune duplication de code.

```
$ cd /etc/rdz
$ mkdir /etc/rdz/tso2
$ cp rsed.envvars /etc/rdz/tso2
$ ls /etc/rdz/tso2/
rsed.envvars
$ oedit /etc/rdz/tso2/rsed.envvars
  -> suppression de la mise en commentaire et modification :
_FEFSCMD_TP_NAME=FEKFTS02
  -> suppression de la mise en commentaire et modification :
-Ddaemon.log=/var/rdz/logs/tso2
  -> ajout à la fin :
# -- NECESSAIRE POUR TROUVER LES FICHIERS DE CONFIGURATION RESTANTS
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

Les commandes ci-dessus créent un répertoire tso2 et copie les fichiers de configuration Developer for System z à modifier dans le nouvel emplacement. La variable \_FEFSCMD\_TP\_NAME\_ du fichier rsed.envvars doit être mise à jour pour définir le nouveau nom de la transaction APPC, et daemon.log doit être mis à jour pour définir le nouvel emplacement du journal du démon (créé automatiquement s'il n'existe pas). La mise à jour de la variable CLASSPATH permet de s'assurer que RSE peut localiser les fichiers de configuration qui n'ont pas été copiés dans tso2.



```

//FEKAPPCC JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID//*
//TPADD EXEC PGM=ATBSDFMU
//SYSSDLIB DD DISP=SHR,DSN=SYS1.APPCTP
//SYSSDOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DDNAME=SYSINISP * use SYSINTSO or SYSINISP
//SYSINISP DD DATA,DLM='QT'
TPADD
TPNAME(FEKFTS02)
ACTIVE(YES)
TPSCHED_DELIMITER(DLM1)
KEEP_MESSAGE_LOG(ERROR)
MESSAGE_DATA_SET(&SYSUID..FEKFTS02.&TPDATE..&TPTIME..LOG)
DATASET_STATUS(MOD)
CLASS(A)
JCL_DELIMITER(DLM2)
//FEKFTS02 JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//CMDSERV EXEC PGM=IKJEFT01,DYNAMNBR=50,
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
//SYSPROC DD DISP=SHR,DSN=FEK.SFEKPROC
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
// SPACE=(TRK,(1,1,5)),LRECL=80,RECFM=FB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD DUMMY
//MYDD DD DISP=SHR,DSN=HLQ1.LLQ1
// DISP=SHR,DSN=HLQ2.LLQ2
DLM2
DLM1
QT

```

Figure 60. FEKAPPCC - Création d'une seconde transaction APPC

Créez ensuite une transaction APPC en personnalisant et en soumettant l'exemple de travail FEK.#CUST.JCL(FEKAPPCC), comme indiqué dans l'exemple ci-dessus. Outre la personnalisation normale (décrite dans le code JCL), vous devez associer TPNAME à TPNAME(FEKFTS02) pour qu'il corresponde à la définition `_FEKFSCMD_TP_NAME_` du nouveau fichier `rsed.envvars`. Il est également recommandé de modifier le nom dans la variable `MESSAGE_DATA_SET` et le nom JOB dans le JCL de transaction.

Les éléments restants créent une nouvelle tâche démarrée pour RSE qui utilise un nouveau numéro de port et les nouveaux fichiers de configuration `/etc/rdz/tso2`.

Pour plus d'informations sur les actions présentées ci-dessus, voir les sections correspondantes dans ce document.



---

## Chapitre 17. Exécution de plusieurs instances

Parfois, vous pouvez avoir besoin de plusieurs instances de Developer for System z actives sur un même système, lors du test d'une mise à niveau, par exemple. Cependant, certaines ressources (les ports TCP/IP, par exemple) ne peuvent pas être partagées. Les paramètres par défaut ne sont donc pas toujours applicables. Consultez les informations de cette annexe afin de programmer la coexistence des différentes instances de Developer for System z, pour pouvoir ensuite les personnaliser à l'aide de ce guide de configuration.

Bien qu'il soit possible de partager certaines parties de Developer for System z entre deux instances (ou plus), il est recommandé de NE PAS le faire, sauf si leurs niveaux de logiciel sont identiques et que les seules modifications sont effectuées dans les membres de configuration. Developer for System z offre suffisamment de possibilités de personnalisation pour que plusieurs instances ne se chevauchent pas et nous vous recommandons vivement d'utiliser ces fonctions.

### Remarque :

- FEK et /usr/lpp/rdz correspondent au qualificatif de haut niveau et au chemin d'accès utilisés lors de l'installation du produit. FEK.#CUST, /etc/rdz et /var/rdz désignent les emplacements par défaut utilisés au cours de la personnalisation du produit (voir «Configuration personnalisée», à la page 15 pour de plus amples informations).
- Il est recommandé d'installer Developer for System z dans un système de fichiers privé (HFS ou zFS) pour faciliter le déploiement des composants z/OS UNIX du produit.
- Si vous ne pouvez pas utiliser un système de fichiers privé, il est recommandé d'utiliser un outil d'archivage (la commande z/OS UNIX tar, par exemple) pour transférer les répertoires z/OS UNIX d'un système à un autre. Il s'agit de préserver les attributs (tels que le contrôle par programme) des fichiers et répertoires Developer for System z.

Pour plus d'informations sur les exemples de commande ci-dessous qui permettent d'archiver et de restaurer le répertoire d'installation de Developer for System z, voir le document *UNIX System Services Command Reference* (SA22-7802)

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

---

## Configuration identique par sysplex

Les fichiers de configuration (et le code) Developer for System z peuvent être partagés entre différents systèmes dans un sysplex, chaque système exécutant sa propre copie identique de Developer System z, si de nouvelles instructions sont respectées.

- Il est recommandé de placer les fichiers journaux dans des emplacements uniques, afin d'éviter qu'un système n'écrase les données d'un autre. En acheminant les fichiers journaux z/OS UNIX vers des emplacements spécifiques avec les directives `daemon.log` et `user.log` de `rsed.envvars`, vous pouvez

partager les fichiers de configuration si vous montez un système de fichiers z/OS UNIX spécifique du système dans le chemin d'accès indiqué. De cette manière, tous les fichiers journaux sont écrits dans le même emplacement logique, mais compte tenu du système de fichiers non partagé du dessous, ils sont placés dans des emplacements physiques différents.

- Les répertoires type de configuration (/etc/rdz/ and /var/rdz/projects/, par exemple) peuvent être partagés par l'intermédiaire d'un sysplex, étant donné que Developer for System z les utilise en mode lecture seule.
- Les répertoires de données temporaires (/tmp/ and /var/rdz/WORKAREA/, par exemple) doivent être uniques par système, étant donné que les noms de fichier temporaire ne sont pas compatible avec sysplex.
- Si vous partagez le code, il est également recommandé de partager les fichiers de configuration pour vous assurez que tous les systèmes sont synchronisés après les opérations de maintenance.

---

## Niveaux de logiciels identiques, fichiers de configuration différents

Dans un nombre limité de circonstances, vous pouvez partager tout sauf (certains) des composants personnalisables. La mise à disposition d'un accès non-SSL pour un utilisation sur site, et d'une communication encodée SSL pour une utilisation hors site est un exemple.

**Avvertissement :** La configuration partagée NE peut PAS être utilisée de manière sûre pour tester la maintenance, ou effectuer une prévisualisation technique ou une nouvelle édition.

Pour configurer une autre instance d'une installation active de Developer for System z, suivez de nouveau la procédure de personnalisation pour les composants qui sont différents, en utilisant d'autres fichiers, répertoires et ports afin d'éviter un chevauchement avec la configuration en cours.

Dans l'exemple SSL mentionné ci-dessus, la configuration du démon RSE en cours peut être clonée, après quoi la configuration clonée peut être mise à jour. Le JCL de démarrage du démon RSE peut ensuite être cloné et personnalisé avec un nouveau port TCP/IP et l'emplacement des fichiers de configuration mis à jour. Les personnalisations du système MVS (moniteur de travaux JES etc.) peuvent être partagées entre les instances SSL et les instances non-SSL. Il en résulte les actions suivantes :

```
$ 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
  -> Suppression de la mise en commentaire et modification :
-Ddaemon.log=/var/rdz/logs/ssl
  -> ajout à la fin :
    # -- NECESSAIRE POUR TROUVER LES FICHIERS DE CONFIGURATION RESTANTS
    CFG_BASE=/etc/rdz
    CLASSPATH=.:$CFG_BASE:$CLASSPATH
    # --
$ oedit /etc/rdz/ssl/ssl.properties
  -> modification : modifier si nécessaire
```

Les commandes ci-dessus copient les fichiers de configuration Developer for System z à modifier dans le répertoire qui vient d'être créé, ssl. La variable

daemon.log du fichier rsed.envvars doit être mise à jour pour définir le nouvel emplacement du journal du journal (créé automatiquement s'il n'existe pas). La mise à jour de la variable CLASSPATH permet de s'assurer que RSE peut localiser les fichiers de configuration qui n'ont pas été copiés dans ssl. Le fichier ssl.properties peut lui-même être mis à jour en fonction de vos besoins.

Les éléments restants créent une nouvelle tâche démarrée pour RSE qui utilise un nouveau numéro de port et les nouveaux fichiers de configuration /etc/rdz/ssl.

Pour plus d'informations sur les actions présentées ci-dessus, voir les sections correspondantes dans ce document.

---

## Dans tous les autres cas

Lorsque des modifications de code sont effectuées (maintenance, prévisualisation technique, nouvelle édition) ou que les modifications que vous effectuez sont complexes, il est recommandé de procéder à une autre installation de Developer for System z. La présente section décrit les points possibles de conflit entre différentes installations.

La liste ci-dessous décrit brièvement les éléments qui doivent être différents entre les instances de Developer for System z (fortement recommandé) :

- SMP/E CSI
- Bibliothèques d'installation
- Port TCP/IP du moniteur de travaux JES, ainsi que son fichier de configuration FEJJCNFG
- Langage JCL d'initialisation du moniteur de travaux JES
- Nom de transaction APPC
- Fichier de configuration RSE rsed.envvars, \*.properties et \*.settings
- Port TCP/IP RSE
- Langage JCL de démarrage de RSE

Vous trouverez ci-dessous une présentation plus détaillée :

- SMP/E CSI
  1. Installez chaque instance de Developer for System z dans un CSI distinct. SMP/E empêchera une seconde installation du même FMID dans un CSI mais acceptera l'installation d'un autre FMID. Si le second FMID correspond à une version plus récente, elle remplace la version existante du produit. Si le second FMID est une version plus ancienne, l'installation échoue en raison de noms de partie en double.
- Bibliothèques d'installation
  1. Installez chaque instance de Developer for System z dans des fichiers et des répertoires différents. N'oubliez pas que vous pouvez uniquement modifier le chemin z/OS UNIX en indiquant un préfixe pour le chemin /usr/lpp/rdz fourni par défaut par IBM . Voici un exemple valide : /service/usr/lpp/rdz .
  2. Le travail de définition de la configuration FEK.SFEKSAMP(FEKSETUP) crée les fichiers et les répertoires utilisés pour enregistrer les fichiers de configuration. Les fichiers de configuration doivent être uniques et c'est la raison pour laquelle vous devez utiliser des noms de fichiers et de répertoires uniques lorsque vous soumettez ce travail pour éviter de remplacer les personnalisations existantes.

- Composants obligatoires
  1. Le fichier de configuration du moniteur de travaux JES  
FEK.#CUST.PARMLIB(FEJJCNFG) contient son numéro de port TCP/IP et ne peut par conséquent pas être partagé. Le membre lui-même peut être renommé (si le langage JCL est également mis à jour), afin de pouvoir placer toutes les versions personnalisées de ce membre dans un même fichier, si vous n'effectuez pas les mises à jour dans le fichier d'installation.
  2. Le langage JCL de démarrage du moniteur de travaux JES  
FEK.#CUST.PROCLIB(JMON) se rapporte à FEJJCNFG et ne peut pas être partagé non plus. Après avoir renommé le membre (et la carte de travail si vous l'avez démarrée en tant que travail), vous pouvez placer tous ceux du langage JCL dans le même fichier.
  3. Le fichier de configuration RSE /etc/rdz/rsed.envvars contient des références au chemin d'installation et éventuellement au chemin du journal du serveur, qui doit être unique. Le nom de fichier est obligatoire, vous ne pouvez donc pas conserver les différentes copies dans le même répertoire.
  4. Le fichier de configuration ISPF.conf contient une référence à FEK.SFEKPROC(FEKFRSRV), le serveur de commandes TSO. Elle est propre au niveau du logiciel, par conséquent, vous devez créer un fichier ISPF.conf.
  5. Tous les autres fichiers de configuration z/OS UNIX (\*.properties, par exemple) doivent résider dans le même répertoire que rsed.envvars et ne peuvent donc pas être partagés, étant donné que rsed.envvars doit se trouver dans un emplacement non partagé.
  6. Le langage JCL de démarrage de FEK.#CUST.PROCLIB(RSED) ne peut pas être partagé car il définit le numéro de port TCP/IP et possède une référence aux répertoires d'installation et de configuration, qui doit être unique. Après avoir renommé le membre (et la carte de travail si vous l'avez démarrée en tant que travail), vous pouvez placer tous ceux du langage JCL dans le même fichier.
  7. Le langage JCL de démarrage du démon lock FEK.#CUST.PROCLIB(LOCKD) ne peut pas être partagé étant donné qu'il comporte une référence aux répertoires d'installation et de configuration, qui doivent être uniques. Après avoir renommé le membre (et la carte de travail si vous l'avez démarrée en tant que travail), vous pouvez placer tous les langages JCL dans le même fichier.
- Composants facultatifs
  1. Les ports TCP/IP REXEC et SSH peuvent être partagés sans aucune restriction.
  2. La transaction APPC contient une référence à FEK.SFEKPROC(FEKFRSRV), le serveur de commandes TSO. Elle est propre au niveau du logiciel, par conséquent, vous devez créer une transaction APPC par instance. Notez que, dans la mesure où le nom de la transaction APPC est modifié, la variable \_FEKFSCMD\_TP\_NAME\_ doit être définie dans rsed.envvars.
  3. Certaines procédures ELAXF\* font référence à hlq.SFEKLOAD, la bibliothèque de chargement de Developer for System z. Pour savoir comment créer différents ensembles qui soient disponibles pour les utilisateurs, consultez les annotations sur JCLLIB dans «Procédures de construction à distance ELAXF\*», à la page 25.
  4. Pour activer deux instances de la procédure mémorisée DB2, vous devez procéder comme suit. Notez toutefois qu'il s'agit d'une description présentée en l'état, non prise en charge :
    - a. Copiez hlq.SFEKPROC(ELAXMREX) dans un membre appelé différemment, par exemple, ELAXMRXX.

- b. Copiez le modèle de membre hlq.SFEKSAMP(ELAXMSAM) dans un membre appelé différemment, par exemple, ELAXMWDZ.
- c. Modifiez le modèle de membre hlq.SFEKSAMP(ELAXMJCL) pour refléter ces modifications de dénomination, par exemple :

```
//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. Poursuivez la personnalisation décrite dans «(Facultatif) Procédure mémorisée DB2», à la page 91, mais avec les nouveaux membres.
  - e. Le nouveau nom de l'environnement WLM (ELAXMWDZ par exemple) doit être utilisé dans l'assistant de la procédure mémorisée DB2 du client.
5. La prise en charge bidirectionnelle dans des régions CICS dépend d'un membre de bibliothèque de chargement et par conséquent n'est pas partageable entre les éditions. Toutefois, si le nom du module de chargement est identique pour toutes les instances, vous pouvez partager la version la plus récente entre les instances, et même entre les éditions. La compatibilité amont est indisponible si le nom du module de chargement a été modifié.
  6. Les modules de chargement du gestionnaire de déploiement d'application inclus dans des régions CICS présentent une compatibilité amont et la version la plus récente peut ainsi être partagée entre les éditions.
  7. La méthode d'accès VSAM CRD du gestionnaire de déploiement d'application présente une compatibilité amont, et la version la plus récente peut ainsi être partagée entre les éditions.
  8. Les définitions de ressource CICS du gestionnaire de déploiement d'application présentent une compatibilité amont, la version la plus récente pouvant ainsi être partagée entre les éditions.
  9. Les méthodes d'accès VSAM CARMA peuvent changer d'un niveau de logiciel à l'autre. Il est donc recommandé de ne pas les partager.





---

## Chapitre 18. Guide de migration

---

### Remarques relatives à la migration

La présente section met en évidence les modifications de l'installation et de la configuration par rapport aux précédentes éditions du produit. Elle fournit également des instructions générales pour la migration de cette édition. Pour de plus amples informations, consultez les sections associées dans ce manuel.

- Si vous disposez de la version précédente d'IBM Rational Developer for System z, d'IBM WebSphere Developer for System z, de IBM WebSphere Developer for zSeries ou d'IBM WebSphere Studio Enterprise Developer, il est recommandé de sauvegarder les fichiers personnalisés associés AVANT d'installer la mise à niveau vers cette version d'IBM Rational Developer for System z Version 7.6.1.
- Voir aussi Chapitre 17, «Exécution de plusieurs instances», à la page 277 si vous prévoyez d'exécuter plusieurs instances de Developer for System z.

**Remarque :** Les informations de migration indiquées ici s'appliquent aux versions de Developer for System z toujours prises en charge au moment de la publication.

### Sauvegarde des fichiers déjà configurés

Si vous disposez d'une version antérieure de Developer for System z, il est recommandé de sauvegarder les fichiers personnalisés associés avant d'installer cette version d'IBM Developer for System z.

Les fichiers personnalisables de Developer for system z sont disponibles dans les emplacements suivants :

- Version 7.5
  - FEK.SFEKSAMP, certains membres sont copiés dans FEK.#CUST.\* par l'exemple de travail FEKSETUP, où \* correspond à PARMLIB, PROCLIB, JCL, CNTL, ASM et COBOL
  - FEK.SFEKSAMV
  - /usr/lpp/rdz/samples/, certains fichiers sont copiés dans /etc/rdz/ et /etc/rdz/sclmdt/\* par l'exemple de travail FEKSETUP, où \* correspond à CONFIG/, CONFIG/PROJECT/ et CONFIG/script/
- Version 7.1
  - FEK.SFEKSAMP
  - CRA.SCRASAMP
  - /usr/lpp/wd4z/rse/lib/, il est recommandé de copier les fichiers personnalisables dans /etc/wd4z/
- Version 7.0
  - FEK.SFEKSAMP
  - CRA.SCRASAMP
  - /usr/lpp/wd4z/rse/lib/, il est recommandé de copier les fichiers personnalisables dans /etc/wd4z/

Les configurations précédentes de Developer for system z documentent également les modifications apportées aux fichiers de configuration d'autres produits.

- Version 7.5

- SYS1.PARMLIB(ASCHPMxx)  
définissez une classe de transaction APPC du service Commandes TSO
- SYS1.PARMLIB(BPXPRMxx)  
définissez les valeurs par défaut du système z/OS UNIX
- SYS1.PARMLIB(COMMNDxx)  
démarrez les serveurs lors de l'initialisation
- SYS1.PARMLIB(LPALSTxx)  
ajoutez FEK.SFEKLPA à la zone LPA
- SYS1.PARMLIB(PROGxx)  
accordez l'autorisation APF à FEK.SFEKAUTH  
ajoutez FEK.SFEKAUTH et FEK.SFEKLOAD à LINKLIST
- (APPC)  
définissez une transaction APPC pour le service Commandes TSO
- (WLM)  
associez un programme transactionnel APPC à un groupe de performances TSO
- (WLM)  
associez un environnement d'application à une procédure mémorisée DB2
- Version 7.1
  - SYS1.PARMLIB(ASCHPMxx)  
définissez une classe de transaction APPC du service Commandes TSO
  - SYS1.PARMLIB(BPXPRMxx)  
définissez les valeurs par défaut du système z/OS UNIX
  - SYS1.PARMLIB(PROGxx)  
accordez l'autorisation APF à FEK.SFEKLOAD
  - /etc/services  
définissez le port du démon RSE
  - /etc/inetd.conf  
définissez le service du démon RSE
  - /etc/SCLMDT/CONFIG/ISPF.conf  
définissez l'emplacement du serveur de commandes TSO
  - (APPC)  
définissez une transaction APPC pour le service Commandes TSO
  - (WLM)  
associez un programme transactionnel APPC à un groupe de performances TSO
  - (WLM)  
associez un environnement d'application à une procédure mémorisée DB2
- Version 7.0
  - SYS1.PARMLIB(ASCHPMxx)  
définissez une classe de transaction APPC du service Commandes TSO
  - SYS1.PARMLIB(BPXPRMxx)  
définissez les valeurs par défaut du système z/OS UNIX
  - SYS1.PARMLIB(PROGxx)  
accordez l'autorisation APF à FEK.SFEKLOAD

- /etc/services  
définissez le port du démon RSE
- /etc/inetd.conf  
définissez le service du démon RSE
- (APPC)  
définissez une transaction APPC pour le service Commandes TSO
- (WLM)  
associez un programme transactionnel APPC à un groupe de performances TSO
- (WLM)  
associez un environnement d'application à une procédure mémorisée DB2

---

## Notes de migration relatives à la Version 7.6.1

Les notes de migration suivantes sont spécifiques de la version 7.6.1. Elles sont valides pour la migration depuis la version 7.6 ou viennent s'ajouter aux notes de migration vers la version 7.6 existante.

- Gestionnaire de déploiement d'application - Les modules ADN\* existants dans la concaténation CICS RPL doivent être mis à jour.
- Gestionnaire de déploiement d'application - Les exemples de membre suivants ont été mis à jour pour ajouter la prise en charge d'URIMAP à l'utilitaire d'administration :
  - ADNJSPAU
  - ADNVCRD
- Gestionnaire de déploiement d'application - Vous devez remplacer un VSAM du référentiel CRD pour activer la prise en charge d'URIMAP.
- CARMA - Prise en charge ajoutée pour un agencement de longueur variable pour le fichier VSAM d'informations personnalisées CARMA, CRASTRS
- CARMA - De nouveaux exemples de membre ont été ajoutés :
  - CRA#VS2 - migration de CRASTRS vers un format de longueur variable
- Moniteur de travaux JES - Utilisation de \_CEE\_ENVFILE\_S dans le JCL de la tâche démarrée.
- Moniteur de travaux JES - Les directives FEJJCNGF suivantes deviennent facultatives :
  - HOST\_CODEPAGE
- PROCLIB - De nouveaux membres PROCLIB ont été ajoutés
  - ELAXFDCL
- RSE - Utilisation de Java 64 bits maintenant prise en charge.
- RSE - Nouvelles commandes de l'opérateur ajoutées (depuis la version 7.6.1.0) :
  - MODIFY DISPLAY PROCESS,DETAIL
- RSE - Les directives non-personnalisables suivantes ont été changées ou sont nouvelles dans rsed.envvars (depuis la 7.6.0.0):
  - (\_RSE\_JAVAOPTS) -DDSTORE\_KEEPALIVE\_RESPONSE\_TIMEOUT
  - (\_RSE\_JAVAOPTS) -DDSTORE\_IO\_SOCKET\_READ\_TIMEOUT
  - (\_RSE\_JAVAOPTS) -DRSECOMM\_LOGFILE\_MAX
- RSE - De nouvelles directives facultatives ont été ajoutées à rsed.envvars (depuis les versions 7.6.0.0 et 7.6.0.1):
  - (\_RSE\_JAVAOPTS) -Denable.automount

- |                   - (\_RSE\_JAVAOPTS) -Ddeny.nozero.port
- |                   - (\_RSE\_JAVAOPTS) -Dsingle.logon
- |                   - (\_RSE\_JAVAOPTS) -Dprocess.cleanup.interval
- |       • RSE - Les messages de console suivants ont été changés ou sont nouveaux
- |        (depuis les versions 7.6.0.1 et 7.6.1.0) :
- |                   - FEK001I
- |                   - FEK210I

---

## Migration de la version 7.5 vers la version 7.6

### IBM Rational Developer for System z, FMID HHOP760

- L'emplacement d'installation SMP/E par défaut pour les composants MVS et z/OS UNIX n'a pas été modifié et correspond toujours à FEK.\* et à /usr/lpp/rdz/.\*.
- Gestionnaire de déploiement d'application - Les modules ADN\* existants dans la concaténation CICS RPL doivent être mis à jour.
- Gestionnaire de déploiement d'application - De nouveaux modules de chargement, qui doivent faire partie de la concaténation RPL CICS, ont été ajoutés pour prendre en charge l'interface RESTful CICS.
  - ADNANAL
  - ADNCRD41
  - ADNREST
- Gestionnaire de déploiement d'application - De nouveaux exemples de membres ont été ajoutés pour prendre en charge l'interface RESTful CICS.
  - ADNCSDRS
  - ADNCSDTX
  - ADNTXNC
- Gestionnaire de déploiement d'application - Les exemples de membres existants sont renommés.
  - ADNARCSD -> ADNCSDAR
  - ADNCMSGH -> ADNMSGHC
  - ADNMFEST -> ADNVMFST
  - ADNPCCSD -> ADNCSDWS
  - ADNSMSGH -> ADNMSGHS
  - ADNVSAM -> ADNVCRD
- Un nouveau type de production RAM est fourni pour accéder à CA Endevor SCM.
  - CRARNDVR
- CARMA - De nouveaux exemples de membres ont été fournis pour prendre en charge CA Endevor SCM RAM.
  - FEK.#CUST.JCL(CRA#VCAD)
  - FEK.#CUST.JCL(CRA#VCAS)
  - FEK.#CUST.CNTL(CRASUBCA)
  - FEK.#CUST.PARMLIB(CRASHOW)
  - FEK.#CUST.PARMLIB(CRATMAP)
  - FEK.SFEKPROC(CRANDVRA)
  - /etc/rdz/crstart.endevor.conf

- CARMA - De nouveaux exemples de membres ont été fournis pour prendre en charge la fusion des définitions RAM.
  - CRA#UADD
  - CRA#UQRY
- File Manager Integration - L'interface de traitement par lots permettant d'accéder à File Manager n'est plus prise en charge.
- File Manager Integration - Le fichier de configuration FMEXT.properties a été intégralement modifié et doit être remplacé.
- Moniteur de travaux JES - Les options LE sont imbriquées dans le module de chargement FEJMON (depuis la version 7.5.0.1), ce qui peut nécessiter la modification de la définition de tâche démarrée. Pour plus d'informations, voir l'exemple de JCL FEK.SFEKSAMP(FEJJJCL).
- Moniteur de travaux JES - De nouvelles directives facultatives ont été ajoutées à FEJCNFG (dans la version 7.5.0.1 et 7.5.1.0).
  - APPLID
  - CONSOLE\_NAME
  - GEN\_CONSOLE\_NAME
- Moniteur de travaux JES - Une nouvelle commande, Afficher JCL, est prise en charge (à partir de la version 7.5.1.0) et peut nécessiter la mise à jour de votre logiciel de sécurité.
- Démon lock – Le démon lock (LOCKD) est une nouvelle tâche démarrée (à partir de la version 7.5.0.1). Cette tâche démarrée peut être interrogée pour identifier le client Developer for z qui détient le verrou d'un fichier. (Les commandes système s'arrêtent au niveau de l'espace adresse, qui correspond au pool d'unités d'exécution RSE.)
- SCLMDT - L'emplacement par défaut des fichiers de configuration du projet SCLMDT a été modifié.
  - /var/rdz/sclmdt
- RSE - De nouvelles commandes de l'opérateur ont été ajoutées.
  - MODIFY RSESTANDARDLOG
- RSE - De nouvelles directives obligatoires ont été ajoutées à rsed.envvars (dans la version 7.5.0.1 et 7.6.0.0).
  - \_RSE\_LOCKD\_PORT
  - (\_RSE\_JAVAOPTS) -Dlock.daemon.port
  - (\_RSE\_JAVAOPTS) -Dlock.daemon.cleanup.interval
  - \_RSE\_LOCKD\_CLASS
  - \_RSE\_HOST\_CODEPAGE
  - (\_RSE\_JAVAOPTS) -Dfile.encoding
  - (\_RSE\_JAVAOPTS) -Dconsole.encoding
- RSE - De nouvelles directives facultatives ont été ajoutées à rsed.envvars (à partir des versions 7.5.0.1, 7.5.1.0 et 7.6.0.0).
  - (\_RSE\_JAVAOPTS) -Duser.log
  - (\_RSE\_JAVAOPTS) -Dkeep.last.log
  - (\_RSE\_JAVAOPTS) -Denable.standard.log
  - (\_RSE\_JAVAOPTS) -DDSTORE\_LOG\_DIRECTORY
  - (\_RSE\_JAVAOPTS) -DHIDE\_ZOS\_UNIX
  - (\_RSE\_JAVAOPTS) -Denable.certificate.mapping
  - GSK\_CRL\_SECURITY\_LEVEL

- GSK\_LDAP\_SERVER
- GSK\_LDAP\_PORT
- GSK\_LDAP\_USER
- GSK\_LDAP\_PASSWORD
- RSE - Certaines directives facultatives ont été modifiées dans `rsed.envvars`.
  - (`_RSE_JAVAOPTS`) -Ddaemon.log
  - (`_RSE_JAVAOPTS`) -Xmx
  - `SCLMDT_CONF_HOME`
- RSE - De nouvelles directives facultatives ont été ajoutées à `ssl.properties` (depuis la version 7.5.1.0 et 7.6.0.0).
  - `server_keystore_label`
  - `server_keystore_type`
- RSE - Le démon RSE prend en charge l'authentification des certificats client X.509 (à partir de la version 7.5.1.0), qui requiert des mises à jour du certificat et de la configuration de sécurité en cours.
- RSE - La sécurité a été renforcée, échec des demandes de connexion avec erreurs PassTicket et FEKAPPL.
- RSE - L'emplacement par défaut de tous les fichiers journaux (du démon et de l'utilisateur) a été modifié.
  - `/var/rdz/logs`
  - `/var/rdz/logs/$LOGNAME`
- RSE - Un nouvel exemple de JCL a été fourni afin de rassembler les informations relatives aux journaux et à la configuration de Developer for System z.
  - FEKLOGS

## Fichiers configurables

Le tableau 47 présente les fichiers qui sont personnalisés dans la version 7.6. Notez que les exemples de bibliothèque Developer for System z, FEK.SFEKSAMP, FEK.SFEKSAMPV et `/usr/lpp/rdz/samples/`, sont fournies avec davantage de membres personnalisables que ceux répertoriés ici (le code source CARMA et les travaux pour leur compilation, par exemple).

**Remarque :** L'exemple de travail FEKSETUP copie tous les membres répertoriés dans des fichiers et répertoires différents, par défaut FEK.#CUST.\* et `/etc/rdz/`.

Tableau 47. Personnalisations de la version 7.6

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL permettant de créer des fichiers et des répertoires et de les remplir avec des fichiers personnalisables	Mis à jour pour inclure de nouveaux membres personnalisables
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	Langage JCL pour le moniteur de travaux JES	Option ajoutée pour modifier les options LE
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Nom fourni pour le membre JMON	Voir membre JMON

Tableau 47. Personnalisations de la version 7.6 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
RSED	FEK.SFEKSAMP (FEKRSED) [FEK.#CUST.PROCLIB]	Langage JCL pour le démon RSE	none
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Nom fourni pour le membre RSED	Voir membre RSED
LOCKD	FEK.SFEKSAMP (FEKLOCKD) [FEK.#CUST.PROCLIB]	JCL pour le démon lock	NOUVEAU, doit être personnalisé
FEKLOCKD	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(LOCKD)]	Nom fourni pour le membre LOCKD	Voir membre LOCKD
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	Code JCL pour les générations de projets distants, etc.	none
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL pour les définitions de sécurité	Mises à jour mineures
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Fichier de configuration du moniteur de travaux JES	De nouvelles directives facultatives ont été ajoutées
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Langage JCL pour les soumissions TSO	none
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	Le langage JCL doit créer la méthode d'accès VSAM au message du gestionnaire CARMA	none
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	Le langage JCL doit créer la méthode d'accès VSAM à la configuration du gestionnaire CARMA	none
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	Le langage JCL doit créer la méthode d'accès VSAM aux informations personnalisées du gestionnaire CARMA	none
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Démarrage par lots CARMA CLIST	none
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Liste de commandes de démarrage du traitement par lots CARMA pour CA Endevor SCM RAM	NOUVEAU, la personnalisation est facultative
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endevor SCM RAM	NOUVEAU, la personnalisation est facultative
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endevor SCM RAM	NOUVEAU, la personnalisation est facultative
CRANDVRA	FEK.SFEKPROC	REXX d'allocation CARMA pour CA Endevor SCM RAM	NOUVEAU, la personnalisation est facultative

Tableau 47. Personnalisations de la version 7.6 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
CRAISPRX	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Exemple de commande exec d'allocation de définition de données pour CARMA à l'aide de la passerelle client TSO/ISPF	none
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de création de la méthode d'accès VSAM au message de la RAM du SCLM	none
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de création de fichiers de la RAM du SCLM	none
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de création de la méthode d'accès VSAM au message de la RAM du PDS	none
CRA#CRAM	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de compilation de la RAM du squelette	none
CRA#VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de création du VSAM de configuration CARMA pour CA Endevor SCM RAM	NOUVEAU, la personnalisation est facultative
CRA#VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de création du VSAM de personnalisation des informations CARMA pour CA Endevor SCM RAM	NOUVEAU, la personnalisation est facultative
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de fusion des définitions RAM	NOUVEAU, la personnalisation est facultative
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL d'extraction des définitions RAM	NOUVEAU, la personnalisation est facultative
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Exemple de code source pour le remplacement de IRXJCL	none
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de compilation de CRAXJCL	none
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de définition du serveur CRD RESTful dans la région CICS primaire	NOUVEAU, la personnalisation est facultative
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de définition d'autres ID transaction dans la région CICS	NOUVEAU, la personnalisation est facultative
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de création d'autres ID transaction	NOUVEAU, la personnalisation est facultative



Tableau 47. Personnalisations de la version 7.6 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de compilation d'ADNMSGHS	Renommé. S'appelait ADNCMSGH
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Exemple de code source pour le gestionnaire de message de pipeline	Renommé. S'appelait ADNSMSGH
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de création du référentiel CRD	Renommé. S'appelait ADNVSAM
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de définition du serveur CRD du service Web dans la région CICS primaire	Renommé. S'appelait ADNPCCSD
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL définissant le serveur CRD dans les régions CICS secondaires	Renommé. S'appelait ADNARCSD
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de mise à jour des valeurs par défaut de CRD	Ajout de définitions pour le service RESTful. Personnalisations à refaire
ADNMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de création et de définition du référentiel de manifestes	Renommé. S'appelait ADNMFEST
ELAXMSAM	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	Procédure JCL de l'espace adresse WLM du compilateur de procédures mémorisées PL/I et COBOL.	none
ELAXMJCL	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de définition du générateur de procédures mémorisées PL/I et COBOL dans DB2	none
FEKAPPC	FEK.SFEKSAMP [FEK.#CUST.JCL]	Le langage JCL doit créer une transaction APPC	none
FEKAPPCL	FEK.SFEKSAMP [FEK.#CUST.JCL]	Le langage JCL doit afficher une transaction APPC	none
FEKAPPCX	FEK.SFEKSAMP [FEK.#CUST.JCL]	Le langage JCL doit supprimer une transaction APPC	none
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de collecter les journaux	NOUVEAU, la personnalisation est facultative
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	Variables d'environnement RSE	Les copies plus anciennes doivent être remplacées par celle-ci (les personnalisations doivent être effectuées une nouvelle fois)

Tableau 47. Personnalisations de la version 7.6 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la passerelle client TSO/ISPF	ISP.SISPLIB ajouté à SYSPROC pour SCLMDT
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration du gestionnaire CARMA	none
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRASTART	none
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRASTART pour CA Endevor SCM RAM	NOUVEAU, la personnalisation est facultative
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration SSL RSE	De nouvelles directives facultatives ont été ajoutées
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de trace RSE	none
propertiescfg.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration des groupes de propriétés résidant sur l'hôte	none
projectcfg.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de projets résidant sur l'hôte	none
FMIEXT.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration d'intégration de File Manager	Les copies plus anciennes doivent être remplacées par celle-ci (les personnalisations doivent être effectuées une nouvelle fois)
uchars.settings	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration des caractères non éditables	none

## Migration de la version 7.1 vers la version 7.5

### IBM Rational Developer for System z, FMID HHOP750

- L'emplacement d'installation SMP/E par défaut des composants MVS n'a pas été modifié et correspond toujours à FEK.\*.
- L'emplacement d'installation SMP/E par défaut des composants z/OS UNIX est devenu /usr/lpp/rdz/\*.
- Common Access Repository Manager (CARMA) a été fusionné avec Developer for System z version 7.5 ; il n'est donc plus nécessaire de l'installer comme produit distinct.
- SCLM Developer Toolkit a été fusionné avec Developer for System z version 7.5 ; il n'est donc plus nécessaire de l'installer comme produit distinct.

- Dans la version 7.5, le service de passerelle TSO/ISPF d'ISPF remplace la fonction SCLM Developer Toolkit utilisée dans la version 7.1 pour se connecter au service Commandes TSO. La méthode de connexion APPC est toujours prise en charge.
- Dans la version 7.5, le serveur RSE n'est plus un processus géré par INETD mais une tâche démarrée. Le serveur RSE utilise maintenant un seul modèle de serveur, tandis que dans les versions précédentes, chaque connexion client-hôte utilisait un serveur RSE privé.
- Tous les modules qui nécessitent une autorisation de l'APF (moniteur de travaux JES et SCLM Developer Toolkit) sont déplacés vers FEK.SFEKAUTH dans la version 7.5, ce qui requiert une mise à jour des définitions APF existantes.
- Le module de chargement du moniteur de travaux JES est déplacé vers FEK.SFEKAUTH dans la version 7.5, ce qui requiert une mise à jour de la procédure de tâche démarrée existante.
- Les modules de chargement CARMA sont déplacés vers de nouvelles bibliothèques, ce qui requiert une mise à jour du script de démarrage du serveur CRASUBMT existant.
- Les modules de chargement SCLM Developer Toolkit ont été déplacés dans les nouvelles bibliothèques, ce qui requiert une mise à jour des définitions LINKLIST existantes.
- ELAXFTS0 est un nouvel exemple de procédure de génération depuis la version 7.1.1, ELAXFCP1 et ELAXFPP1 sont des nouveautés introduites à partir de la version 7.5.
- uchars.settings est un nouveau fichier de configuration pour les caractères non éditables.
- propertiescfg.properties est un nouveau fichier de configuration pour les groupes de propriétés par défaut.
- FEJJCNFG, CRASRV.properties et FMIEXT.properties possèdent de nouvelles directives optionnelles.
- rsed.envvars a été modifié dans la version 7.5 et doit être remplacé.
- Le fichier exemple ISPF.conf fourni avec la version 7.5 est similaire à celui utilisé par SCLM Developer Toolkit dans la version 7.1.
- Certaines personnalisations existantes du gestionnaire de déploiement d'application doivent être refaites.
- Le gestionnaire de déploiement d'application comporte de nouvelles fonctions qui doivent être personnalisées.
- Les paramètres de sécurité du serveur RSE ont été considérablement modifiés dans la version 7.5.
- Le profil de sécurité MVS.MCSOPER.JMON est nouveau pour le moniteur de travaux JES dans la version 7.5.
- Le script de démarrage CARMA a changé de nom et a été placé dans un nouvel emplacement, ce qui requiert une mise à jour du fichier de configuration CRASRV.properties existant.
- Le script de démarrage FMI a changé de nom et a été placé dans un nouvel emplacement, ce qui requiert une mise à jour du fichier de configuration FMIEXT.properties existant.
- Un nouveau module de chargement a été ajouté pour le support directionnel, ce qui requiert une mise à jour de la concaténation DFHRPL CICS existante si vous utilisez la bibliothèque FEK.SFEKLOAD.
- Les modifications apportées au paramètre MAXPROCUSER de SYS1.PARMLIB(BPXPRMxx) sont également documentées.

## Fichiers configurables

Le tableau 48 présente les fichiers qui sont personnalisés dans la version 7.5. Notez que les exemples de bibliothèque Developer for System z, FEK.SFEKSAMP, FEK.SFEKSAMPV et /usr/lpp/rdz/samples/, sont fournies avec davantage de membres personnalisables que ceux répertoriés ici, tels que du code source CARMA et des travaux pour leur compilation.

**Remarque :** L'exemple de travail FEKSETUP copie tous les membres répertoriés dans des fichiers et répertoires différents, par défaut FEK.#CUST.\* et /etc/rdz/\*.

Tableau 48. Personnalisations de la version 7.5

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL permettant de créer des fichiers et des répertoires et de les remplir avec des fichiers personnalisables	NOUVEAU, doit être personnalisé
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	Langage JCL pour le moniteur de travaux JES	STEPLIB remplacé par SFEKAUTH
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	Langage JCL pour le démon RSE	NOUVEAU, doit être personnalisé
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	Langage JCL de génération de projets distants, etc.	ELAXFTSO, ELAXFCP1 et ELAXFPP1 sont nouveaux
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL pour les définitions de sécurité	NOUVEAU, requis
FEJJCNGF	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Fichier de configuration du moniteur de travaux JES	<ul style="list-style-type: none"> <li>Certaines directives sont devenues facultatives</li> <li>De nouvelles directives facultatives ont été ajoutées</li> </ul>
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Langage JCL pour les soumissions TSO	Le nom du travail peut à présent être une variable
CRAISPRX	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Exemple de commande exec d'allocation de définition de données pour CARMA à l'aide de la passerelle client TSO/ISPF	NOUVEAU, la personnalisation est facultative
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Exemple de code source pour le remplacement de IRXJCL	NOUVEAU, la personnalisation est facultative
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de compilation de CRAXJCL	NOUVEAU, la personnalisation est facultative

Tableau 48. Personnalisations de la version 7.5 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
ADNSMSGH	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Exemple de code source pour le gestionnaire de message de pipeline	Les copies plus anciennes doivent être remplacées par la présente (les personnalisations doivent être effectuées à nouveau)
ADNPCCSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de définition du serveur CRD dans une région CICS primaire	Les copies plus anciennes doivent être remplacées par la présente (les personnalisations doivent être effectuées à nouveau)
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de mise à jour des valeurs par défaut de CRD	NOUVEAU, la personnalisation est facultative
ADNMFEST	FEK.SFEKSAMP [FEK.#CUST.JCL]	Langage JCL de création et de définition du référentiel de manifestes	NOUVEAU, la personnalisation est facultative
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	Variables d'environnement RSE	Les copies plus anciennes doivent être remplacées par la présente (les personnalisations doivent être effectuées à nouveau)
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la passerelle client TSO/ISPF	Identique au fichier ISPF.conf fourni avec SCLMDT dans la v7.1
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration du gestionnaire CARMA	<ul style="list-style-type: none"> <li>• L'emplacement et le nom du script de démarrage ont été modifiés</li> <li>• De nouvelles directives facultatives ont été ajoutées</li> </ul>
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRASTART	NOUVEAU, la personnalisation est facultative

Tableau 48. Personnalisations de la version 7.5 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
FMIEXT.properties	/usr/lpp/rdz/samples/ [etc/rdz/]	Fichier de configuration d'intégration de File Manager	<ul style="list-style-type: none"> <li>L'emplacement et le nom du script de démarrage ont été modifiés</li> <li>De nouvelles directives facultatives ont été ajoutées</li> </ul>
uchars.settings	/usr/lpp/rdz/samples/ [etc/rdz/]	Fichier de configuration des caractères non éditables	NOUVEAU, la personnalisation est facultative

## Migration de la version 7.0 à la version 7.1

### IBM Rational Developer for System z, FMID HHOP710

- L'emplacement d'installation SMP/E par défaut des composants MVS et z/OS UNIX n'a pas été modifié et correspond toujours à FEK.\* et à /usr/lpp/wd4z/\*.
- Ajout** : choix de configuration - commandes TSO/ISPF par l'intermédiaire d'une transaction APPC ou par l'intermédiaire de SCLM Developer Toolkit
- Modification** : la transaction APPC exploite une nouvelle fonction ISPF
- Ajout** : les membres personnalisables suivants sont nouveaux :
  - samplib ELAXFADT
  - samplib ADNCMSGH
  - /usr/lpp/wd4z/rse/lib/FMIEXT.properties
- Modification** : les membres suivants ont été déplacés :
  - SFEKDLL(FEJBDTRX) -> SFEKLOAD(FEJBDTRX)
- Modification** : les membres personnalisables suivants ont été modifiés :
  - samplib FEKFAPPCC
  - /usr/lpp/wd4z/rse/lib/rsed.envvars
  - /usr/lpp/wd4z/rse/lib/setup.env.zseries
  - /usr/lpp/wd4z/rse/lib/server.zseries

### IBM Common Access Repository Manager (CARMA), FMID HCMA710

- L'emplacement d'installation SMP/E par défaut des composants MVS n'a pas été modifié et correspond toujours à CRA.\*.
- Modification** : la journalisation est inscrite dans l'instruction de définition de données CARMALOG
- Modification** : la méthode d'accès VSAM au message CARMA (CRAMSG) et la méthode d'accès VSAM à la configuration (CRADEF) ont été mises à jour
- Ajout** : les membres personnalisables suivants sont nouveaux :
  - samplib CRA#ECOB
  - samplib CRA#EPDS

- samplib CRA#ERAM
- samplib CRA#ESLM
- **Changement de nom** : les membres personnalisables suivants ont été renommés :
  - samplib CRAREPR -> CRA\$VDEF
  - samplib CRAMREPR -> CRA\$VMSG
  - samplib CRASREPR -> CRA\$VSTR
  - samplib CRASALX -> CRA#ASLM
  - samplib CRACOBJ1 -> CRA#CCB1
  - samplib CRACOBJ2 -> CRA#CCB2
  - samplib CRACLICM -> CRA#CCLT
  - samplib CRARAMCS -> CRA#CPDS
  - samplib CRARAMCM -> CRA#CRAM
  - samplib CRATREPR -> CRA#VPDS
  - samplib CRALREPR -> CRA#VSLM
  - samplib CRACLIRN -> CRA#XCLT
- **Modification** : les membres personnalisables suivants ont été modifiés :
  - clist CRASUBMT

## Fichiers configurables

Le tableau 23 présente les fichiers qui sont personnalisés dans la version 7.1. Notez que les exemples de bibliothèque CARMA et Developer for System z CRA.SCRASAMP, FEK.SFEKSAMP et /usr/lpp/wd4z/rse/lib/ sont fournis avec davantage de membres personnalisables que ceux répertoriés ici (le code source CARMA et les travaux permettant de les compiler).

Tableau 49. Personnalisations de la version 7.1

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
ELAXF*	FEK.SFEKSAMP	Code JCL pour les générations de projets distants et autres travaux.	ELAXFADT est nouveau
CRA\$VMSG	CRA.SCRASAMP	Le langage JCL doit créer la méthode d'accès VSAM au message du gestionnaire CARMA	<ul style="list-style-type: none"> <li>• Renommé. S'appelait CRAMREPR</li> <li>• La méthode d'accès VSAM créée par ce travail est mise à jour</li> </ul>
CRA\$VDEF	CRA.SCRASAMP	Le langage JCL doit créer la méthode d'accès VSAM à la configuration du gestionnaire CARMA	<ul style="list-style-type: none"> <li>• Renommé. S'appelait CRAREPR</li> <li>• La méthode d'accès VSAM créée par ce travail est mise à jour</li> </ul>

Tableau 49. Personnalisations de la version 7.1 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Remarques sur la migration
CRA\$VSTR	CRA.SCRASAMP	Le langage JCL doit créer la méthode d'accès VSAM aux informations personnalisées du gestionnaire CARMA	Renommé. S'appelait CRASREPR
CRASUBMT	CRA.SCRASAMP	Démarrage par lots CARMA CLIST	Ajout de DD CARMALOG
CRA#VSLM	CRA.SCRASAMP	Langage JCL de création de la méthode d'accès VSAM au message de la RAM du SCLM	Renommé. S'appelait CRALREPR
CRA#ASLM	CRA.SCRASAMP	Langage JCL de création de fichiers de la RAM du SCLM	Renommé. S'appelait CRASALX
CRA#VPDS	CRA.SCRASAMP	Langage JCL de création de la méthode d'accès VSAM au message de la RAM du PDS	Renommé. S'appelait CRATREPR
CRA#CRAM	CRA.SCRASAMP	Langage JCL de compilation de la RAM du squelette	Renommé. S'appelait CRARAMCM
FEKAPPCC	FEK.SFEKSAMP	Le langage JCL doit créer une transaction APPC	Exploitation du support NEST ISPF
r sed.envvars	/usr/lpp/wd4z/rse/lib/ [/etc/wd4z/]	Variables d'environnement RSE	Les copies plus anciennes doivent être remplacées par la présente (les personnalisations doivent être effectuées à nouveau)
FMIEXT.properties	/usr/lpp/wd4z/rse/lib/ [/etc/wd4z/]	Fichier de configuration d'intégration de File Manager	NOUVEAU, doit être personnalisé



---

## Annexe A. Configuration de l'authentification SSL et X.509

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration du protocole SSL (Secure Socket Layer), ou au cours de la vérification et/ou de la modification d'une configuration existante. Elle contient un exemple de configuration pour prendre en charge les utilisateurs qui s'authentifient à l'aide d'un certificat X.509.

Une communication sécurisée vous assure que votre partenaire de communication est bien celui qu'il prétend être, et que la transmission des informations se fait d'une manière qui rend difficile toute interception et lecture des données par des tiers. Le protocole SSL fournit cette capacité dans un réseau TCP/IP. Il fonctionne par l'emploi de certificats numériques pour vous identifier et d'un protocole à clé publique pour chiffrer la communication. Voir le document *Security Server RACF Security Administrator's Guide* (SA22-7683) pour de plus amples informations sur les certificats numériques et le protocole de clé publique utilisés par le protocole SSL.

Les actions nécessaires pour la configuration des communications SSL pour Developer for System z varient largement d'un site à l'autre, selon les véritables besoins, la méthode de communication RSE employée, et ce qui est déjà disponible au niveau du site.

Dans la présente annexe, nous allons cloner les définitions RSE en cours de manière à avoir une seconde connexion au démon RSE qui utilisera le protocole SSL. Nous créerons également nos propres certificats de sécurité destinés à être utilisés par les différents composants de la connexion RSE.

- «Choix de l'emplacement de stockage des clés privées et des certificats», à la page 300
- «Création d'un fichier de clés avec RACF», à la page 301
- «Clonage de la configuration RSE existante», à la page 303
- «Mise à jour du fichier rsed.envvars pour assurer la coexistence», à la page 303
- «Mise à jour du fichier ssl.properties pour activer SSL», à la page 303
- «Activation de SSL en créant un démon RSE», à la page 304
- «Test de la connexion», à la page 305
- «(Facultatif) Ajout du support d'authentification du client via des certificats X.509», à la page 308
- «(Facultatif) Création d'une base de données de clés avec gskkyman», à la page 308
- «(Facultatif) Création d'un magasin de clés avec keytool», à la page 311

Une convention d'attribution de nom uniforme est utilisée dans cette annexe :

- Certificat : rdzrse
- Stockage de clés et de certificats : rdzssl.\*
- Mot de passe : rsessl
- ID utilisateur du démon : stcrse

Certaines des tâches décrites ci-après nécessitent des actions de votre part dans z/OS UNIX. Vous pouvez les effectuer en lançant la commande TSO **OMVS**. Utilisez la commande **exit** pour retourner à TSO.

## Choix de l'emplacement de stockage des clés privées et des certificats

Les certificats d'identité et les clés de chiffrement/déchiffrement utilisés par le protocole SSL sont stockés dans un fichier de clés. Différentes implémentations de ce fichier de clés existent, selon le type d'application.

Toutefois, toutes les implémentations suivent le même principe. Une commande génère une paire de clés (une clé publique et la clé privée associée). La commande intègre ensuite la clé publique à un certificat X.509 autosigné, qui est stocké comme une chaîne de certificats à un seul élément. Cette chaîne de certificats et la clé privée sont stockées en tant qu'entrée (identifiée par un alias) dans un fichier de clés.

Le démon RSE est une application du système SSL, qui utilise un fichier de base de données de clés. Cette base de données de clés peut être un fichier physique créé par gskkyman ou un fichier de clés géré par votre logiciel de sécurité conforme à SAF (RACF, par exemple). Le serveur RSE (démarré par le démon) est une application SSL Java qui utilise un magasin de clés créé par keytool ou un fichier de clés géré par votre logiciel de sécurité.

Tableau 50. Mécanismes de stockage des certificats SSL

Stockage des certificats	Créé et géré par	Démon RSE	Serveur RSE
Fichier de clés	Produit de sécurité compatible avec SAF	pris en charge	pris en charge
Base de données de clés	gskkyman de z/OS UNIX	pris en charge	/
Magasin de clés	Outil de clé de Java	/	pris en charge

Pour établir la connexion via SSL, le magasin de clés et le fichier de la base de données de clés sont nécessaires sous la forme d'un fichier z/OS UNIX ou d'un jeu de clés conforme à SAF :

- Magasin de clés (RACF ou outil de clé)
- base de données de clés (RACF ou gskkyman)

### Remarque :

- Il est conseillé d'utiliser des fichiers de clés conformes à SAF pour la gestion des certificats.
- Un certificat partagé peut être utilisé si le démon et le serveur RSE utilisent la même méthode de gestion des certificats.
- L'exécution du démon RSE doit être contrôlé par programme. L'utilisation de System SSL implique que SYS1.SIEALNKE soit contrôlé par programme via le logiciel de sécurité.
- Pour exécuter une application du système SSL (connexion par démon), SYS1.SIEALNKE doit être dans LINKLIST ou dans STEPLIB. Si vous préférez la méthode STEPLIB, ajoutez l'instruction suivante à la fin de rsed.envvars.

```
STEPLIB=$STEPLIB:SYS1.SIEALNKE
```

Gardez toutefois les remarques suivantes à l'esprit :

- L'utilisation de STEPLIB dans z/OS UNIX a un impact négatif sur les performances.

- Si une bibliothèque STEPLIB est autorisée par APF, il doit en être de même pour toutes les bibliothèques. Les bibliothèques perdent leur autorisation APF lorsqu'elles sont mélangées avec des bibliothèques non autorisées dans STEPLIB.
- Le système SSL utilise ICFS (Integrated Cryptographic Service Facility) si celui-ci est disponible. ICSF met à disposition un support de chiffrement matériel qui est utilisé à la place des algorithmes logiciels du système SSL. Pour plus d'informations, voir *System SSL Programming* (SC24-5901).

Pour obtenir des informations sur RACF et les certificats numériques, voir le document *Security Server RACF Security Administrator's Guide* (SA22-7683). La documentation relative à gskkyman est disponible dans le document *System SSL Programming* (SC24-5901) et la documentation de keytool se trouve à l'adresse <http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html>.

---

## Création d'un fichier de clés avec RACF

N'exécutez pas cette étape si vous utilisez gskkyman pour créer la base de données de clés du démon RSE et keytool pour créer le magasin de clés du serveur RSE.

La commande **RACDCERT** installe et maintient les clés privées et les certificats dans RACF. RACF prend en charge la gestion en groupe de multiples clés privées et certificats. Ces groupes sont des fichiers de clés.

Pour obtenir des informations détaillées sur la commande **RACDCERT**, voir le document *Security Server RACF Command Language Reference* (SA22-7687).

```

RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ACCESS(READ) ID(stcrse)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ACCESS(READ) ID(stcrse)
SETOPTS 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))

```

L'exemple ci-dessus commence par la création des profils nécessaires et par l'autorisation d'accès de l'ID utilisateur STCRSE aux jeux de clés et aux certificats détenus par cet ID utilisateur. L'ID utilisateur utilisé doit correspondre à celui employé pour exécuter le démon RSE SSL. L'étape suivante crée un certificat auto-signé avec l'intitulé rdzrse. Aucun mot de passe n'est nécessaire. Ce certificat est alors ajouté au fichier de clés nouvellement créé (rdzssl.racf). Exactement comme avec le certificat, aucun mot de passe n'est nécessaire pour le fichier de clés.

Le résultat peut être vérifié par l'intermédiaire de l'option list :

```

RACDCERT ID(stcrse) LIST
Digital certificate information for user STCRSE:

Label: rdzrse
Certificate ID: 2QjW10Xi0sXZ1aaEqZmihUBA
Status: TRUST
Start Date: 2007/05/24 00:00:00

```

```

End Date: 2017/05/21 23:59:59
Serial Number:
>00<
Issuer's Name:
>CN=rdz rse ssl.OU=rdz.O=IBM.L=Raleigh.SP=NC.C=US<
Subject's Name:
>CN=rdz rse ssl.OU=rdz.O=IBM.L=Raleigh.SP=NC.C=US<
Private Key Type: Non-ICSF
Private Key Size: 1024
Ring Associations:
  Ring Owner: STCRSE
  Ring:
    >rdzssl.racf<

```

## (Facultatif) Utilisation d'un certificat signé

Les certificats peuvent être autosignés ou signés par une autorité de certification (CA). Un certificat signé par une autorité de certification signifie que l'autorité de certification garantit que le propriétaire du certificat est bien la personne qu'il prétend être. La procédure de signature ajoute les données d'identification (certificat) de l'autorité de certification à votre certificat pour former une chaîne de certificats à plusieurs éléments.

Lorsque vous utilisez un certificat signé par une autorité de certification, vous pouvez éviter les questions de validation de la relation de confiance du client Developer for System z, si le client fait déjà confiance à l'autorité de certification.

Suivez les étapes suivantes pour créer et utiliser un certificat signé par une autorité de certification :

1. Créez un certificat autosigné.  
RACDCERT ID(stcrse) GENCERT WITHLABEL('rdzrse') . . .
2. Créez une demande de signature pour ce certificat.  
RACDCERT ID(stcrse) GENREQ (LABEL('rdzrse')) DSN(dsn)
3. Envoyez la demande de signature à l'autorité de certification de votre choix.
4. Vérifiez si les données d'identification (certificat) de l'autorité de certification sont déjà connues.  
RACDCERT CERTAUTH LIST
5. Marquez l'autorité de certification comme digne de confiance.  
RACDCERT CERTAUTH ALTER(LABEL('CA cert')) TRUST  
Vous pouvez ajouter le certificat de l'autorité de certification de la base de données.  
RACDCERT CERTAUTH ADD(dsn) WITHLABEL('CA cert') TRUST
6. Ajoutez le certificat signé à la base de données. Il doit remplacer le certificat autosigné.  
RACDCERT ID(stcrse) ADD(dsn) WITHLABEL('rdzrse') TRUST

**Remarque :** Ne supprimez PAS le certificat autosigné avant de le remplacer. Si tel est le cas, ne perdez pas la clé privée associée au certificat. Le certificat serait inutilisable.

7. Créez un jeu de clés.  
RACDCERT ID(stcrse) ADDRING(rdzssl.racf)
8. Ajoutez le certificat signé au jeu de clés.  
RACDCERT ID(stcrse) CONNECT(ID(stcrse) LABEL('rdzrse'))  
RING(rdzssl.racf)
9. Ajouter le certificat de l'autorité de certification au jeu de clés.

```
RACDCERT ID(stcrse) CONNECT(CERTAUTH LABEL('CA cert'))
RING(rdzssl.racf))
```

---

## Clonage de la configuration RSE existante

Dans cette étape, une nouvelle instance des fichiers de configuration RSE est créée, pour que la configuration SSL puisse être exécutée en parallèle avec celle(s) qui existe(nt) déjà. Dans les exemples de commandes ci-dessous, on suppose que les fichiers de configuration se trouvent dans /etc/rdz/ qui est l'emplacement utilisé par défaut pour l'étape «Configuration personnalisée», à la page 15.

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

Les commandes z/OS UNIX indiquées ci-dessus créent un sous-répertoire appelé `ssl` pour y placer les fichiers de configuration à modifier. Les autres fichiers de configuration, le répertoire d'installation et les composants MVS sont partagés car ils ne sont pas propres à SSL.

La réutilisation de la plupart des fichiers de configuration permet de se consacrer aux modifications nécessaires à la configuration SSL et d'éviter de réaliser de nouveau la configuration RSE (par exemple, vous pouvez éviter de définir un nouvel emplacement pour `ISPF.conf`.)

---

## Mise à jour du fichier `rsed.envvars` pour assurer la coexistence

Jusqu'à présent, les définitions sont une copie exacte de la configuration en cours, ce qui signifie que les journaux du nouveau démon RSE remplacent les fichiers journaux du serveur en cours. RSE doit également savoir où se trouvent les fichiers de configuration qui n'ont pas été copiés dans le répertoire `ssl`. Ces deux problèmes peuvent être corrigés en apportant des modifications mineures au fichier `rsed.envvars`.

```
$ oedit /etc/rdz/ssl/rsed.envvars
-> Suppression de la mise en commentaire et modification :
-Ddaemon.log=/var/rdz/logs/ssl
-> Ajout à la fin :
# -- NECESSAIRE POUR TROUVER LES FICHIERS DE CONFIGURATION RESTANTS
CFG_BASE=/etc/rdz
CLASSPATH=.:$CFG_BASE:$CLASSPATH
# --
```

Les modifications ci-dessus définissent un nouvel emplacement du journal (qui est créé par le démon RSE si l'emplacement du journal n'existe pas). Elles mettent également à jour la variable `CLASSPATH` afin que les processus RSE SSL recherchent les fichiers de configuration dans le répertoire en cours (`/etc/rdz/ssl`), puis dans le répertoire d'origine (`/etc/rdz`).

---

## Mise à jour du fichier `ssl.properties` pour activer SSL

Le fichier `ssl.properties` est mis à jour afin de demander à RSE de démarrer en utilisant une communication chiffrée SSL.

```
$ oedit /etc/rdz/ssl/ssl.properties
-> modification : enable_ssl=true
-> suppression de la mise en commentaire et modification :
daemon_keydb_file=rdzssl.racf
```

```

-> suppression de la mise en commentaire et modification :
daemon_key_label=rdzrse
-> suppression de la mise en commentaire et modification :
server_keystore_file=rdzssl.racf
-> suppression de la mise en commentaire et modification :
server_keystore_label=rdzrse
-> suppression de la mise en commentaire et modification :
server_keystore_type=JCERACFKS

```

Les modifications ci-dessus activent SSL et indiquent au démon et au serveur RSE que leur certificat (partagé) est stocké sous le nom rdzrse dans le fichier de clés rdzssl.racf. Le mot clé JCERACFKS indique au serveur RSE qu'un fichier de clés conforme à SAF est utilisé en tant que magasin de clés.

---

## Activation de SSL en créant un démon RSE

Comme indiqué précédemment, nous allons créer une deuxième connexion qui utilisera la couche SSL, ce qui implique la création d'un démon RSE. Le démon RSE peut être une tâche démarrée ou un travail utilisateur. Nous utiliserons la méthode du travail utilisateur pour la configuration initiale (test). Dans les instructions suivantes, il est supposé que l'exemple de langage JCL se trouve dans FEK.#CUST.PROCLIB(RSED), qui est l'emplacement utilisé par défaut pour l'étape «Configuration personnalisée», à la page 15:

1. Créez un membre FEK.#CUST.PROCLIB(RSEDSSL) et copiez dans ce dernier l'exemple de langage JCL FEK.#CUST.PROCLIB(RSED).
2. Personnalisez RSEDSSL en ajoutant une carte de travail en haut et une instruction exec en bas. Fournissez également un nouveau numéro de port (4047) et l'emplacement des fichiers de configuration SSL (/etc/rdz/ssl) (voir l'exemple de code suivant). Notez que nous appliquons l'ID utilisateur STCRSE car il a reçu des droits d'accès aux certificats et aux fichiers de clés à l'étape précédente.

```

//RSEDSSL JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),USER=STCRSE
//*
//* RSE DAEMON - SSL
//*
//RSED      PROC IVP='',                                * 'IVP' to do an IVP test
//          PORT=4047,
//          HOME='/usr/lpp/rdz',
//          CNFG='/etc/rdz/ssl'
//*
//RSE       EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
//          PARM='PGM &HOME./bin/rsed.sh &IVP &PORT &CNFG'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//PEND
//*
//RSED      EXEC RSED
//

```

*Figure 61. RSEDSSL - Travail de l'utilisateur du serveur RSE pour SSL*

**Remarque :** L'ID utilisateur affecté au travail RSEDSSL dispose des mêmes droits que le démon RSE d'origine. Le profil de FACILITY BPX.SERVER et le profil de PTKTDATA IRRPTAUTH.FEKAPPL.\* sont des éléments clés ici.

## Test de la connexion

La configuration de l'hôte SSL est maintenant terminée et le démon RSE pour la couche SSL peut être démarré par la soumission du travail FEK.#CUST.PROCLIB(RSEDSSL) précédemment créé.

La nouvelle configuration peut maintenant être testée via la connexion au client System z. Dans la mesure où nous avons créé une configuration pour l'utilisation avec SSL (par clonage d'une configuration existante), une nouvelle connexion doit être définie sur le client avec l'utilisation du port 4047 pour le démon RSE.

A la connexion, l'hôte et le client démarrent avec un protocole d'établissement de liaison pour configurer un chemin d'accès sécurisé. L'échange de certificats fait partie de ce protocole d'établissement de liaison. Si le client Developer for System z ne reconnaît pas le certificat de l'hôte ou l'autorité de certification qui l'a signé, il demande à l'utilisateur si ce certificat est digne de confiance.



Figure 62. Boîte de dialogue Importation du certificat hôte

En cliquant sur le bouton Finish, l'utilisateur peut accepter le certificat comme étant sécurisé, après quoi l'initialisation de la connexion se poursuit.

**Remarque :** Le démon et le serveur RSE peuvent utiliser deux emplacements de certificat différents, ce qui donne deux certificats différents et par conséquent, deux confirmations.

Une fois que le client connaît le certificat, cette boîte de dialogue n'est plus affichée. La liste des certificats de confiance peut être gérée en sélectionnant **Windows > Préférences... > Système distant > SSL**, ce qui provoque l'affichage de la boîte de dialogue suivante :



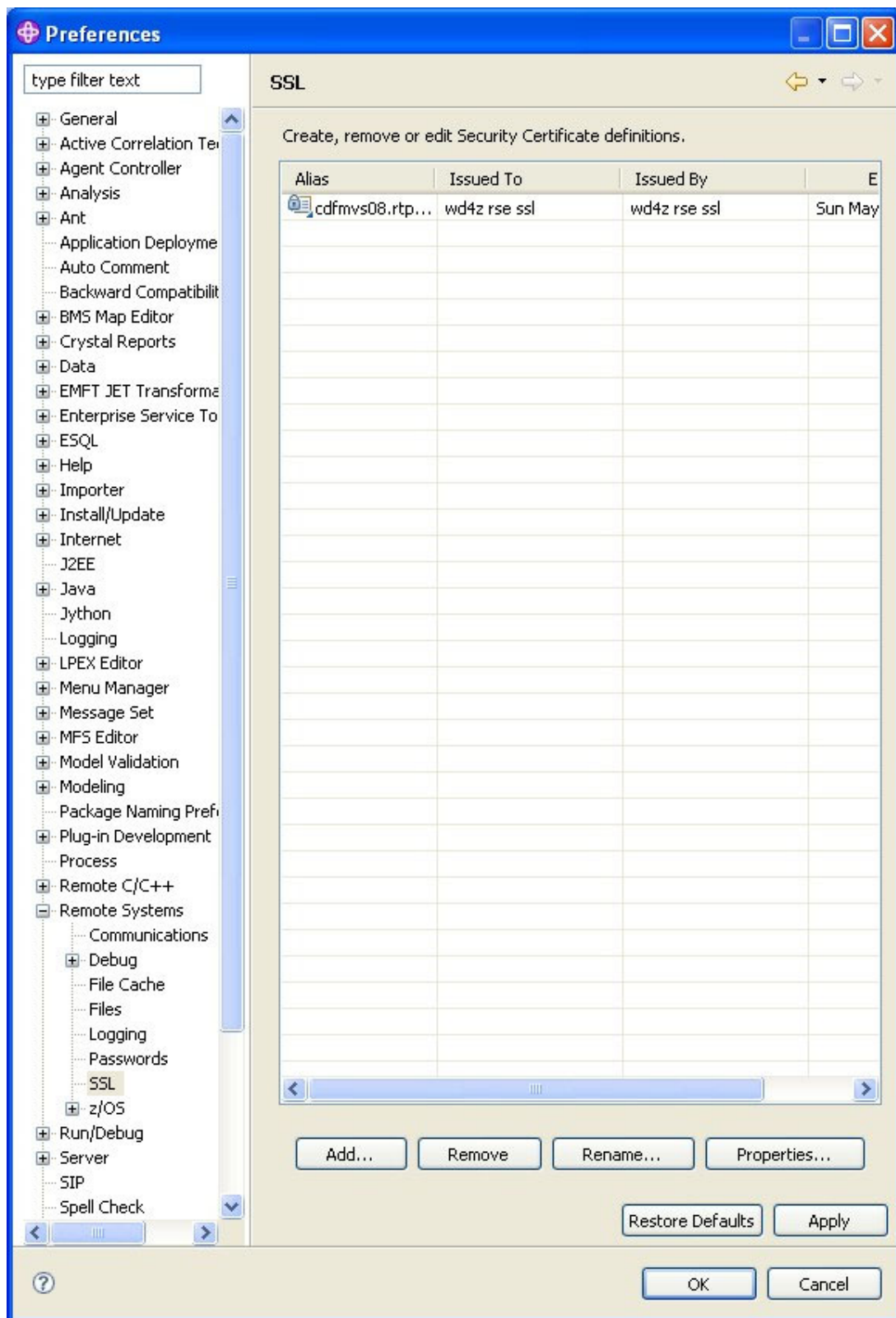


Figure 63. Boîte de dialogue Préférences - SSL

En cas d'échec des communications SSL le client renvoie un message d'erreur. Des informations complémentaires sont disponibles dans les différents fichiers journaux

du serveur et de l'utilisateur, comme indiqué à la section «Journalisation du démon RSE et du pool d'unités d'exécution», à la page 141 et «Journalisation pour l'utilisateur RSE», à la page 142.

---

## (Facultatif) Ajout du support d'authentification du client via des certificats X.509

Le démon RSE prend en charge les utilisateurs qui s'authentifient eux-mêmes à l'aide d'un certificat X.509. L'utilisation de communications chiffrées SSL est indispensable pour cette fonction car il s'agit d'une extension de l'authentification hôte avec un certificat utilisé dans SSL.

Il y a plusieurs méthodes pour effectuer l'authentification d'un utilisateur via un certificat, comme indiqué à la section «Authentification du client à l'aide de certificats X.509», à la page 173. Les étapes suivantes décrivent la configuration nécessaire pour que votre logiciel de sécurité authentifie le certificat à l'aide de l'extension de certificat HostIdMappings.

1. Remplacez le certificat qui identifie l'autorité de certification utilisée pour signer le certificat client par un certificat d'autorité de certification hautement sécurisée. Bien que l'état TRUST soit suffisant pour une validation de certificat, l'état HIGHTRUST est appliqué car il est utilisé pour l'authentification du certificat dans le cadre de la procédure de connexion.

```
RACDCERT CERTAUTH ALTER(LABEL('HighTrust CA')) HIGHTRUST
```

2. Ajoutez le certificat de l'autorité de certification au fichier de clés, `rdzssl.racf` afin qu'il soit disponible pour valider les certificats client.

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

La configuration du certificat de l'autorité de certification est terminée.

3. Définissez une ressource (format `IRR.HOST.hostname`) dans la classe `SERVAUTH` du nom d'hôte, `CDFMVS08.RALEIGH.IBM.COM`, défini dans l'extension `HostIdMappings` du certificat client.

```
RDEFINE SERVAUTH IRR.HOST.CDFMVS08.RALEIGH.IBM.COM UACC(NONE)
```

4. Accordez à l'ID utilisateur de la tâche démarrée, `STCRSE`, l'accès à cette ressource avec les droits `READ`.

```
PERMIT IRR.HOST.CDFMVS08.RALEIGH.IBM.COM CLASS(SERVAUTH) +  
ACCESS(READ) ID(stcrse)
```

5. Activez les modifications apportées à la classe `SERVAUTH`. Utilisez la première commande si la classe `SERVAUTH` n'est pas encore active. Utilisez la seconde pour régénérer une configuration active.

```
SETOPTS CLASSACT(SERVAUTH) RACLIST(SERVAUTH)  
ou  
SETOPTS RACLIST(SERVAUTH) REFRESH
```

La configuration du logiciel de sécurité pour l'extension `HostMappingscat` de l'autorité de certification est terminée.

6. Redémarrez la tâche démarrée RSE pour commencer à accepter des connexions client à l'aide de certificats X.509.

---

## (Facultatif) Création d'une base de données de clés avec gskkyman

N'exécutez pas cette étape si vous utilisez un fichier de clés conforme à SAF pour la base de données de clés du démon RSE.

gskkyman est un programme z/OS UNIX basé sur le shell, piloté par menus, qui crée, remplit et gère un fichier z/OS UNIX qui contient les clés privées, les demandes de certificats et les certificats. Ce fichier z/OS UNIX est une base de données de clés.

**Remarque :** Les instructions suivantes peuvent être nécessaires pour configurer l'environnement pour gskkyman. Pour en savoir plus, voir *System SSL Programming* (SC24-5901).

```
PATH=$PATH:/usr/lpp/gskssl/bin
export NLSPATH=/usr/lpp/gskssl/lib/nls/msg/En_US.IBM-1047/%N:$NLSPATH
export STEPLIB=$STEPLIB:SYS1.SIEALNKE
```

```
$ cd /etc/rdz/ssl
```

```
$ gskkyman      Menu de la base de données
```

```
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
```

L'exemple précédent commence par la création d'une base de données de clés appelée `rdzssl.kdb` avec le mot de passe `rsessl`. Lorsque la base de données existe, elle est enrichie en créant un certificat autosigné valide pendant 10 ans environ (sans compter les jours des années bissextiles). Le certificat est conservé sous le nom `rdzrse` et avec même le mot de passe (`rsessl`) que celui utilisé pour la base de données de clés (il s'agit d'un élément prérequis par RSE).

`gskkyman` attribue un masque de bits d'autorisation (très sûr, accès du seul propriétaire) de 600 à la base de données de clés. Mis à part le cas où le démon utilise le même ID utilisateur que le créateur de la base de données de clés, les autorisations doivent être définies d'une manière moins restrictive. Le masque 644 (le propriétaire a des droits d'accès en lecture/écriture, tout le monde a des droits d'accès en lecture) est utilisable pour la commande **`chmod`**.

Ce résultat peut être vérifié en sélectionnant l'option **Show certificate information** dans le sous-menu **Manage keys and certificates** :

```
$ 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

L'exemple de fichier `ssl.properties` ci-après indique que les directives `daemon_*` diffèrent de celles de l'exemple de fichier de clés SAF présenté plus haut.

```

$ oedit /etc/rdz/ssl/ssl.properties
-> modification : enable_ssl=true
-> suppression de la mise en commentaire et modification :
daemon_keydb_file=rdzssl.kdb
-> suppression de la mise en commentaire et modification :
daemon_keydb_password=rsessl
-> suppression de la mise en commentaire et modification :
daemon_key_label=rdzrse
-> suppression de la mise en commentaire et modification :
server_keystore_file=rdzssl.racf
-> suppression de la mise en commentaire et modification :
server_keystore_label=rdzrse
-> suppression de la mise en commentaire et modification :
server_keystore_type=JCERACFKS

```

Les modifications ci-dessus activent SSL et indiquent au démon RSE que le certificat est stocké sous le nom `rdzrse` dans la base de données de clés `rdzssl.kdb` avec le mot de passersessl. Le serveur RSE continue à utiliser un fichier de clés conforme à SAF.

---

## (Facultatif) Création d'un magasin de clés avec keytool

N'exécutez pas cette étape si vous utilisez un fichier de clés conforme à SAF pour le magasin de clés du serveur RSE.

"keytool -genkey" génère une paire de clés privées et un certificat autosigné associé, qui est stocké sous la forme d'une entrée (identifiée par un alias) dans un (nouveau) magasin de clés.

**Remarque :** Java doit être inclus dans vos répertoires de recherches de commandes. L'instruction suivante peut être nécessaire pour exécuter keytool, où `/usr/lpp/java/J5.0` est le répertoire d'installation de Java : `PATH=$PATH:/usr/lpp/java/J5.0/bin`

Toutes les informations peuvent être transmises comme paramètres, mais en raison des limitations de longueur de la ligne de commande, une certaine interactivité est nécessaire :

```
$ 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
```

Le certificat d'auto-signature créé précédemment est valide pendant environ 10 ans (sans compter les jours intercalaires des années bissextiles). Il est stocké dans /etc/rdz/ssl/rdzssl.jks avec l'alias rdzrse. Son mot de passe (rsessl) est identique au mot de passe du fichier de clés, ce qui est un élément prérequis pour RSE.

Le résultat peut être vérifié par l'intermédiaire de l'option -list :

```
$ 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
```

Dans l'exemple de fichier ssl.properties ci-après, les directives server\_\* diffèrent de celles de l'exemple de fichier de clés SAF présenté plus haut.

```
$ oedit /etc/rdz/ssl/ssl.properties
-> modification : enable_ssl=true
-> suppression de la mise en commentaire et modification :
daemon_keydb_file=rdzssl.racf
-> suppression de la mise en commentaire et modification :
daemon_key_label=rdzrse
-> suppression de la mise en commentaire et modification :
server_keystore_file=rdzssl.jks
-> suppression de la mise en commentaire et modification :
server_keystore_password=rsessl
-> suppression de la mise en commentaire et modification :
server_keystore_label=rdzrse
-> suppression de la mise en commentaire et modification
facultatifs : server_keystore_type=JKS
```

Les modifications ci-dessus activent SSL et indiquent au serveur RSE que le certificat est stocké sous le nom `rdzrse` dans le magasin de clés `rdzssl.jks` avec le mot de passe `rsessl`. Le démon RSE utilise toujours un fichier de clés conforme à SAF.





---

## Annexe B. Configuration de TCP/IP

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration du TCP/IP, ou au cours de la vérification ou de la modification d'une configuration existante.

Pour plus d'informations sur la configuration TCP/IP, voir *Communications Server: IP Configuration Guide* (SC31-8775) et *Communications Server: IP Configuration Reference* (SC31-8776).

---

### Dépendance au nom d'hôte

Lorsque vous utilisez APPC pour le service Commandes TSO, Developer for System z dépend de la validité du nom d'hôte du TCP/IP quand il est initialisé. Cela implique que les différents fichiers de configuration TCP/IP et du programme de résolution soient configurés correctement.

Vous pouvez tester votre configuration TCP/IP à l'aide du programme de vérification d'installation fekfivpt. La commande doit renvoyer un résultat comparable à celui de cet exemple (\$ correspond à l'invite z/OS UNIX) :

```
$ fekfivpt
```

```
Wed Jul  2 13:11:54 EDT 2008
uid=1(USERID) gid=0(GROUP)
using /etc/rdz/rsed.envvars
```

```
-----
TCP/IP resolver configuration (z/OS UNIX search order):
-----
```

```
Resolver Trace Initialization Complete -> 2008/07/02 13:11:54.745964
```

```
res_init Resolver values:
```

```
Global Tcp/Ip Dataset = None
Default Tcp/Ip Dataset = None
Local Tcp/Ip Dataset  = /etc/resolv.conf
Translation Table      = Default
UserId/JobName         = USERID
Caller API             = LE C Sockets
Caller Mode            = EBCDIC
(L) DataSetPrefix     = TCPIP
(L) HostName          = CDFMVS08
(L) TcpIpJobName       = TCPIP
(L) DomainOrigin       = RALEIGH.IBM.COM
(L) NameServer         = 9.42.206.2
                      9.42.206.3
(L) NsPortAddr        = 53          (L) ResolverTimeout      = 10
(L) ResolveVia        = UDP          (L) ResolverUdpRetries    = 1
(*) Options NDots      = 1
(*) SockNoTestStor     =
(*) AlwaysWto          = NO          (L) MessageCase         = MIXED
(*) LookUp             = DNS LOCAL
```

```
res_init Succeeded
```

```
res_init Started: 2008/07/02 13:11:54.755363
```

```
res_init Ended: 2008/07/02 13:11:54.755371
```

```
*****
```

```
MVS TCP/IP NETSTAT CS V1R9      TCPIP Name: TCPIP      13:11:54
```

```
Tcpip started at 01:28:36 on 06/23/2008 with IPv6 enabled
```

```
-----
```

```
host IP address:
-----
hostName=CDFMVS08
hostAddr=9.42.112.75
bindAddr=9.42.112.75
localAddr=9.42.112.75

Success, addresses match
```

---

## Présentation des programmes de résolution

Le programme de résolution joue le rôle des programmes comme un client qui accède aux serveurs de noms pour une résolution nom/adresse ou adresse/nom. Pour résoudre la requête du programme demandeur, le programme de résolution peut accéder aux serveurs de noms disponibles, utiliser des définitions locales (/etc/resolv.conf, /etc/hosts, /etc/ipnodes, HOSTS.SITEINFO, HOSTS.ADDRINFO, ou ETC.IPNODES, par exemple) ou utiliser une combinaison des deux.

Quand l'espace adresse du programme de résolution démarre, il lit un fichier de configuration du programme de résolution facultatif indiqué par la carte SETUP DD dans la procédure JCL du programme de résolution. Si les informations de configuration ne sont pas fournies, le programme de résolution utilise l'ordre de recherche MVS ou z/OS UNIX natif applicable, sans aucune information GLOBALTCPIPDATA, DEFAULTTCPIPDATA, GLOBALIPNODES, DEFAULTIPNODES ou COMMONSEARCH.

---

## Présentation des ordres de recherche d'informations de configuration

Il est important de connaître les fonctions de l'ordre de recherche des fichiers de configuration utilisés par TCP/IP, et de savoir quand vous pouvez remplacer l'ordre de recherche par défaut par des variables d'environnement, JCL ou autre variable fournie. Vous pouvez ainsi adapter votre fichier de données locales et les normes de dénomination d'un fichier d'un système hiérarchique ; il est également utile de savoir s'il s'agit du fichier de données de configuration ou du fichier du système hiérarchique qui est utilisé au moment d'identifier les incidents.

Autre point important, quand un ordre de recherche est appliqué à n'importe quel fichier de configuration, la recherche s'arrête au premier fichier trouvé. Par conséquent, vous risquez de trouver des résultats inattendus si vous enregistrez des informations de configuration dans un fichier qui n'est jamais trouvé, soit parce que d'autres fichiers le précèdent dans l'ordre de recherche ou parce que le fichier n'est pas inclus dans l'ordre de recherche choisi par l'application.

Quand vous recherchez des fichiers de configuration, vous pouvez indiquer clairement au protocole TCP/IP l'emplacement de la plupart des fichiers de configuration en utilisant des instructions de définition de données dans les procédures JCL ou en définissant des variables d'environnement. Sinon, vous pouvez laisser le protocole TCP/IP déterminer de manière dynamique l'emplacement des fichiers de configuration, en fonction des ordres de recherche documentés dans le guide *Communications Server: IP Configuration Guide* (SC31-8775).

Le composant de configuration de la pile TCP/IP utilise TCPIP.DATA au cours de l'initialisation de la pile TCP/IP afin d'en déterminer le paramètre HOSTNAME. Pour obtenir sa valeur, l'ordre de recherche de l'environnement z/OS UNIX est utilisé.

**Remarque :** L'utilitaire du programme de résolution de trace permet de déterminer les valeurs TCPIP.DATA qui sont utilisées par le programme de résolution et leur emplacement au moment de la lecture. Pour plus d'informations sur le démarrage dynamique de la trace, voir le document *Communications Server: IP Diagnosis Guide* (GC31-8782). Une fois que la fonction de trace est active, exécutez une commande TSO **NETSTAT HOME** et une commande de shell z/OS UNIX **netstat -h** pour afficher les valeurs. Une commande PING d'un nom d'hôte lancée via une commande TSO et l'interpréteur de commandes z/OS UNIX affiche également l'activité de tous les serveurs DNS qui peuvent être configurés.

---

## Ordres de recherche dans l'environnement z/OS UNIX

Le tableau ou fichier particulier recherché correspond à un fichier MVS ou à un fichier de système hiérarchique, en fonction des paramètres de configuration du programme de résolution et de la présence de ces fichiers sur le système.

### Fichiers de configuration du programme de résolution de base

Le fichier de configuration du programme de résolution de base contient des instructions TCPIP.DATA. Outre les directives du programme de résolution, il est référencé pour déterminer, entre autres, le préfixe du fichier (valeur de l'instruction DATASETPREFIX) à utiliser lors de la tentative d'accès à certains des fichiers de configuration spécifiés dans cette section.

L'ordre de recherche permettant d'accéder au fichier de configuration du programme de résolution de base est le suivant :

1. **GLOBALTCPIPDATA**

Si ce fichier est défini, la valeur de l'instruction de configuration GLOBALTCPIPDATA du programme de résolution est utilisée (voir aussi «Présentation des programmes de résolution», à la page 316). La recherche se poursuit pour trouver un autre fichier de configuration. La recherche s'arrête avec le fichier trouvé suivant.

2. La valeur de la variable d'environnement **RESOLVER\_CONFIG**.

La valeur utilisée est celle de la variable d'environnement. La recherche échoue si le fichier n'existe pas ou s'il se trouve dans un autre emplacement.

3. **/etc/resolv.conf**

4. Carte **//SYSTCPD DD**

Le fichier utilisé est celui qui est attribué au nom DD SYSTCPD. Dans l'environnement z/OS UNIX, un processus enfant n'a pas accès à SYSTCPD DD. En effet, l'attribution SYSTCPD n'est pas héritée du processus père sur le processus parallèle de traitement() ou les appels de fonction de commande exec.

5. **userid.TCPIP.DATA**

userid désigne l'ID utilisateur qui est associé à l'environnement de sécurité en cours (espace adresse, tâche ou unité d'exécution).

6. **jobname.TCPIP.DATA**

jobname correspond au nom indiqué dans l'instruction JCL JOB pour les travaux par lots ou le nom de la procédure pour une procédure démarrée.

7. **SYS1.TCPPARMS(TCPDATA)**

8. **DEFAULTTCPIPDATA**

Si ce fichier est défini, la valeur de l'instruction de configuration DEFAULTTCPIPDATA du programme de résolution est utilisée (voir aussi «Présentation des programmes de résolution», à la page 316).

## 9. TCPIP.TCPIP.DATA

### Tables de conversion

Les tables de conversion (EBCDIC en ASCII et ASCII en EBCDIC) permettent de déterminer les fichiers de conversion à utiliser. L'ordre de recherche permettant d'accéder à ce fichier de configuration est le suivant. L'ordre de recherche s'arrête au premier fichier trouvé :

1. La valeur de la variable d'environnement **X\_XLATE**. La valeur de la variable d'environnement correspond au nom de la table de conversion créée par la commande TSO CONVXLAT.
2. **userid.STANDARD.TCPXLBIN**  
userid désigne l'ID utilisateur qui est associé à l'environnement de sécurité en cours (espace adresse ou tâche/unité d'exécution).
3. **jobname.STANDARD.TCPXLBIN**  
jobname correspond au nom indiqué dans l'instruction JCL JOB pour les travaux par lots ou le nom de la procédure pour une procédure démarrée.
4. **hlq.STANDARD.TCPXLBIN**  
hlq représente la valeur de l'instruction DATASETPREFIX spécifiée dans le fichier de configuration du programme de résolution de base (le cas échéant) ; sinon, hlq correspond au protocole TCPIP par défaut.
5. Si aucune table n'est trouvée, le programme de résolution utilise une table codée en dur par défaut identique à la table figurant dans le membre de fichier SEZATCPX (STANDARD).

### Tables de système hôte local

Par défaut, le programme de résolution tente tout d'abord d'utiliser n'importe quel serveur de noms de domaine configuré pour les demandes de résolution. Si la demande de résolution n'est pas satisfaite, les tables de système hôte local sont utilisées. Le comportement du programme de résolution est contrôlé par les instructions TCPIP.DATA.

Les instructions du programme de résolution TCPIP.DATA définissent si et comment des serveurs de noms de domaine doivent être utilisés. L'instruction LOOKUP TCPIP.DATA permet également de contrôler l'utilisation des serveurs de noms de domaine et des tables de système hôte local. Pour de plus amples informations sur les instructions TCPIP.DATA, voir le document *Communications Server: IP Configuration Reference* (SC31-8776).

Le programme de résolution se sert de l'ordre de recherche unique Ipv4 pour trouver des informations de nom de site sans restrictions pour les appels API getnetbyname. L'ordre de recherche unique Ipv4 pour des informations de nom de site est le suivant. La recherche s'arrête au premier fichier trouvé :

1. La valeur de la variable d'environnement **X\_SITE**.  
La valeur de la variable d'environnement correspond au nom du fichier d'informations HOSTS.SITEINFO créé par la commande TSO MAKESITE.
2. La valeur de la variable d'environnement **X\_ADDR**.  
La valeur de la variable d'environnement correspond au nom du fichier d'informations HOSTS.ADDRINFO créé par la commande TSO MAKESITE.

3. **/etc/hosts**
4. **userid.HOSTS.SITEINFO**  
userid désigne l'ID utilisateur qui est associé à l'environnement de sécurité en cours (espace adresse ou tâche/unité d'exécution).
5. **jobname.HOSTS.SITEINFO**  
jobname correspond au nom indiqué dans l'instruction JCL JOB pour les travaux par lots ou le nom de la procédure pour une procédure démarrée.
6. **hlq.HOSTS.SITEINFO**  
hlq représente la valeur de l'instruction DATASETPREFIX spécifiée dans le fichier de configuration du programme de résolution de base (le cas échéant) ; sinon, hlq correspond au protocole TCPIP par défaut.

---

## Application de ces informations de configuration à Developer for System z

Comme indiqué précédemment, Developer for System z dépend de la validité du nom d'hôte TCP/IP lors de son initialisation, lors de l'utilisation d'APPC. Cela implique que les différents fichiers de configuration TCP/IP et du programme de résolution soient configurés correctement.

L'exemple ci-dessous présente certaines tâches de la configuration TCP/IP et du programme de résolution. Il ne couvre pas l'ensemble de la configuration TCP/IP ou du programme de résolution, il souligne uniquement certains aspects clés qui peuvent s'appliquer à votre site :

1. Dans le JCL ci-dessous, nous voyons que le protocole TCP/IP va se servir de SYS1.TCPPARMS(TCPDATA) pour déterminer le nom d'hôte de la pile.
 

```
//TCPIP    PROC  PARM='CTRACE(CTIEZB00)',PROF=TCPPROF,DATA=TCPDATA
//*
//* TCP/IP NETWORK
//*
//TCPIP    EXEC  PGM=EZBTCP,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) indique qu le nom du système doit être le nom d'hôte et qu'aucun serveur de noms de domaine (DNS) n'est utilisé. Tous les noms sont résolus en consultant le tableau du site.
 

```
; HOSTNAME indique le nom d'hôte TCP de ce système. S'il n'est pas
; spécifié, le paramètre HOSTNAME par défaut sera le nom de noeud spécifié
; dans le membre IEFSSNx PARMLIB.
;
; HOSTNAME
;
; DOMAINORIGIN indique l'origine du domaine qui sera ajoutée
; aux noms d'hôte transmis au programme de résolution. Si un nom d'hôte contient
; des points, le paramètre DOMAINORIGIN ne sera pas ajouté au
; nom d'hôte.
;
DOMAINORIGIN  RALEIGH.IBM.COM
;
; NSINTERADDR indique l'adresse IP du serveur de noms.
; LOOPBACK (14.0.0.0) indique votre serveur de noms local. Si vous n'utilisez
; pas de serveur de noms, ne codez pas d'instruction NSINTERADDR.
```

```
; (Mettez en commentaire la ligne NSINTERADDR ci-dessous). Cette action va
; résoudre tous les noms via la consultation du tableau de site.
;
; NSINTERADDR 14.0.0.0
;
; TRACE RESOLVER va créer une trace complète des requêtes à destination
; et des réponses provenant du serveur de noms ou des tableaux de site
; à écrire dans la console de l'utilisateur. Cette commande s'applique à
; des fins de débogage uniquement.
;
; TRACE RESOLVER
```

3. Dans le langage JCL du programme de résolution, l'instruction SETUP DD n'est pas utilisée. Comme mentionné dans «Présentation des programmes de résolution», à la page 316, cela signifie que GLOBALTCPIPDATA et les autres variables ne seront pas utilisées.

```
//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. Supposons que la variable d'environnement RESOLVER\_CONFIG n'est pas définie, le tableau 51, à la page 321 nous montre que le programme de résolution va tenter d'utiliser /etc/resolv.conf comme fichier de configuration de base.

```
TCPIPJOBNAME TCPIP
DomainOrigin RALEIGH.IBM.COM
HostName CDFMVS08
```

Comme mentionné dans «Ordres de recherche dans l'environnement z/OS UNIX», à la page 317, le fichier de configuration de base contient des instructions TCPIP.DATA. Si le nom du système est CDFMVS08 (TCPDATA indique que le nom du système est utilisé comme nom d'hôte), nous pouvons voir que /etc/resolv.conf est en synchronisation avec SYS1.TCPPARMS(TCPDATA). Aucune définition de système de nom de domaine n'existe, par conséquent la consultation du tableau de site sera utilisée.

5. Le tableau 51, à la page 321 nous indique également que nous n'avons rien à faire pour utiliser la table de conversion ASCII-EBCDIC par défaut.
6. En supposant que la commande TSO **MAKESITE** n'est pas utilisée (possibilité de créer les variables X\_SITE et X\_ADDR), /etc/hosts sera le tableau de site désigné pour la consultation du nom.

```
# 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
```

Le contenu minimal de ce fichier comprend des informations sur le système actuel. Dans l'exemple ci-dessus, nous définissons cdfmvs08 et cdfmvs08.raleigh.ibm.com comme nom valide pour l'adresse IP du système z/OS.

Si nous utilisons un serveur de noms de domaine (DNS), le DNS contiendrait des informations /etc/hosts, et /etc/resolv.conf et SYS1.TCPPARMS(TCPDATA) auraient des instructions identifiant le DNS sur notre système.

Pour éviter toute confusion, il est recommandé de conserver les fichiers de configuration TCP/IP et du programme de résolution en synchronisation les uns avec les autres.

Tableau 51. Définitions locales disponibles pour le programme de résolution

Description de type de fichier	API concernée(s)	Fichiers candidats
Fichiers de configuration du programme de résolution de base	Toutes les API	<ol style="list-style-type: none"> <li>1. GLOBALTCPIPDATA</li> <li>2. Variable d'environnement RESOLVER_CONFIG</li> <li>3. /etc/resolv.conf</li> <li>4. SYSTCPD DD-name</li> <li>5. userid.TCPIP.DATA</li> <li>6. jobname.TCPIP.DATA</li> <li>7. SYS1.TCPPARMS(TCPDATA)</li> <li>8. DEFAULTTCPIPDATA</li> <li>9. TCPIP.TCPIP.DATA</li> </ol>
Tables de conversion	Toutes les API	<ol style="list-style-type: none"> <li>1. Variable d'environnement X_XLATE</li> <li>2. userid.STANDARD.TCPXLBIN</li> <li>3. jobname.STANDARD.TCPXLBIN</li> <li>4. hlq.STANDARD.TCPXLBIN</li> <li>5. Table de conversion fournie par le programme de résolution, membre STANDARD dans SEZATCPX</li> </ol>
Tables de système hôte local	endhostent endnetent getaddrinfo gethostbyaddr gethostbyname gethostent GetHostNumber GetHostResol GetHostString getnameinfo getnetbyaddr getnetbyname getnetent IsLocalHost Resolve sethostent setnetent	<p>IPv4</p> <ol style="list-style-type: none"> <li>1. Variable d'environnement X_SITE</li> <li>2. Variable d'environnement X_ADDR</li> <li>3. /etc/hosts</li> <li>4. userid.HOSTS.xxxxINFO</li> <li>5. jobname.HOSTS.xxxxINFO</li> <li>6. hlq.HOSTS.xxxxINFO</li> </ol> <p>IPv6</p> <ol style="list-style-type: none"> <li>1. GLOBALIPNODES</li> <li>2. Variable d'environnement RESOLVER_IPNODES</li> <li>3. userid.ETC.IPNODES</li> <li>4. jobname.ETC.IPNODES</li> <li>5. hlq.ETC.IPNODES</li> <li>6. DEFAULTIPNODES</li> <li>7. /etc/ipnodes</li> </ol>

**Remarque :** Le tableau 51 est une copie partielle d'un tableau situé dans le document *Communications Server: IP Configuration Guide* (SC31-8775). Consultez ce manuel pour voir le tableau complet.

## Résolution erronée de l'adresse hôte

Des incidents dans lesquels le programme de résolution TCP/IP ne parvient pas à résoudre correctement l'adresse hôte, peuvent être engendrés par un fichier de configuration manquant ou incomplet du programme de résolution. Le message `lock.log` suivant est une indication évidente de l'apparition de cet incident :

```
clientip(0.0.0.0) <> callerip(<adresse IP de l'hôte>)
```



Pour procéder à une vérification, exécutez le programme de vérification d'installation TCP/IP fekfivpt, comme spécifié au Chapitre 7, «Vérification de l'installation», à la page 111. La section relative à la configuration du programme de résolution de la sortie ressemblera à l'exemple suivant :

```
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
```

Vérifiez que les définitions du fichier référencées par "Fichier Tcp/Ip local" sont correctes.

Cette zone est vide si vous n'utilisez pas un nom par défaut pour le fichier du programme de résolution IP (à l'aide de l'ordre de recherche z/OS UNIX). Si tel est le cas, ajoutez l'instruction suivante dans rsed.envvars, où <fichier du programme de résolution> ou <données du programme de résolution représente le nom de votre fichier de programme de résolution IP :

```
RESOLVER_CONFIG=<fichier du programme de résolution>
```

ou

```
RESOLVER_CONFIG='<fichier du programme de résolution>'
```



---

## Annexe C. Configuration d'INETD

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration d'INETD ou au cours de la vérification ou de la modification d'une configuration existante. INETD est utilisé par Developer for System z pour la fonction REXEC/SSH.

Le démon INETD fournit une gestion des services d'un réseau IP. Il réduit la charge du système en appelant d'autres démons uniquement lorsqu'ils sont nécessaires et en fournissant en interne des services Internet simples (écho, par exemple). INETD consulte le fichier de configuration `inetd.conf` afin de déterminer quels services supplémentaires proposer. `ETC.SERVICES` sert à lier les services aux ports.

---

### `inetd.conf`

Les services qui dépendent de INETD sont définis dans `inetd.conf`, qui est lu par INETD au moment du démarrage. Le nom et emplacement par défaut de `inetd.conf` est `/etc/inetd.conf`. Un exemple de fichier `inetd.conf` se trouve dans `/samples/inetd.conf`.

La syntaxe suivante s'applique aux entrées `inetd.conf`:

- Les commentaires commencent avec un signe dièse (#) ou un point-virgule (;) et se poursuivent jusqu'à la fin de la ligne
- Les entrées sont sensibles à la casse
- Les entrées sont sensibles au champ mais pas à la colonne
- Les champs sont séparés par un espace ou un caractère de tabulation
- Les entrées peuvent s'étendre sur plusieurs lignes, selon cette syntaxe supplémentaire :
  - La séparation doit se situer entre deux mots distincts (séparés par un espace ou un caractère de tabulation)
  - La ligne de continuation doit démarrer par un espace ou un caractère de tabulation
  - Aucun commentaire ne peut être incorporé dans la suite

Chaque entrée se compose de 7 champs à position fixe, selon le format :

```
service_name socket_type protocol wait_flag userid server_program
                server_program_arguments
```

#### **[ip\_address:]service\_name**

`ip_address` est une adresse IP locale, suivie par un double point (:). Si elle est indiquée, l'adresse est utilisée à la place d'`INADDR_ANY` ou de la valeur par défaut en cours. Pour demander spécifiquement `INADDR_ANY`, utilisez `"*:"`. Si `ip_address` (ou un double point) est indiquée sans autre entrée sur la ligne, elle devient la nouvelle valeur par défaut pour les lignes suivantes jusqu'à ce qu'une nouvelle valeur par défaut soit indiquée. `service_name` est un nom de service identifié, ou défini par l'utilisateur. Le nom indiqué doit correspondre l'un des noms de serveur définis dans `ETC.SERVICES`.

**socket\_type**

stream ou dgram, pour indiquer que respectivement un flux ou un datagramme est utilisé pour le service.

**protocol[,sndbuf=n[,rcvbuf=n]]**

protocol peut être tcp[4|6] ou udp[4|6], et est utilisé pour qualifier plus avant le nom de service. Le nom de service et le protocole doivent tous les deux correspondre à une entrée dans ETC.SERVICES, mis à part que le "4" ou "6" ne devrait pas être inclus dans l'entrée ETC.SERVICES.

sndbuf et rcvbuf indiquent la taille des mémoires tampon d'envoi et de réception. La taille, représentée par n, peut être exprimée en octets, ou un "k" ou un "m" peuvent être nécessaires pour indiquer respectivement des kilooctets ou des mégaoctets. sndbuf et rcvbuf peuvent être utilisés dans n'importe quel ordre.

**wait\_flag[.max]**

wait ou nowait. wait indique que le démon dispose d'une seule unité d'exécution et qu'une nouvelle requête ne sera pas traitée tant que la précédente n'aura pas été terminée. Si nowait est indiqué, INETD émet une acceptation lorsqu'une demande de connexion est reçue sur une prise pour flot de données. Si wait est indiqué, il est de la responsabilité du serveur d'émettre une acceptation, s'il s'agit d'une prise pour flot de données.

max est le nombre maximum d'utilisateurs autorisés à effectuer une demande de service dans un intervalle de 60 secondes. La valeur par défaut est 40. En cas de dépassement, le port du service est arrêté.

**userid[group]**

userid est l'ID utilisateur que le démon en processus parallèle de traitement doit utiliser pour son exécution. Cet ID utilisateur peut être différent de celui d'INETD. Les autorisations affectées à cet ID utilisateur dépendent des besoins du service. L'ID utilisateur d'INETD nécessite une autorisation BPX.DAEMON pour faire passer le traitement par duplication à cet ID utilisateur.

La valeur facultative group, qui est séparée de l'ID utilisateur par un point (.), permet au serveur de s'exécuter avec un ID groupe différent de celui par défaut pour cet ID utilisateur.

**server\_program**

server\_program est le chemin d'accès complet du service. Par exemple, /usr/sbin/rlogind est le chemin d'accès complet de la commande rlogind.

**server\_program\_arguments**

20 arguments au maximum. Le premier argument est le nom du serveur.

---

## ETC.SERVICES

INETD utilise ETC.SERVICES pour mapper les numéros de port et les protocoles vers les services qu'ils doivent prendre en charge. Il peut s'agir d'un fichier MVS ou z/OS UNIX. Un exemple est fourni dans SEZAINST(SERVICES), qui est également disponible comme /usr/lpp/tcpip/samples/services. L'ordre de recherche pour ETC.SERVICES dépend de la méthode de démarrage d'INETD : z/OS UNIX ou MVS natif.

La syntaxe suivante s'applique à la spécification d'informations de maintenance :

- Un fichier MVS ETC.SERVICES doit être fixe ou à bloc fixe avec un LRECL entre 56 et 256
- Un fichier HFS ETC.SERVICES peut avoir une longueur de ligne maximale de 256
- Les éléments sur une ligne sont séparés par des espaces ou des caractères de tabulation
- Chaque service figure en liste sur une seule ligne
- Un nom de service doit démarrer en première position sur une ligne
- La longueur maximale pour un nom de service et pour un nom d'alias est de 32 caractères
- Un maximum de 35 alias sera reconnu
- Les noms de service et d'alias sont sensibles à la casse
- Les commentaires commencent avec un signe dièse (#) ou un point-virgule (;) et se poursuivent jusqu'à la fin de la ligne

Chaque entrée se compose de quatre champs à position fixe, selon le format :

service\_name    port\_number/protocol    aliases

**service\_name**

Indique un nom de service identifié, ou défini par l'utilisateur.

**port\_number**

Indique le numéro de port de socket utilisé pour le service

**protocol**

Indique le protocole de transport utilisé pour le service. Les valeurs valides sont tcp et udp

**aliases**

Indique une liste de noms de service non officiels

## Ordre de recherche utilisé dans l'environnement z/OS UNIX

L'ordre de recherche utilisé pour accéder à ETC.SERVICES dans z/OS UNIX est indiqué ci-après. La recherche s'arrête au premier fichier trouvé :

1. **/etc/services**
2. **userid.ETC.SERVICES**  
userid est l'ID utilisateur qui est utilisé pour démarrer INETD.
3. **hlq.ETC.SERVICES**  
hlq représente la valeur de l'instruction DATASETPREFIX spécifiée dans le fichier de configuration du programme de résolution de base (le cas échéant) ; sinon, hlq correspond au protocole TCPIP par défaut.

## Ordre de recherche dans l'environnement MVS natif

L'ordre de recherche permettant d'accéder à ETC.SERVICES dans le système MVS natif est le suivant. La recherche s'arrête au premier fichier trouvé :

1. **//SERVICES DD card**  
Le fichier alloué à l'instruction de définition de données SERVICES est utilisé.
2. **userid.ETC.SERVICES**  
userid est l'ID utilisateur qui est utilisé pour démarrer INETD.
3. **jobname.ETC.SERVICES**  
jobname correspond au nom indiqué dans l'instruction JCL JOB pour les travaux par lots ou au nom de la procédure pour une procédure démarrée

#### 4. hlq.ETC.SERVICES

hlq représente la valeur de l'instruction DATASETPREFIX spécifiée dans le fichier de configuration du programme de résolution de base (le cas échéant) ; sinon, hlq correspond au protocole TCPIP par défaut.

**Remarque :** Le démarrage d'INETD par l'intermédiaire de BPXPATCH ne résulte pas dans l'utilisation de l'ordre de recherche du système MVS natif, dans la mesure où BPXBATCH exécute la commande de démarrage dans l'environnement z/OS UNIX. L'ordre de recherche du système MVS natif est utilisé uniquement lors du démarrage d'un module de chargement MVS (SEZALOAD(FTP), par exemple).

---

## Définitions de ports PROFILE.TCPIP

Ne confondez pas les définitions PORT (ou PORTRANGE) de PROFILE.TCPIP avec les ports définis dans ETC.SERVICES dans la mesure où ces définitions ont des buts différents. Les ports définis dans PROFILE.TCPIP sont utilisés par le protocole TCPIP pour vérifier si le port est réservé pour un service donné. ETC.SERVICES est utilisé par INETD pour mapper un port à un service.

Lorsqu'INETD reçoit une demande sur un port contrôlé, il génère un processus enfant parallèle (avec le service demandé), qui est appelé inetdx, où inetd est le nom de travail pour INETD (selon la méthode de démarrage) et où x est un chiffre.

Cela complique la réservation de ports. Ainsi, dans le cas où un port contrôlé INETD est réservé dans PROFILE.TCPIP, il est recommandé d'utiliser le nom de la procédure JCL démarrée pour l'espace adresse de noyau z/OS UNIX afin de permettre à presque tous les processus de se lier au port. Ce nom est généralement OMVS, sauf si un nom différent est explicitement indiqué dans le paramètre STARTUP\_PROC du membre parmlib BPXPRMxx.

La liste suivante explique comment déterminer le nom de travail, étant donné l'environnement dans lequel l'application est exécutée.

- Les applications exécutées à partir d'un lot utilisent le nom du travail par lots.
- Les applications lancées à partir de la console opérateur MVS utilisent le nom de la procédure démarrée (STC) comme nom de travail.
- Les applications exécutées à partir d'un ID utilisateur TSO utilisent cet ID utilisateur comme nom de travail.
- Les applications exécutées à partir de l'interpréteur de commandes de z/OS présentent normalement un nom de travail qui est l'ID utilisateur connecté plus un suffixe d'un caractère.
- Les utilisateurs autorisés peuvent exécuter des applications à partir de l'interpréteur de commandes de z/OS et utilisent la variable d'environnement \_BPX\_JOBNAME pour définir le nom de travail. Dans ce cas, la valeur indiquée pour la variable d'environnement est le nom de travail.
- Le nom de la procédure de langage JCL démarrée pour l'espace adresse de noyau des services système UNIX peut être utilisé pour permettre à n'importe quel demandeur du socket API bind() (sauf pour les utilisateurs de l'API en langage Pascal) de se lier à ce port. Ce nom est généralement OMVS, sauf si un nom différent est explicitement indiqué dans le paramètre STARTUP\_PROC du membre parmlib BPXPRMxx.
- Les applications z/OS UNIX démarrées par INETD utilisent le nom de travail du serveur INETD.

**Remarque :** Bien qu'il ne soit pas conseillé de procéder de la sorte, les ports définis dans ETC.SERVICES peuvent être différents du numéro de port réservé pour le service dans PROFILE.TCPIP.

---

## /etc/inetd.pid

Le processus INETD crée un fichier temporaire, /etc/inetd.pid, qui contient l'ID processus (PID) du démon INETD donc l'exécution est en cours. Cette valeur de PID est utilisée pour identifier les enregistrements du journal système qui ont le processus INETD pour origine, et pour fournir la valeur de PID aux commandes qui en nécessitent une, comme arrêter. Elle est également utilisée comme mécanisme de verrouillage afin de ne pas avoir plusieurs INETD actifs.

---

## Démarrage

La mise en oeuvre z/OS UNIX d'INETD est assurée par défaut dans /usr/sbin/inetd et prend en charge deux paramètres de démarrage facultatifs qui ne dépendent pas de la position :

/usr/sbin/inetd [-d] [inetd.conf]

**-d** Option de débogage. La sortie de débogage est inscrite dans stderr, qui peut être routé vers un fichier par le démon syslogd. Voir le document *Communications Server IP Configuration Guide* (SC31-8775) pour de plus amples informations sur syslogd. Notez que lorsqu'il est démarré de cette manière, INETD ne génère pas de processus enfant parallèle pour démarrer un service.

### inetd.conf

Fichier de configuration. La valeur par défaut est /etc/inetd.conf

Il est recommandé de démarrer INETD au moment de l'IPL. La façon la plus courante de procéder est de le démarrer à partir de /etc/rc ou de /etc/inittab (à partir de z/OS 1.8 seulement). Il peut également être démarré à partir d'un travail ou d'une tâche démarrée en utilisant BPXBATCH ou à partir d'une session de l'interpréteur de commandes d'un utilisateur ayant les droits d'accès correspondants.

## /etc/rc

Lorsqu'il est démarré à partir du script de shell d'initialisation de z/OS UNIX, /etc/rc, INETD utilise l'ordre de recherche de z/OS UNIX pour trouver ETC.SERVICES. Un exemple de fichier /etc/rc est fourni comme /samples/rc. Les exemples de commande suivants peuvent être utilisés pour démarrer INETD :

```
# Start INETD
_BPX_JOBNAME='INETD' /usr/sbin/inetd /etc/inetd.conf &
sleep 5
```

## /etc/inittab

z/OS 1.8 et au dessus met à disposition une méthode alternative, /etc/inittab, pour l'émission des commandes lors de l'initialisation de z/OS UNIX. /etc/inittab permet de définir le paramètre respawn, qui redémarre automatiquement le processus lorsqu'il se termine (un WTOR est envoyé à l'opérateur pour un deuxième redémarrage dans les 15 minutes). Lorsqu'il est démarré à partir de /etc/inittab, INETD utilise l'ordre de recherche de z/OS UNIX pour trouver ETC.SERVICES. Un exemple de fichier /etc/inittab est fourni comme /samples/inittab. L'exemple de commande suivant peut être utilisé pour démarrer INETD :

```
# Start INETD
inetd::respfrk:/usr/sbin/inetd /etc/inetd.conf
```

**Remarque :** Notez que le paramètre respfrk utilisé dans l'exemple transférera l'attribut respawn à tous les traitements par duplication, y compris RSE. Lorsque le client ferme la connexion, respawn la redémarre. Le serveur RSE s'arrêtera à nouveau plus tard, en raison du dépassement du délai d'attente. Voir le document *UNIX System Services Planning* (GA22-7800), pour de plus amples informations sur inittab.

## BPXBATCH

La méthode de démarrage BPXBATCH fonctionne pour les tâches démarrées et les travaux d'utilisateur. Veuillez noter qu'INETD est un processus d'arrière-plan, de sorte que l'étape BPXBATCH du démarrage d'INETD se termine en quelques secondes après le démarrage. Lorsqu'il est démarré par BPXBATCH, INETD utilise l'ordre de recherche de z/OS UNIX pour trouver ETC.SERVICES. Le langage JCL répertorié dans l'exemple de code suivant est un exemple de procédure de démarrage d'INETD (l'étape KILL retire un processus INETD actif, s'il y en a un) :

```
//INETD    PROC PRM=
//*
//KILL      EXEC PGM=BPXBATCH,REGION=0M,
//          PARM='SH ps -e | grep inetd | cut -c 1-10 | xargs -n 1 kill'
//*
//INETD     EXEC PGM=BPXBATCH,REGION=0M,
//          PARM='PGM /usr/sbin/inetd &PRM'
//STDERR DD SYSOUT=*
//* STDIN, STDOUT and STDENV are defaulted to /dev/null
//*
```

Figure 64. Langage JCL de démarrage d'INETD

**Remarque :**

- STDIN, STDOUT et STDERR doivent être des fichiers z/OS UNIX lorsqu'ils sont attribués. STDENV peut être soit un fichier MVS soit un fichier z/OS UNIX. A partir de z/OS 1.7, SYSOUT peut être affecté à STDOUT et à STDERR. Voir le document *UNIX System Services Command Reference* (SA22-7802) pour en apprendre davantage sur BPXBATCH.
- inetd.conf peut être un fichier ou un membre MVS quand INETD est démarré par BPXBATCH. Pour ce cela, codez l'instruction PARM comme dans l'exemple suivant (utilisez des apostrophes simples (') uniquement)) :  

```
// PARM='PGM /usr/sbin/inetd //'SYS1.TCPPARMS(INETCONF)'' &PRM'
```

## Session de Shell

Lorsqu'il est démarré à partir d'une session de shell, INETD utilise l'ordre de recherche de z/OS UNIX pour trouver ETC.SERVICES. Les exemples de commande suivants peuvent être utilisés (par un utilisateur disposant des droits d'accès nécessaires) pour arrêter et démarrer INETD (# désigne l'invite z/OS UNIX) :

```
# ps -e | grep inetd
  7 ?          0:00 /usr/sbin/inetd
# kill 7
# _BPX_JOBNAME='INETD' /usr/sbin/inetd &
```

**Remarque :** Cette méthode n'est pas recommandée pour le démarrage initial, /etc/rc ou /etc/inittab sont plus appropriées dans la mesure où elles sont exécutées lors de l'initialisation de z/OS UNIX.

---

## Sécurité

INETD est un processus z/OS UNIX et nécessite donc des définitions OMVS valides dans le logiciel de sécurité pour l'ID utilisateur associé à INETD. UID, HOME et PROGRAM doivent être définis pour l'ID utilisateur, ainsi que le GID pour le groupe par défaut de l'utilisateur. Si INETD est démarré par /etc/rc ou par /etc/inittab, l'ID utilisateur est hérité du noyau z/OS UNIX, il s'agit de OMVSKERN par défaut.

```
ADDGROUP OMVSGRP OMVS(GID(1))
ADDUSER OMVSKERN DFLTGRP(OMVSGRP) NOPASSWORD +
      OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
```

INETD est un démon qui nécessite l'accès à des fonctions (setuid(), par exemple). L'ID utilisateur utilisé pour démarrer INETD nécessite ainsi des droits d'accès READ au profil BPX.DAEMON dans la classe FACILITY. Dans le cas où ce profil n'est pas défini, un UID 0 est obligatoire.

```
PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(OMVSKERN)
```

L'ID utilisateur INETD nécessite également des droits EXECUTE pour le programme inetd (/usr/sbin/inetd), des droits READ à vos fichiers inetd.conf et ETC.SERVICES et des droits WRITE à /etc/inetd.pid. Si vous désirez exécuter INETD sans UID 0, vous pouvez donner des droits d'accès CONTROL au profil SUPERUSER.FILESYS dans la classe UNIXPRIV afin de fournir les permissions nécessaires pour les fichiers z/OS UNIX.

Les programmes qui nécessitent des droits d'accès de démon doivent être contrôlés par programme si BPX.DAEMON est défini dans la classe FACILITY. Cela est déjà fait pour le programme INETD par défaut (/usr/sbin/inetd), mais doit être défini manuellement si vous utilisez une copie ou une version personnalisée. Utilisez la commande **extattr +p** pour rendre un fichier z/OS UNIX contrôlé par programme. Utilisez la classe RACF PROGRAM pour qu'un module de chargement MVS puisse être contrôlé par programme.

Les programmeurs système qui ont besoin de redémarrer INETD depuis leur session de shell démarreront INETD à l'aide de leurs autorisations. Ils doivent ainsi disposer de la même liste de permissions que les ID utilisateur INETD normaux. Ils doivent de plus disposer de permissions de lister et d'arrêter les processus INETD. Cela peut être accompli de plusieurs façons.

- UID 0

Cette solution n'est pas recommandée pour les ID utilisateur "humains" dans la mesure où il n'y a pas de restrictions liées à z/OS UNIX.

- Droits d'accès READ au profil BPX.SUPERUSER dans la classe FACILITY

Permettent à l'utilisateur d'obtenir un UID 0 par l'intermédiaire de la commande **su**. Il s'agit de la configuration recommandée.

- Droits d'accès aux profils individuels qui couvrent les autorisations obligatoires

- Droits d'accès READ à SUPERUSER.PROCESS.GETPSENT dans la classe UNIXPRIV (pour la commande **ps**)
- Droits d'accès READ à SUPERUSER.PROCESS.KILL dans la classe UNIXPRIV (pour la commande **kill**)
- Droits d'accès READ à BPX.JOBNAME dans la classe FACILITY (pour la variable d'environnement **\_BPX\_JOBNAME**)



Voir le document *UNIX System Services Command Reference* (SA22-7802) pour en savoir plus sur les commandes **extattr** et **su**. Voir le document *UNIX System Services Planning* (GA22-7800) pour en savoir plus sur la classe UNIXPRIV et les profils BPX.\* dans la classe FACILITY. Voir le document *Security Server RACF Security Administrator's Guide* (SA22-7683) pour obtenir davantage d'informations sur les définitions de segment OMVS et la classe PROGRAM.

---

## Exigences de Developer for System z

Developer for System z dépend d'INETD pour la gestion REXEC et/ou SSH. Il peut également imposer des exigences supplémentaires à la configuration d'INETD décrite ci-avant.

REXEC (ou SSH) est utilisé aux fins suivantes, comme indiqué à la section «(Facultatif) Utilisation de REXEC (ou SSH)», à la page 104.

- Actions à distance (basées sur l'hôte) dans les sous-projets z/OS UNIX
- Méthode alternative de démarrage du serveur RSE

Les actions distantes dans les sous-projets z/OS UNIX ne requièrent pas de paramètres spécifique. En revanche, la méthode de démarrage RSE alternative requiert des paramètres spécifiques.

### INETD

Les paramètres environnementaux d'INETD, qui sont transmis lors du démarrage du processus, ainsi que les permissions pour l'ID utilisateur d'INETD doivent être configurés convenablement pour qu'INETD puisse démarrer le serveur RSE.

- Si INETD est lancé par JCL à l'aide de BPXBATCH, la taille de la région doit être 0.
- Si INETD est démarré à partir d'une session de shell TSO/OMVS, la taille de la région TSO doit être 2096128 ou plus.
- Si INETD est démarré par `/etc/rc` ou par `/etc/inittab`, la taille de la région de SYS1.PROCLIB(BPX0INIT) est utilisée, sa valeur par défaut étant 0.

### REXEC (ou SSH)

Le démon REXEC (ou SSH) qui est démarré par INETD lorsqu'un client se connecte au port 512 (ou 22, respectivement) est utilisé pour procéder à l'authentification, démarrer le serveur RSE, et renvoyer le numéro de port pour la suite de la communication en retour vers le client. Pour cela, l'ID utilisateur qui est affecté au démon REXEC (ou SSH) (dans `inetd.conf`) nécessite les autorisations suivantes :

- Définitions OMVS valides dans le logiciel de sécurité ; UID, HOME et PROGRAM doivent être définis, ainsi que le GID pour le groupe par défaut de l'utilisateur
- Droits d'accès READ au profil BPX.DAEMON dans la classe FACILITY
- Droits d'accès READ et EXECUTE aux répertoires d'installation de Developer for System z, `/usr/lpp/rdz/*` par défaut
- Droits d'accès READ et EXECUTE aux répertoires de configuration de Developer for System z, `/etc/rdz/*` par défaut



---

## Annexe D. Configuration APPC

Cette annexe vous aide à résoudre certains incidents fréquents qui peuvent survenir lors de la configuration d'APPC (Advanced Program-to-Program Communication) ou au cours de la vérification ou de la modification d'une configuration existante.

Voir le document *MVS Planning: APPC/MVS Management* (SA22-7599) and *MVS Initialization and Tuning Reference* (SA22-7592) pour de plus amples informations sur la gestion APPC et sur les membres parmlib étudiés ci-après.

Notez que l'ensemble de la configuration d'APPC n'est pas abordé ici, mais uniquement certains aspects clés qui peuvent s'appliquer à votre site.

Le membre SYS1.SAMPLIB(ATBALL) contient une liste et des descriptions de tous les (exemples de) membres liés à APPC dans SYS1.SAMPLIB.

---

### Méthode d'accès VSAM

APPC/MVS stocke ses données de configuration dans les membres SYS1.PARMLIB suivants et dans deux fichiers VSAM :

- Le fichier VSAM du programme de transaction (TP) (nom par défaut : SYS1.APPCTP) contient les informations de planification et de sécurité pour les programmes z/OS.
- Le fichier VSAM des informations complémentaires (SI) (nom par défaut : SYS1.APPCSI) contient la traduction des noms de destination symbolique utilisés par les serveurs APPC/MVS et TP locaux de z/OS.

Un TP est un programme d'application qui utilise APPC pour communiquer avec un TP sur le même système ou sur un autre afin d'accéder à des ressources. La configuration d'APPC pour Developer for System z active un nouveau TP appelé FEKFRSRV, qui est désigné comme le service Commandes TSO.

Le travail suivant est une concaténation des exemples de membre SYS1.SAMPLIB(ATBTPVSM) et SYS1.SAMPLIB(ATBSIVSM) et peut être utilisé pour définir les méthodes d'accès VSAM d'APPC.

```

//APPCVSAM JOB <paramètres du travail>
//*
/* CAUTION: This is neither a JCL procedure nor a complete job.
/* Before using this sample, you will have to make the following
/* modifications:
/* 1. Change the job parameters to meet your system requirements.
/* 2. Change ***** to the volume that will hold the APPC VSAMs.
//*
//TP      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE CLUSTER (NAME(SYS1.APPCTP) -
                        VOLUME(*****)) -
                        INDEXED REUSE -
                        SHAREOPTIONS(3 3) -
                        RECORDSIZE(3824 7024) -
                        KEYS(112 0) -
                        RECORDS(300 150)) -
        DATA      (NAME(SYS1.APPCTP.DATA)) -
        INDEX      (NAME(SYS1.APPCTP.INDEX))
/*
//SI      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE CLUSTER (NAME(SYS1.APPCSI) -
                        VOLUME(*****)) -
                        INDEXED REUSE -
                        SHAREOPTIONS(3 3) -
                        RECORDSIZE(248 248) -
                        KEYS(112 0) -
                        RECORDS(50 25)) -
        DATA      (NAME(SYS1.APPCSI.DATA)) -
        INDEX      (NAME(SYS1.APPCSI.INDEX))
/*

```

Figure 65. Langage JCL pour la création des méthodes VSAM d'APPC

## VTAM

APPC est une implémentation du protocole architecture SNA LU 6.2. L'architecture SNA fournit des formats et des protocoles qui définissent une plage de composants SNA physiques et logiques (l'unité logique (LU), par exemple). LU 6.2 est un type d'unité logique qui est spécialement conçue pour gérer les communications entre les programmes d'application.

Pour utiliser SNA sur MVS, vous devez installer et configurer la méthode d'accès virtuelle en télétraitement VTAM (Virtual Telecommunications Access Method). VTAM doit être active avant que les tâches système APPC puissent être utilisées.

La partie spécifique d'APPC de la configuration de VTAM se compose de trois étapes :

1. Définition du nom de mode utilisé (APPCHOST, par exemple) pour VTAM à l'aide de SYS1.SAMPLIB(ATBLJOB) pour assembler et procéder à l'édition de liens de SYS1.SAMPLIB(ATBLMODE) dans votre SYS1.VTAMLIB. Pour plus de détails, voir le membre SYS1.SAMPLIB(ATBLMODE).
2. Définition d'APPC/MVS comme une application VTAM en copiant l'exemple de membre SYS1.SAMPLIB(ATBAPPL) dans un fichier de la concaténation SYS1.VTAMLST. Pour plus de détails, voir le membre SYS1.SAMPLIB(ATBAPPL).

- Utilisation de la commande de console **v net,act,id=atbappl** pour activer l'application nouvellement définie (où **net** est identique au nom de votre tâche démarrée VTAM). Utilisez la commande de console **d net,appls** pour vérifier que l'application est active. Ajoutez le nom du membre à **SYS1.VTAMLST(ATCCONxx)** si vous voulez qu'il soit actif lorsque VTAM démarre.

L'ACBNAME de MVSLU01 utilisé dans l'exemple de membre **SYS1.SAMPLIB(ATBAPPL)** peut être modifié pour se conformer aux normes du site, mais doit correspondre aux définitions dans le membre **SYS1.PARMLIB(APPCPMxx)**.

```

MVSLU01 APPL  ACBNAME=MVSLU01,          C
                APPC=YES,                C
                AUTOSES=0,                C
                DDRAINL=NALLOW,          C
                DLOGMOD=APPCHOST,         C
                DMINWNL=5,                C
                DMINWNR=5,                C
                DRESPL=NALLOW,           C
                DSESLIM=10,               C
                LMDENT=19,                C
                MODETAB=LOGMODES,         C
                PARSESS=YES,              C
                SECACPT=CONV,              C
                SRBEXIT=YES,              C
                VPACING=1                  C

```

Figure 66. **SYS1.SAMPLIB(ATBAPPL)**

Voir le document *Communications Server IP SNA Network Implementation Guide* (SC31-8777) pour de plus amples informations sur la configuration de VTAM.

---

## **SYS1.PARMLIB(APPCPMxx)**

Pour activer et prendre en charge le flux de conversations entre les systèmes, les sites doivent définir des unités logiques entre lesquelles les sessions peuvent être liées. Un site a besoin de définir au moins une unité logique avant le traitement APPC/MVS, même quand le traitement APPC demeure sur un seul système. Les unités logiques correspondent à certaines des définitions créées dans **SYS1.PARMLIB(APPCPMxx)**.

Le service Commandes TSO requiert de configurer APPC de façon à disposer d'une unité logique qui puisse traiter à la fois les demandes entrantes et sortantes.

La définition d'unité logique doit être ajoutée au membre **SYS1.PARMLIB(APPCPMxx)** et inclure les paramètres **BASE** et **SCHED(ASCH)**. Le membre **APPCPMxx** spécifie également le profil de transaction (TP) et les fichiers VSAM d'informations complémentaires (SI) qui seront utilisés.

L'exemple de code suivant est un exemple de membre **SYS1.PARMLIB(APPCPMxx)** que vous pouvez utiliser pour le service Commandes TSO.

```

LUADD ACBNAME(MVSLU01)
      BASE
      SCHED(ASCH)
      TPDATA(SYS1.APPCTP)
SIDEINFO DATASET(SYS1.APPCSI)

```

Figure 67. **SYS1.PARMLIB(APPCPMxx)**

Lorsqu'un système comporte plusieurs noms d'unité logique, vous devrez peut-être apporter des modifications en fonction de l'unité logique sélectionnée par le

système comme l'unité logique de base (BASE). L'unité logique de base du système est déterminée par les conditions suivantes :

1. L'unité logique de base du système est représentée par la dernière instruction LUADD qui contient à la fois les paramètres NOSCHED et BASE. Ce type d'unité logique permet de traiter les demandes sortantes lorsque aucun planificateur de transactions n'est actif.
2. Si aucune instruction LUADD ne contient à la fois NOSCHED et BASE, l'unité logique de base du système est représentée par la dernière instruction LUADD qui contient le paramètre BASE et spécifie ASCH comme le planificateur de transaction APPC/MVS. Cette action s'effectue par le codage de SCHED(ASCH) ou le non codage du paramètre SCHED (ASCH est la valeur par défaut de SCHED).

**Remarque :** La commande de l'opérateur **D APPC,LU,ALL** affiche toutes les définitions de LU actives et marque la LU de base.

Si votre système comporte une unité logique contenant les paramètres BASE et NOSCHED, elle sera utilisée comme l'unité logique de base mais le Service Commandes TSO ne fonctionnera pas, car cette unité ne possède pas de planificateur de transactions pour traiter les demandes pour la transaction FEKFRSRV. Si cette unité logique ne peut pas être modifiée pour supprimer le paramètre NOSCHED, la variable d'environnement `rsed.envvars` `_FEKFSCMD_PARTNER_LU` peut être définie avec l'unité logique qui possède les paramètres BASE et SCHED(ASCH), par exemple :

```
_FEKFSCMD_PARTNER_LU=MVSLU01
```

Pour plus d'informations sur `rsed.envvars`, voir «`rsed.envvars` - Fichier de configuration RSE», à la page 32.

---

## SYS1.PARMLIB(ASCHPMxx)

Le planificateur de transactions APPC/MVS (le nom par défaut est ASCH) lance et planifie des programmes de transaction en réponse aux demandes entrantes de conversations. Le membre `SYS1.PARMLIB(ASCHPMxx)` contrôle son fonctionnement, avec les définitions de classe de transaction par exemple.

La classe de transaction APPC utilisée pour le service Commandes TSO doit comporter suffisamment de demandeurs APPC pour en autoriser un par utilisateur de Developer for System z.

Le service Commandes TSO nécessite également que les spécifications par défaut soient indiquées dans les sections `OPTIONS` et `TPDEFAULT`.

L'exemple de code suivant est un membre `SYS1.PARMLIB(ASCHPMxx)` que vous pouvez utiliser pour le service Commandes TSO.

```

CLASSADD
  CLASSNAME(A)
  MAX(20)
  MIN(1)
  MSGLIMIT(200)

OPTIONS
  DEFAULT(A)

TPDEFAULT
  REGION(2M)
  TIME(5)
  MSGLEVEL(1,1)
  OUTCLASS(X)

```

Figure 68. SYS1.PARMLIB(ASCHPMxx)

**Remarque :**

- Pour les besoins du débogage, il se peut que le point service IBM vous demande d'augmenter la valeur de MSGLIMIT, de sorte qu'une plus grande quantité de sortie soit écrite dans le fichier journal.
- La commande de l'opérateur **D ASCH,ALL** affiche toutes les classes du planificateur de transactions APPC actives.

---

## Activation des modifications d'APPC

Les modifications de configuration présentées dans les étapes précédentes peuvent maintenant être activées. Il est possible de procéder de différentes manières, selon les circonstances.

- APPC n'est pas encore actif. Saisissez les commandes de console suivantes pour lancer APPC/MVS (où xx représente les deux derniers caractères du membre SYS1.PARMLIB) :

1. S APPC,SUB=MSTR,APPC=xx
2. S ASCH,SUB=MSTR,ASCH=xx

Ajoutez ces commandes à SYS1.PARMLIB(COMMNDxx) pour les lancer au démarrage du système.

- APPC est déjà actif. APPC peut recharger les membres SYS1.PARMLIB de manière dynamique via une commande de console **SET** (où xx représente les deux derniers caractères du membre SYS1.PARMLIB) :

1. SET APPC=xx
2. SET ASCH=xx

Les commandes de la console **D APPC** et **D ASCH** peuvent être employées pour vérifier la configuration d'APPC. Voir le document *MVS System Commands* (GC28-1781) pour de plus amples informations sur les commandes de la console.

---

## Définition de la transaction du service Commandes TSO

Une fois qu'APPC/MVS est actif, le service Commandes TSO de Developer for System z peut être défini, voir «(Facultatif) Transaction APPC pour le service Commandes TSO», à la page 106.

La définition de la transaction APPC décrite s'effectue par personnalisation et soumission de FEK.#CUST.JCL(FEKAPPC).

La transaction APPC peut également être définie en mode interactif par l'intermédiaire de l'interface ISPF APPC, comme cela est décrit dans le livre blanc. Ce livre blanc explique également comment configurer la transaction APPC afin de collecter des informations statistiques propres aux utilisateurs.

Le livre blanc *APPC and WebSphere Developer for System z* (SC23-5885-00) est disponible dans la bibliothèque Internet de Developer for System z, <http://www-306.ibm.com/software/awdtools/rdz/library/>.

**Remarque :** Le programme transactionnel (TP) en langage JCL qui est utilisé par APPC pour lancer le service Commandes TSO a changé dans Developer for System z version 7.1. Si vous suivez les instructions du livre blanc pour définir le TP, vous devez ajouter le mot clé NESTMACS à la ligne PARM, par exemple :

```
// PARM='ISPSTART CMD(%FEKFRSRV TIMEOUT=60) NEWAPPL(ISR) NESTMACS'
```

---

## (Facultatif) Options de configuration alternatives

Developer for System z prend en charge les options de configuration alternatives APPC et VTAM dont certaines sont décrites dans la présente section.

### Nom de transaction alternatif

Le nom de transaction par défaut pour le service Commandes TSO est FEKFRSRV, comme décrit à la section «(Facultatif) Transaction APPC pour le service Commandes TSO», à la page 106. Comme détaillé dans la section, ce nom peut être modifié lorsque vous définissez la transaction sur APPC.

Notez que la modification du nom de transaction dans APPC implique que le nouveau nom soit attribué à `_FEKFSCMD_TP_NAME_` dans `rsed.envvars`, comme indiqué dans «rsed.envvars - Fichier de configuration RSE», à la page 32.

### Unités logiques multiples

APPC est un protocole de communication qui permet à un programme (le noeud partenaire) d'interagir avec un programme situé sur l'hôte (le noeud local). Avec Developer for System z, le noeud partenaire (serveur Commandes TSO) et le noeud local (serveur RSE) sont actifs dans le même système z/OS. Par défaut, ils utilisent la même définition d'unité logique (BASE) pour communiquer entre-eux.

Vous pouvez spécifier un nom d'unité logique partenaire alternatif pour le service Commandes TSO dans la directive `_FEKFSCMD_PARTNER_LU_` de `rsed.envvars` (voir «rsed.envvars - Fichier de configuration RSE», à la page 32). Notez que vous ne pouvez pas modifier l'unité logique locale qui doit toujours être une unité logique BASE valide (doit posséder les mots clés BASE et SCHED).

### Sécurité des unités logiques

VTAM prend en charge une configuration APPC sécurisée dans laquelle la communication entre les unités logiques partenaire et locale doit être définie dans le logiciel de sécurité.

Vous l'activez en ajoutant `VERIFY=REQUIRED` à la définition VTAM de l'unité logique (BASE) locale. Les définitions de sécurité doivent être créées dans la classe APPCLU, tel que spécifié dans le document *MVS Planning: APPC/MVS Management* (SA22-7599).

Notez que lorsque cette configuration est active dans VTAM et que la configuration de votre logiciel de sécurité n'est pas achevée, l'initialisation de la communication avec le service Commandes TSO échouera et aucun message dans le journal système ne mentionnera le refus de VTAM de configurer la connexion. Le test du programme de vérification d'installation APPC (fekfi vpa) échouera avec le message "Return code 1 - Allocate Failure no retry".





---

## Annexe E. Conditions requises

La présente annexe répertorie les conditions requises et les corequis liés à l'hôte correspondant à cette version de Developer for System z.

Voir *Rational Developer for System z Prerequisites* (SC23-7659) dans la bibliothèque en ligne de Developer for System z (<http://www-01.ibm.com/software/awdtools/rdz/library/>) pour obtenir une aide en ligne des conditions requises obligatoires et facultatives.

Les produits répertoriés dans la présente section sont disponibles à la date de publication du présent document. Consultez le site Web IBM Software Support Lifecycle à l'adresse <http://www.ibm.com/software/support/lifecycle/> pour savoir si un produit sélectionné est toujours disponible au moment où vous souhaitez utiliser la fonction Developer for System z.

---

### Éléments prérequis de l'hôte z/OS

Pour utiliser Developer for System z, vous devez disposer de l'environnement ci-dessous avec les éléments prérequis appropriés :

#### z/OS

L'un des niveaux suivants doit être installé sur le système hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5694-A01	z/OS v 1.11	ISPF : <ul style="list-style-type: none"><li>• APAR OA29489 (Passerelle client TSO/ISPF) PTF UA51713</li></ul> TCP/IP : <ul style="list-style-type: none"><li>• Aucune PTF ou aucun niveau de service requis</li></ul>
5694-A01	z/OS v 1.10	ISPF : <ul style="list-style-type: none"><li>• APAR OA29489 (Passerelle client TSO/ISPF) PTF UA51712</li></ul> TCP/IP : <ul style="list-style-type: none"><li>• APAR PK74282 (croissance de mémoire fixe CSM) PTF UK41810</li></ul>
5694-A01	z/OS v 1.9	ISPF : <ul style="list-style-type: none"><li>• APAR OA29489 (Passerelle client TSO/ISPF) PTF UA51687</li></ul> TCP/IP : <ul style="list-style-type: none"><li>• APAR PK74282 (croissance de mémoire fixe CSM) PTF UK41812</li></ul>

Numéro de programme	Nom	PTF ou niveaux de service requis
5694-A01	z/OS v 1.8	<p>ISPF :</p> <ul style="list-style-type: none"> <li>• APAR OA20345 (sortie de commande imbriquée) PTF UA33575</li> <li>• APAR OA20449 (ajout de la prise en charge de NESTMACS) PTF UA34052</li> <li>• APAR OA29489 (Passerelle client TSO/ISPF) PTF UA51686</li> </ul> <p>TCP/IP :</p> <ul style="list-style-type: none"> <li>• APAR PK74282 (croissance de mémoire fixe CSM) PTF UK41811</li> </ul>

Le site Web du produit associé est :

<http://www-03.ibm.com/systems/z/os/zos/>

**Remarques :**

1. Les actions à distance (basées sur l'hôte) pour les sous-projets z/OS UNIX nécessitent que la version z/OS UNIX de REXEC ou SSH soit active sur l'hôte.
2. z/OS comprend les composants suivants, qui doivent être installés, configurés et en état de fonctionner :
  - Utilitaire ISPF (Interactive System Productivity Facility)  
– <http://www-01.ibm.com/software/awdtools/ispf/>
  - Language Environment  
– <http://www-03.ibm.com/servers/eserver/zseries/zos/le/>
  - RACF ou produit de sécurité équivalent  
– <http://www-03.ibm.com/servers/eserver/zseries/zos/racf/>
  - Composant VTAM d'IBM Communications Server  
– <http://www-01.ibm.com/software/network/commserver/zos/>
  - Composant Services IP d'IBM Communications Server  
– <http://www-01.ibm.com/software/network/commserver/zos/>
  - Editeur de liens
  - APPC (facultatif)

**Remarque :**

- APPC est un composant obligatoire si le service pour l'APAR ISPF OA29489 n'est pas disponible sur votre système hôte.
- APPC peut être remplacé par la fonctionnalité de passerelle client ISPF fournie dans ISPF for z/OS 1.10 et disponible dans ISPF for z/OS 1.8 et 1.9 avec les PTF appropriées appliquées.

## SMP/E

Pour installer Developer for System z, l'un des niveaux suivants doit être installé :

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-G44	IBM System Modification Program Extended (SMP/E) for z/OS v 3.5	Aucune PTF ou aucun niveau de service requis
5655-G44	IBM System Modification Program Extended (SMP/E) for z/OS v 3.4	Aucune PTF ou aucun niveau de service requis

Le site Web du produit associé est :

<http://www-03.ibm.com/systems/z/os/zos/smpe/>

## SDK for z/OS Java 2 Technology Edition

Pour prendre en charge les applications qui utilisent Remote Systems Explorer (RSE), l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-R32	IBM 64-bit SDK for z/OS, Java 2 Technology Edition, v 6.0	édition de service 7
5655-R31	IBM 31-bit SDK for z/OS, Java 2 Technology Edition, v 6.0	édition de service 7
5655-N99	IBM 64-bit SDK for z/OS, Java 2 Technology Edition, v 5.0	édition de service 11
5655-N98	IBM 31-bit SDK for z/OS, Java 2 Technology Edition, v 5.0	édition de service 11

Le site Web du produit associé est :

<http://www.ibm.com/servers/eserver/zseries/software/java/>

**Remarque :** Les PTF destinées à Developer for System z APAR PM07305 doivent être appliquées si une version 64 bits de Java est utilisée. Les PTF sont accessibles via la page de service recommandée Developer for System z, <http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335>.

---

## Corequis pour l'hôte z/OS

Les produits figurant dans la présente section et les logiciels suivants sont requis pour prendre en charge les composants spécifiques de Developer for System z. Le poste de travail client Developer for System z peut être installé sans ces logiciels. Pour pouvoir utiliser le composant, vous devez toutefois installer le logiciel hôte requis indiqué (qui doit être opérationnel à l'exécution) .

## z/OS

Numéro de programme	Nom	PTF ou niveaux de service requis
5694-A01	z/OS v 1.11	HLASM Aucune PTF ou aucun niveau de service requis  XL C/C++ Aucune PTF ou aucun niveau de service requis  SCLM Aucune PTF ou aucun niveau de service requis  LE (PL/I) Aucune PTF ou aucun niveau de service requis  TN3270 Aucune PTF ou aucun niveau de service requis
5694-A01	z/OS v 1.10	HLASM Aucune PTF ou aucun niveau de service requis  XL C/C++ Aucune PTF ou aucun niveau de service requis  SCLM Aucune PTF ou aucun niveau de service requis  LE (PL/I) Aucune PTF ou aucun niveau de service requis  TN3270 Aucune PTF ou aucun niveau de service requis
5694-A01	z/OS v 1.9	HLASM Aucune PTF ou aucun niveau de service requis  XL C/C++ Aucune PTF ou aucun niveau de service requis  SCLM • APAR OA27379 (recherche SCLM) PTF UA46330 + UA46331, UA46332, UA46333, UA46334 (dépend du langage)  LE (PL/I) Aucune PTF ou aucun niveau de service requis  TN3270 Aucune PTF ou aucun niveau de service requis

Numéro de programme	Nom	PTF ou niveaux de service requis
5694-A01	z/OS v 1.8	HLASM Aucune PTF ou aucun niveau de service requis  XL C/C++ Aucune PTF ou aucun niveau de service requis  SCLM <ul style="list-style-type: none"> <li>• APAR OA21104 (mode de génération informatif) PTF UA35046 + UA35056, UA35057, UA35058 ou UA35059 (en fonction de la langue)</li> <li>• APAR OA16924 (amélioration de SCLMINFO) PTF UA29772 + UA29922, UA29923, UA29924 ou UA29925 (en fonction de la langue)</li> <li>• APAR OA16804 (ajout d'un support d'ID utilisateur de substitution) PTF UA33524 + UA33533, UA33534, UA33535 ou UA33536 (en fonction de la langue)</li> </ul> LE (PL/I) <ul style="list-style-type: none"> <li>• APAR PK41552 (nouveaux messages PL/I pour Developer for System z ) PTF UK24482 (Anglais) ou UK24483 (Japonais)</li> </ul> TN3270 Aucune PTF ou aucun niveau de service requis

Le site Web du produit associé est :

<http://www-03.ibm.com/systems/z/os/zos/>

**Remarque :**

1. JES3 v 1.10 ou ultérieure est un élément corequis pour les utilisateurs de JES3 qui souhaitent utiliser le support du moniteur de travaux pour afficher la sortie des travaux actifs.
2. L'assembleur de haut niveau (HLASM) doit être installé sur l'hôte avec le niveau de service répertorié, afin de compiler les programmes assembleurs développés ou édités dans Developer for System z.

Le site Web du produit associé est :

<http://www.ibm.com/software/awdtools/hlasm/>

3. Le compilateur XL C/C++ doit être installé sur l'hôte avec le niveau de service répertorié, afin de compiler les programmes C/C++ développés ou édités dans Developer for System z.

Le site Web du produit associé est :

<http://www.ibm.com/software/awdtools/czos/>

4. SCLM doit être installé sur l'hôte avec le niveau de service répertorié, afin de prendre en charge SCLM Developer Toolkit.

Le site Web du produit associé est :

<http://www.ibm.com/software/awdtools/sclmsuite/sclm/>

**Remarque :**

- L'APAR OA16804 est obligatoire uniquement si vous souhaitez utiliser un déploiement, une promotion et une génération sécurisés.
  - APAR OA26997 est obligatoire uniquement pour la prise en charge de la sécurité membre.
  - APAR OA27379 est obligatoire uniquement pour la prise en charge de la sécurité membre ou la fonctionnalité de recherche SCLM.
5. Language Environment doit être installé sur l'hôte avec le niveau de service répertorié, afin de prendre en charge Enterprise Service Tools pour PL/I.

Le site Web du produit associé est :

<http://www-03.ibm.com/servers/eserver/zseries/zos/le/>

6. TN3270 doit être installé sur l'hôte avec le niveau de service répertorié, afin de prendre en charge l'émulateur Host Connect. TN3270 fait partie du composant IP Services d'IBM Communications Server.

Le site Web du produit associé est :

<http://www-01.ibm.com/software/network/commserver/zos/>

## Compilateur COBOL

Pour compiler les programmes COBOL développés ou édités dans Developer for System z, l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-S71	IBM Enterprise COBOL for z/OS v 4.2	Aucune PTF ou aucun niveau de service requis
5655-S71	IBM Enterprise COBOL for z/OS v 4.1	Aucune PTF ou aucun niveau de service requis
5535-G53	IBM Enterprise COBOL for z/OS v 3.4	Aucune PTF ou aucun niveau de service requis

Le site Web du produit associé est :

<http://www.ibm.com/software/awdtools/cobol/zos/>

**Remarque :** IBM Enterprise COBOL for z/OS v 4.1 est requis pour qu'Enterprise Service Tools puisse générer la conversion XML compilée qui utilise la fonction XMLSS-based XML PARSE dans COBOL v 4.1.

## Compilateur PL/I

Pour compiler les programmes PL/I développés ou édités dans Developer for System z, l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-H31	IBM Enterprise PL/I for z/OS v 3.9	Aucune PTF ou aucun niveau de service requis

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-H31	IBM Enterprise PL/I for z/OS v 3.8	Aucune PTF ou aucun niveau de service requis
5655-H31	IBM Enterprise PL/I for z/OS v 3.7	Aucune PTF ou aucun niveau de service requis
5655-H31	IBM Enterprise PL/I for z/OS v 3.6	Aucune PTF ou aucun niveau de service requis

Le site Web du produit associé est :

<http://www.ibm.com/software/awdtools/pli/plizos/>

## Debug Tool for z/OS

Pour prendre en charge le débogage à distance à partir de Developer for System z, l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	Langage de programmation	APAR, PTF ou niveaux de service requis
5655-V50	IBM Debug Tool for z/OS V10.1	COBOL, PL/I, C/C++, assembleur et autres fonctions	Toute maintenance disponible
5655-U27	IBM Debug Tool for z/OS V9.1	COBOL, PL/I, C/C++, assembleur et autres fonctions	Toute maintenance disponible
5655-S16	IBM Debug Tool Utilities and Advanced Functions for z/OS V8.1.0	COBOL, PL/I, C/C++, assembleur et autres fonctions	Toute maintenance disponible
5655-S17	IBM Debug Tool for z/OS V8.1.0	COBOL, PL/I, Assembleur, C/C++	Toute maintenance disponible

**Remarque :** La prise en charge des configurations de débogage CICS dans IBM Rational Developer for System z v7.6.1 ou supérieur nécessite IBM Debug Tool v10.1 ou v9.1 (numéro PTF - UK52904).

Le site Web du produit associé est :

<http://www.ibm.com/software/awdtools/debugtool/>

**Remarque :** Debug Tool Utilities and Advanced Functions (DTU&AF) est une version élaborée de Debug Tool.

Dès la version 9, Debug Tool for z/OS et Debug Tool Utilities and Advanced Functions ont été fusionnés en une seule offre.

## CICS Transaction Server

Pour prendre en charge les applications avec des instructions CICS imbriquées, l'un des niveaux suivants doit être installé :

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-S97	IBM CICS Transaction Server for z/OS v 4.1	Aucune PTF ou aucun niveau de service requis
5697-E93	IBM CICS Transaction Server for z/OS v 3.2	UK34221
5697-E93	IBM CICS Transaction Server for z/OS v 3.1	UK15767, UK15764, UK11782, UK11294, UK12233, UK12521, UK15261, UK15271, UK34221, UK34078

Le site Web du produit associé est :

<http://www.ibm.com/software/http/cics/tserver/>

**Remarque :**

- Pour utiliser CICS Transaction Server avec l'outil de débogage, vous devez effectuer des tâches de configuration supplémentaires.
- L'interface RESTful disponible dans CICS Transaction Server v 4.1 ou ultérieures est obligatoire pour prendre en charge les nouvelles fonctions d'Application Deployment Manager, de Service Component Architecture et d'Enterprise Service Tools dans IBM Rational Developer for System z v 7.6 ou ultérieures.
- CICS Transaction Server v 3.2 ou ultérieures est obligatoire pour la prise en charge de la plupart des fonctions d'Enterprise Service Tools.

Pour obtenir la liste des exigences spécifiques d'exécution, reportez-vous à la documentation relative à Enterprise Service Tools dans le centre de documentation de IBM Rational Developer for System z, à l'adresse suivante : <http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/>.

- Vous devez disposer au moins de CICS Transaction Server v 3.1 avec le service UK34221 pour Application Deployment Manager.

## IMS

Pour prendre en charge les applications utilisant la base de données et les transmissions de données IMS, l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5635-A02	IBM IMS v 11.1	Aucune PTF ou aucun niveau de service requis
5635-A01	IBM IMS v 10.1	Aucune PTF ou aucun niveau de service requis
5655-J38	IBM IMS v 9.1	Aucune PTF ou aucun niveau de service requis

Le site Web du produit associé est :

<http://www.ibm.com/software/data/ims/ims/>



**Remarque :**

- Pour fonctionner avec l'outil de débogage, IMS requiert des une configuration supplémentaire.
- La version 10.1 ou ultérieures d'IMS, IMS Connect et IMS SOAP Gateway sont obligatoires pour Enterprise Service Tools.

**DB2 for z/OS**

Pour prendre en charge DB2, l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5635-DB2	IBM DB2 for z/OS v 9.1	Aucune PTF ou aucun niveau de service requis
5625-DB2	IBM DB2 Universal Database for z/OS v 8.1	Aucune PTF ou aucun niveau de service requis

Le site Web du produit associé est :

<http://www.ibm.com/software/data/db2/zos/>

**Rational Team Concert for System z**

Pour le contrôle des sources Jazz utilisant les projets distants de Developer for System z, le niveau suivant doit être installé.

Numéro de programme	Nom	PTF ou niveaux de service requis
5724-V82	Rational Team Concert for System z Server v 2.0	<p>Identificateur de modification de fonction HAHA200 – Serveur coopératif</p> <ul style="list-style-type: none"><li>• UK54064</li><li>• UK54071</li><li>• UK54073</li><li>• UK54095</li><li>• UK54098</li></ul> <p>Identificateur de modification de fonction HAHB200 – Kit d'outils</p> <ul style="list-style-type: none"><li>• UK54065</li><li>• UK54066</li><li>• UK54099</li></ul> <p>Identificateur de modification de fonction HAHC200 – Moniteur de travaux</p> <ul style="list-style-type: none"><li>• Aucune PTF ou aucun niveau de service requis</li></ul> <p>Identificateur de modification de fonction HAHD200 – Agent BuildForge</p> <ul style="list-style-type: none"><li>• UK54097</li></ul>

Le site Web du produit associé est :

<http://www-01.ibm.com/software/awdtools/rtcz/>

**Remarque :** Rational Team Concert Server v 1.0 ou Rational Team Concert for System z Server v 1.0.1 offre une prise en charge sélective de certaines fonctions de Developer for System z (les projets locaux, par exemple). Il est commandé d'utiliser Rational Team Concert for System z Server v 2.0.0.2 pour utiliser toutes les fonctions intégrées

## Gestionnaire de fichiers

Pour prendre en charge l'intégration de File Manager, l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-U29	IBM File Manager for z/OS v 10.1	• UK54428

Le site Web du produit associé est :

<http://www.ibm.com/software/awdtools/filemanager/>

## Fault Analyzer

Pour prendre en charge l'intégration de Fault Analyzer, les niveaux suivants doivent être installés sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5655-V51	IBM Fault Analyzer v 10.1	Aucune PTF ou aucun niveau de service requis
5655-U28	IBM Fault Analyzer v 9.1	Aucune PTF ou aucun niveau de service requis
5655-S15	IBM Fault Analyzer v 8.1	Aucune PTF ou aucun niveau de service requis

Le site Web du produit associé est le suivant :

<http://www.ibm.com/software/awdtools/faultanalyzer/>

## REXX

Pour utiliser SCLM Developer Toolkit, l'un des niveaux suivants doit être installé sur l'hôte :

Numéro de programme	Nom	PTF ou niveaux de service requis
5695-014	Bibliothèque IBM pour REXX sous zSeries v 1.4	Aucune PTF ou aucun niveau de service requis
5695-014	Bibliothèque IBM pour REXX sous zSeries - Alternate Library v 1.4.0 (FMIDs HWJ9143, JWJ9144)	Aucune PTF ou aucun niveau de service requis

Une version de REXX/370 Alternate Library est disponible sur le site Web du produit :

<http://www.ibm.com/software/awdtools/rexx/rexxzseries/>

## Ported Tools

IBM Ported Tools for z/OS doit être installé (sous z/OS UNIX) pour autoriser le déploiement sécurisé par le biais de sftp ou scp dans SCLM Developer Toolkit.

Une version d'IBM Ported Tools for z/OS est disponible sur le site Web du produit :

[http://www-03.ibm.com/servers/eserver/zseries/zos/unix/port\\_tools.html](http://www-03.ibm.com/servers/eserver/zseries/zos/unix/port_tools.html)

## Ant

Apache Ant doit être installé (dans z/OS UNIX) pour procéder aux générations JAVA/J2EE dans SCLM Developer Toolkit.

Apache Ant est un outil de génération à source ouverte Java que vous pouvez télécharger à partir du site Web du produit :

<http://ant.apache.org/>

## Endevor

Vous devez installer CA Endevor Software Change Manager Release 12 pour utiliser Developer for System z Interface for CA Endevor SCM.

CA Endevor SCM est un produit de CA. Le site Web du produit associé est :

<http://www.ca.com/us/products/product.aspx?ID=259>



---

# Bibliographie

---

## Publications référencées

Les publications suivantes sont référencées dans ce document :

Tableau 52. Publications référencées

Titre de la publication	Référence de la commande	Référence	Site Web de référence
Java Diagnostic Guide	SC34-6650	Java 5.0	<a href="http://www.ibm.com/developerworks/java/jdk/diagnosis/">http://www.ibm.com/developerworks/java/jdk/diagnosis/</a>
Java SDK and Runtime Environment User Guide	/	Java 5.0	<a href="http://www-03.ibm.com/servers/eserver/zseries/software/java/">http://www-03.ibm.com/servers/eserver/zseries/software/java/</a>
Program Directory for IBM Rational Developer for System z	GI11-7314	Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
Rational Developer for System z Common Access Repository Manager Developer's Guide	SC23-7660	Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
Rational Developer for System z Prerequisites	SC23-7659	Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
Rational Developer for System z - Guide de démarrage rapide de configuration de l'hôte	GI11-7313	Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
Rational Developer for System z - Guide de planification du système hôte	GI11-7244	Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
SCLM Developer Toolkit - Guide d'administration	SC11-6464	Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
APPC and WebSphere Developer for System z	SC23-5885	Livre blanc	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
Communications Server IP Configuration Guide	SC31-8775	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server IP Configuration Reference	SC31-8776	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server IP Diagnosis Guide	GC31-8782	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server IP System Administrator's Commands	SC31-8781	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server SNA Network Implementation Guide	SC31-8777	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Communications Server SNA Operations	SC31-8779	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>

Tableau 52. Publications référencées (suite)

Titre de la publication	Référence de la commande	Référence	Site Web de référence
Cryptographic Services System SSL Programming	SC24-5901	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
DFSMS Macro Instructions for Data Sets	SC26-7408	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
DFSMS Using data sets	SC26-7410	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Language Environment Customization	SA22-7564	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Language Environment Debugging Guide	GA22-7560	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS JCL Reference	SA22-7597	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Planning APPC/MVS Management	SA22-7599	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS Planning Workload Management	SA22-7602	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
MVS System Commands	SA22-7627	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
TSO/E Customization	SA22-7783	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
TSO/E REXX Reference	SA22-7790	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
UNIX System Services Command Reference	SA22-7802	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
UNIX System Services Planning	GA22-7800	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
UNIX System Services User's Guide	SA22-7801	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Using REXX and z/OS UNIX System Services	SA22-7806	z/OS 1.9	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Resource Definition Guide	SC34-6430	CICSTS 3.1	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>
Resource Definition Guide	SC34-6815	CICSTS 3.2	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>

Tableau 52. Publications référencées (suite)

Titre de la publication	Référence de la commande	Référence	Site Web de référence
Resource Definition Guide	SC34-7000	CICSTS 4.1	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html">https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html</a>
RACF Security Guide	SC34-6454	CICSTS 3.1	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>
RACF Security Guide	SC34-6835	CICSTS 3.2	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>
RACF Security Guide	SC34-7003	CICSTS 4.1	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html">https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp?topic=/com.ibm.cics.ts.home.doc/library/library_html.html</a>
Language Reference	SC27-1408	Enterprise COBOL for z/OS	<a href="http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html">http://www-03.ibm.com/systems/z/os/zos/bkserv/zapplsbooks.html</a>

Les sites Web suivants sont référencés dans le présent document :

Tableau 53. Sites Web référencés

Description	Site Web de référence
Centre de documentation de Developer for System z	<a href="http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp">http://publib.boulder.ibm.com/infocenter/ratdevz/v7r6/index.jsp</a>
Support de Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/support/">http://www-306.ibm.com/software/awdtools/rdz/support/</a>
Bibliothèque de Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/library/">http://www-306.ibm.com/software/awdtools/rdz/library/</a>
Page d'accueil de Developer for System z	<a href="http://www-306.ibm.com/software/awdtools/rdz/">http://www-306.ibm.com/software/awdtools/rdz/</a>
Developer for System z - Service recommandé	<a href="http://www-01.ibm.com/support/docview.wss?rs=2294&amp;context=SS2QJ2&amp;uid=swg27006335">http://www-01.ibm.com/support/docview.wss?rs=2294&amp;context=SS2QJ2&amp;uid=swg27006335</a>
Developer for System z - Demande d'amélioration	<a href="https://www.ibm.com/developerworks/support/rational/rfe/">https://www.ibm.com/developerworks/support/rational/rfe/</a>
Bibliothèque Internet z/OS	<a href="http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/">http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/</a>
Centre de documentation CICSTS	<a href="https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp">https://publib.boulder.ibm.com/infocenter/cicsts/v4r1/index.jsp</a>
Télécharger Apache Ant	<a href="http://ant.apache.org/">http://ant.apache.org/</a>
Documentation du l'outil de clé Java	<a href="http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html">http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html</a>
Page d'accueil du support de l'autorité de certification	<a href="https://support.ca.com/">https://support.ca.com/</a>

## Publications d'information

Les publications suivantes peuvent s'avérer utiles pour vous aider à comprendre les incidents de configuration pour les composants hôte requis :

Tableau 54. Publications d'information

Titre de la publication	Référence de la commande	Référence	Site Web de référence
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>

Tableau 54. Publications d'information (suite)

Titre de la publication	Référence de la commande	Référence	Site Web de référence
Guide du programmeur système pour : Workload Manager	SG24-6472	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	<a href="http://www.redbooks.ibm.com/">http://www.redbooks.ibm.com/</a>



---

## Glossaire

### A

**Action de verrouillage :** Verrouille un membre.

**Attribut bidirectionnel :** Type de texte, orientation du texte, Permutation numérique et Permutation symétrique.

### B

**Base de données :** Collection d'éléments de données liés entre eux ou indépendants, stockés ensemble, destinée à être utilisée dans une ou plusieurs applications.

**Bibliothèque de chargement :** Bibliothèque contenant des modules de chargement.

**Bidirectionnel (bidi) :** Caractérise des scripts, tels que l'arabe et l'hébreu, qui s'exécutent généralement de droite à gauche, à l'exception des nombres, qui s'exécutent de gauche à droite. La définition de ce terme est extraite du glossaire Localization Industry Standards Association (LISA).

### C

**Compiler :**

1. Dans les langages Integrated Language Environment (ILE), traduire des instructions source en modules qui peuvent ensuite être associés en programmes ou programmes de service.
2. Traduire tout ou partie d'un programme exprimé en langage haut niveau en un programme exprimé en langage intermédiaire, un langage d'assemblage ou un langage machine.

**Conteneur :**

1. Dans CoOperative Development Environment/400, objet système qui contient et organise des fichiers source. Par exemple, un conteneur peut être une bibliothèque i5/OS ou un fichier partitionné pour MVS.
2. Dans l'architecture J2EE, entité qui fournit la gestion du cycle de vie, la sécurité, le déploiement et des services d'exécution pour les composants. (Sun) Chaque type de conteneur (EJB, Web, JSP, servlet, applet et client d'application) fournit également des services spécifiques des composants.
3. Dans Backup Recovery and Media Services, objet physique utilisé pour stocker ou déplacer des supports, tels qu'une case, un chemin ou une armoire.

4. Dans un serveur Virtual Tape Server (VTS), réceptacle dans lequel un ou plusieurs volumes logiques exportés (LVOL) peuvent être stockés. Un volume empilé qui contient un ou plusieurs LVOL et qui réside en dehors d'une bibliothèque VTS est considéré comme le conteneur de ces volumes.
5. Lieu de stockage physique des données. Par exemple, un fichier, un répertoire ou un périphérique.
6. Colonne ou rang utilisé pour la disposition d'un portlet ou d'un autre conteneur sur une page.
7. Élément de l'interface utilisateur qui contient des objets. Dans le gestionnaire de dossier, objet qui contient les autres dossiers ou documents.

### D

**Déboguer :** Détecter, diagnostiquer et éliminer les erreurs des programmes.

**Demande de génération :** Demande du client pour effectuer une transaction de génération.

**Désinstallation en mode silencieux :** Processus de désinstallation qui n'envoie pas les messages vers la console mais stocke les messages et erreurs dans des fichiers journaux après que la commande d'installation a été appelée.

### F

**Fichier :** Unité principale de stockage et d'extraction des données, qui est constituée d'une collection de données disposée selon une des structures imposées et décrites par les données de contrôle auxquelles le système a accès.

**Fichier de réponses :**

1. Fichier qui contient un ensemble de réponses prédéfinies aux questions envoyées par un programme afin d'éviter d'entrer ces valeurs une par une.
2. Fichier ASCII qui peut être personnalisé au moyen des données de configuration pour automatiser une installation. Les données de configuration sont généralement entrées lors d'une installation interactive alors qu'un fichier de réponses permet d'effectuer l'installation sans aucune intervention.

### I

**ID action :** Identificateur numérique d'une action entre 0 et 999

**Installation en mode silencieux :** Installation qui n'envoie pas les messages vers la console mais stocke les messages et les erreurs dans des fichiers journaux. Une installation en mode silencieux peut également utiliser des fichiers de réponses pour entrer les données.

**Instance de référentiels :** Projet ou composant existant dans un SCM.

**Interactive System Productivity Facility (ISPF) :** Logiciel sous licence IBM servant d'éditeur plein écran et de gestionnaire de boîte de dialogue. Utilisé dans l'écriture de programmes d'application, il permet de générer des panneaux d'écran standard et des boîtes de dialogue interactives entre le programmeur et l'utilisateur final. ISPF est constitué de quatre composants principaux : DM, PDF, SCLM et C/S. Le composant DM (Dialog Manager) est le gestionnaire qui fournit des services pour les boîtes de dialogue et les utilisateurs finaux. Le composant PDF (Program Development Facility) offre des services d'aide au développeur de boîtes de dialogue ou d'applications. Le composant SCLM (Software Configuration Library Manager) offre aux développeurs d'applications des services destinés à leurs bibliothèques de développement d'applications. Le composant C/S (Client/Server), qui permet d'exécuter ISPF sur un poste de travail programmable, d'afficher les panneaux au moyen de la fonction d'affichage sur le système d'exploitation de votre poste de travail et d'intégrer des outils et données de poste de travail au moyen des outils et des données de l'hôte.

**Interpréteur :** Programme qui traduit et exécute successivement toutes les instructions en langage de programmation haut niveau.

**Interpréteur de commandes :** Interface logicielle entre les utilisateurs et le système d'exploitation qui interprète les commandes et les interactions utilisateur et les communique au système d'exploitation. Un ordinateur possède des interpréteurs de commandes sur plusieurs niveaux, qui correspondent aux différents niveaux d'interaction avec l'utilisateur.

**Isomorphe :** A chaque élément composé (c'est-à-dire, contenant d'autres éléments) du document d'instance XML lancé à partir de la racine correspond un seul et unique élément de groupe COBOL dont la profondeur d'imbrication est identique à la profondeur d'imbrication de son équivalent XML. Chaque élément non composé (à savoir, ne contenant pas d'autres éléments) dans le document d'instance XML, en partant du haut, comporte une seule donnée élémentaire COBOL correspondante dont la profondeur d'imbrication est identique à la profondeur d'imbrication de son équivalent XML et dont l'adresse mémoire lors de l'exécution peut être identifiée de manière unique.

## L

**Liste des tâches :** Liste des procédures qui peuvent être exécutées par un seul flux de contrôle.

## M

**Mémoire tampon des erreurs :** Partie de mémoire servant à contenir provisoirement les données de sortie des erreurs.

## N

**Nom d'interpréteur de commandes :** Nom de l'interface de l'interpréteur de commandes.

**Non isomorphe :** Mappage simple d'éléments COBOL et d'éléments XML faisant partie de documents XML et de groupes COBOL de forme non identique (non isomorphe). Un mappage non isomorphe peut également être créé entre des éléments non isomorphes de structures isomorphes.

## P

**Passerelle :**

1. Composant intermédiaire qui relie Internet aux environnements intranet lors des appels de service Web.
2. Logiciel qui fournit des services entre les points d'arrêt final et le reste de l'environnement Tivoli.
3. Composant de Voice over Internet Protocol constituant un pont entre VoIP et les environnements commutés par circuit.
4. Périphérique ou programme utilisé pour la connexion de réseaux ou systèmes à d'autres architectures réseau. Ces systèmes peuvent présenter des caractéristiques différentes, telles que des protocoles de communication différents, une architecture réseau ou des stratégies de sécurité différentes, la passerelle réalisant alors leur traduction et leur connexion.

**Perspective :** Groupe de vues présentant les divers aspects des ressources d'un plan de travail. L'utilisateur du plan de travail peut basculer entre les perspectives en fonction de la tâche en cours et personnaliser l'affichage des vues et des éditeurs depuis la perspective.

**Perspective Systèmes distants :** Offre une interface permettant de gérer des systèmes distants par l'intermédiaire de conventions similaires à ISPF.

## R

**RAM :** Repository Access Manager

### Référentiel :

1. Zone de stockage des données. Chaque référentiel comporte un nom et un type d'élément métier associé. Par défaut, son nom sera le même que celui de l'élément métier. Par exemple, un référentiel de factures sera appelé Factures. Il existe deux types de référentiels d'information : local (spécifiques du processus) et global (réutilisable).
2. Fichier VSAM dans lequel sont stockés les états des processus du BTS. Lorsqu'un processus ne s'exécute pas sous le contrôle du BTS, son état (et les états des tâches qui le composent) sont protégés en écriture dans un fichier de référentiel. Les états de tous les processus d'un type de processus donné (et les instances de leurs tâches) sont stockés dans le même fichier de référentiel. Les enregistrements des types multiprocessus peuvent être écrits dans ce même référentiel.
3. Zone de stockage permanente du code source et des autres ressources d'application. Dans un environnement de programmation en équipe, un référentiel partagé permet à plusieurs utilisateurs d'accéder en même temps aux ressources de l'application.
4. Collection d'informations sur les gestionnaires de file d'attente qui sont membres d'un cluster. Ces informations comprennent les noms des gestionnaires de files d'attente, leurs emplacements, leurs canaux, les files d'attente qu'ils hébergent, etc.

## S

**Script de shell :** Fichier contenant des commandes qui peuvent être interprétées par l'interpréteur de commandes. Pour que le shell exécute les commandes du script, l'utilisateur doit saisir le nom du fichier de script à l'invite de commande du shell.

**Section de liaison :** Section de la division des données d'une unité activée (programme ou méthode appelé(e)) qui décrit les éléments de données disponibles à partir d'une unité d'activation (programme ou méthode). L'unité activée et l'unité d'activation peuvent toutes deux se référer à ces éléments de données.

### Serveur d'applications :

1. Programme qui traite toutes les opérations d'une application qui s'exécutent entre des ordinateurs dotés d'un navigateur et les applications dorsales ou bases de données de l'entreprise. Une classe spécifique de serveurs d'applications Java prend en charge la norme J2EE. Le code J2EE peut être facilement porté entre ces différents serveurs. Ils peuvent supporter des JSP et des servlets destinés

au contenu Web dynamique et des EJB pour les transactions et l'accès aux bases de données.

2. Cible d'une demande émise à partir d'une application distante. Dans l'environnement DB2, la fonction du serveur d'applications est fournie par la fonction de données réparties et permet d'accéder aux données DB2 à partir d'applications distantes.
3. Programme serveur dans un réseau réparti qui fournit l'environnement d'exécution d'un programme d'application.
4. Cible d'une demande émise par un demandeur d'application. Le système de gestion de base de données du site du serveur de l'application fournit les données demandées.
5. Logiciel qui traite les communications avec le client lorsque celui-ci demande un actif et les requêtes du Content Manager.

**Session de débogage :** Tâches de débogage exécutées entre l'heure à laquelle le développeur lance le débogage, et l'heure à laquelle il sort de l'application.

**Sidedeck :** Bibliothèque qui publie les fonctions d'un programme DLL. Les noms des entrées et des modules sont stockés dans la bibliothèque après la compilation du code source.

**Système de fichiers distant :** Système de fichiers résidant sur un serveur ou un système d'exploitation séparé.

**Système distant :** Tout autre système du réseau avec lequel votre système peut communiquer.

## T

**Transaction de génération :** Travail démarré sur MVS pour effectuer des générations après qu'une demande de génération a été reçue du client.

## U

**URL :** Uniform Resource Locator

## V

**Vue Console de sortie :** Affiche les données de sortie d'un processus et vous permet d'entrer à partir du clavier les données d'un processus.

**Vue Définition de données :** Affiche une image locale des bases de données, ainsi que des objets qu'elles contiennent. Elle fournit également les fonctions nécessaires pour manipuler ces objets et les exporter vers une base de données distante.

**Vue de sortie :** Affiche les messages, les paramètres et résultats associés aux objets sur lesquels vous travaillez.

**Vue du navigateur :** Fournit une vue hiérarchique des ressources du plan de travail.

**Vue Référentiels :** Affiche les emplacements des référentiels CVS qui ont été ajoutés à votre plan de travail.

**Vue Serveurs :** Présente une liste de tous les serveurs et des configurations qui leur sont associées.

---

## Remarques relatives à la documentation pour IBM Rational Developer for System z

© Copyright IBM Corporation - 2010

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

IBM EMEA Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
France

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues auprès du IBM Intellectual Property Department de votre pays ou par écrit à l'adresse suivante :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni à aucun pays dans lequel il serait contraire aux lois locales : LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

Intellectual Property Dept. for Rational Software  
IBM Corporation  
3039 Cornwallis Road, PO Box 12195  
Research Triangle Park, NC 27709  
France

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans cette documentation et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'IBM Customer Agreement, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

---

## Licence de copyright

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Les exemples de programme sont fournis "EN L'ETAT", sans garantie d'aucune sorte. IBM ne peut en aucun cas être tenu pour responsable des dommages liés à l'utilisation de ces exemples de programme.

---

## Marques

IBM, le logo IBM et [ibm.com](http://www.ibm.com) sont des marques d'International Business Machines Corp. dans la plupart des juridictions du monde. Les autres noms de produit et service sont des marques d'IBM ou d'autres sociétés. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark information" à <http://www.ibm.com/legal/copytrade.shtml>.

Rational est une marque d'International Business Machines Corporation et de Rational Software Corporation aux Etats-Unis et/ou dans certains autres pays.

Intel® et Pentium® sont des marques d'Intel Corporation aux Etats-Unis et/ou dans certains autres pays.

Microsoft®, Windows® et le logo Windows sont des marques ou des marques déposées de Microsoft Corporation au Etats-Unis et/ou dans certains autres pays.

Java et toutes les marques et logos incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group aux Etats-Unis et/ou dans certains autres pays.





---

# Index

## Caractères spéciaux

.dstoreMemLogging 140  
.dstoreTrace 140  
/etc/inittab, initialisation z/OS UNIX, INETD 327  
/etc/rc, démarrage INETD z/OS UNIX 327  
\_RSE\_CMDSERV\_OPTS, Définition de paramètres de démarrage Java supplémentaires 47  
\_RSE\_JAVAOPTS, définition de paramètres de démarrage Java supplémentaires 42  
\_RSE\_PORTRANGE 163

## A

accès aux bibliothèques du système, Amélioration 250  
accès aux fichiers spoule, Conditionnel 171  
Accès conditionnel aux fichiers spoule 171  
Actions à distance basée sur l'hôte, sous-projets z/OS UNIX 105  
Actions conditionnelles sur les travaux 168  
Actions sur les travaux - Limitations liées à l'exécution 170  
Activation de Common Access Repository Manager 51  
Activation des modèles Repository Access Manager (RAM) 62  
Activation du Repository Access Manager du gestionnaire de configuration logicielle de CA Endevor 64  
Activation du Repository Access Manager du squelette 63  
Activation du Repository Access Manager PDS 63  
Activation du Repository Access Manager SCLM 63  
activer le partage de classe, machines virtuelles Java (JVM) 253  
ADM, personnalisation 75  
administrateur, SCLMDT 89  
administrateur non-système, droits de mise à jour 203  
administrateur SCLMDT 89  
ADNCSDAR, RESTful non primaire et Web Service non primaire 78  
ADNCSDRS, RESTful primaire 77  
ADNCSDTX, ID modification RESTful 78  
ADNCSDWS, service Web primaire 80  
ADNJSPAU, utilitaire d'administration 261

ADNJSPAU, utilitaire d'administration CICS 77  
ADNMSGHC, gestionnaire de message de pipeline 79  
ADNMSGHS, gestionnaire de message de pipeline 79  
ADNTXNC, ID modification RESTful 78  
ADNVCRD, référentiel de CRD 76  
ADNVMFST, référentiel de manifeste 81  
adresse hôte non résolue, programme de résolution TCP/IP lock.log 321  
Ajustement d'ISPF.conf 61  
Ajustement de CRASERV.properties 60  
Amélioration de l'accès aux bibliothèques du système 250  
Amélioration des performances du contrôle d'autorisations d'accès 251  
analyse d'utilisation, espace de stockage 228  
Ant  
    éléments corequis 349  
Ant, Installation et personnalisation 88  
Apache Ant  
    éléments corequis 349  
APF, autorisation 152  
APPC, Configuration 331  
APPCPMxx 333  
applications prises en charge 4  
ASCHPMxx 334  
    MAX 240  
ASSIZEMAX 185  
attribut du système de fichiers, SETUID 150  
audit.log 141  
authentification, configuration SSL et X.509 299  
authentification, moniteur de travaux JES 162  
Authentification du client à l'aide de certificats X.509 173  
authentification du moniteur de travaux JES 162  
authentification par le démon RSE 175  
Authentification par logiciel de sécurité 174  
Autorisation APF FEK.SFEKAUTH 182  
Autorisation de contrôle de programmes 151

## B

base de données de clés, Création avec gskkyman 308  
Bibliothèques, Amélioration de l'accès au système 250  
bibliothèques, exécution Language Environment 250  
bibliothèques contrôlées par programme MVS pour RSE, Définition 188

bibliothèques contrôlées pour RSE, Définition par programme MVS 188  
bibliothèques d'exécution, Language Environment 250  
Bibliothèques d'exécution Language Environment 250  
bibliothèques du système, Amélioration de l'accès à 250  
bibliothèques pour RSE, définition par programme MVS 188  
bit sticky, disponibilité du module de chargement sur z/OS UNIX 153  
BPXBATCH, démarrage INETD 328  
BPXPRMxx 245  
    INADDRANYCOUNT 238  
    MAXASSIZE 155, 185, 237  
    MAXFILEPROC 237  
    MAXMMAPAREA 237  
    MAXPROCSYS 158, 236  
    MAXPROCUSER 158, 236  
    MAXSOCKETS 238  
    MAXTHREADS 236  
    MAXTHREADTASKS 236  
    MAXUIDS 158, 237  
BPXPRMxx, définition des limites dans  
    MAXASSIZE 16  
    MAXPROCUSER 16  
    MAXTHREADS 16  
    MAXTHREADTASKS 16

## C

CA Endevor RAM SCM 64  
CA Endevor SCM RAM - démarrage, CRASTART 68  
CA Endevor SCM RAM - démarrage, soumission par lots 65  
CA Endevor SCM RAM, personnalisation 70  
caractères, non alphanumériques dans un diagramme de syntaxe 137  
caractères non alphanumériques, diagramme de syntaxe 137  
Caractères non éditables, personnalisation à gérer 103  
Caractères non éditables, uchars.settings 103  
CARMA, activation 51  
CARMA, FMID HCMA710 296  
CARMA, interface RSE avec 53  
CARMA, traçage 149  
CARMA et ports TCP/IP 166  
CEE.SCEELPA  
    SYS1.PARMLIB(LPALSTxx) 250  
certificat, X.509 162  
certificat signé, auto-signé ou signé par l'autorité de certification 302  
certificat X.509 162  
certificats, authentification du client via X.509 173

- certificats X.509, authentification du client 173
- changements dans les fichiers de configuration, autres produits 283
- chiffrement, ssl.properties 95
- chiffrement à l'aide de la couche SSL, Communication 163
- Chiffrement des communications à l'aide de la couche SSL 163
- chiffrement RSE SSL, ssl.properties, propriétés du démon 95
- propriétés du serveur 95
- choix de l'emplacement de stockage des clés privées et des certificats 300
- CICS Transaction Server
  - éléments corequis 345
- CICSplex SM Business Application Services (BAS) 258
- CICSTS, sécurité 177
- classification des charges de travail, WLM 205
- CLASSPATH 278
- clés privées et certificats, choix de l'emplacement de stockage 300
- clonage de la configuration RSE existante 303
- COBOL
  - vérification à distance 149
- coexistence, mise à jour de rsed.envvars pour activer la coexistence 303
- commande exec d'allocation (JCL de transaction APPC) 273
- commande exec d'allocation, utilisation 271
- commande Modify (F) 129
- commande Start (S) 127
- commande Stop (P) 133
- commandes, démarrer le démon 128
- commandes, Démarrer le moniteur de travaux JES 127
- commandes, modifier le démon RSE 130
- commandes, Modifier le moniteur de travaux JES 129
- commandes, Modify (F) 129
- commandes, Modify du démon lock 133
- commandes, opérateur 127
- commandes, Start (S) 127
- commandes, Start du démon lock 128
- commandes, Stop (P) 133
- Commandes de l'opérateur 127
- Commandes ISPF 48, 61
- Commandes TSO 61
- COMMNDxx, ajout de tâches démarrées 18
- Common Access Repository Manager, Activation 51
- Common Access Repository Manager (CARMA), FMID HCMA710 296
- communication, Externe 164
- communication, Interne 165
- communication, via SSL 172, 260
- communication chiffrée, SSL 172, 177, 260
- communication chiffrée via SSL 172, 177, 260
- Communication interne< 165
- Communications externes 164
- communications externes à des ports spécifiques, limitations 163
- compilateur COBOL
  - éléments corequis 344
- compilateur PL/I
  - éléments corequis 344
- composants, Facultatifs 49
- composants, Installation et configuration 3
- Composants facultatifs 49
- composants hôte, Installation et configuration 3
- compréhension de Developer for System z 193
- Conditions requises, Developer for System z 339
- configuration, identique par sysplex 277
- configuration, personnalisée 15
- configuration APPC sécurisée, VTAM 336
- configuration de trace, rsecomm.properties 98
- configuration de trace RSE, rsecomm.properties, 98
- configuration des composants de l'hôte 3
- configuration du moniteur de travaux JES GEN\_CONSOLE\_NAME 170
- Configuration identique par sysplex 277
- configuration nécessaire pour les logiciels et produits requis 9
- configuration personnalisée 15
- configuration REXEC 105
- Configuration RSE, Clonage de l'existant 303
- configuration SSH 105
- configuration TCP/IP, netstat 116
- configurations Developer for System z multiples, utilisation de fichiers ISPF.conf multiples avec 272
- configurations Developer for System z multiples, utilisation de transactions APPC multiples avec 274
- connexion, démon lock 118
- connexion, démon RSE 117
- connexion, moniteur de travaux JES 117
- connexion, REXEC 122
- Connexion au service, Commandes TSO (APPC) 120
- Connexion au service Commandes TSO (APPC) 120
- connexion de la configuration de l'hôte SSL, Test 305
- connexion de la passerelle client TSO/ISPF d'ISPF, vérification 118
- connexion du démon RSE 117
- Connexion du moniteur de travaux JES 117
- connexion refusée 158
- Connexion refusée 158
- connexion SCLMDT, vérification 120
- consignation, démon lock 141
- Consignation, test du programme de vérification de l'installation fekfivpi 145
- consignation dans le journal d'audit, géré par le démon RSE 167
- consignation des messages de l'installation, ressources CICS 259
- consignation des messages de l'installation des ressources, CICS 259
- Consignation des tests, programme de vérification de l'installation fekfivpi 145
- Consignation des tests du programme de vérification de l'installation (IVP) fekfivpi.log 145
- fekfivps.log 145
- Consignation des tests du programme de vérification de l'installation (IVP) fekfivpi 145
- fekfivpi.log 145
- Consignation du démon lock 141
- contrôle, réseau 243
- contrôle d'audit
  - \_RSE\_HOST\_CODEPAGE 167
  - audit.\* options 167
  - daemon.log 167
  - enable.audit.log 167
- contrôle d'autorisations d'accès, Amélioration des performances de 251
- contrôle de RSE 240
- contrôle de z/OS UNIX 241
- contrôle du système de fichiers z/OS UNIX 243
- conversion de nom, SCLM 86
- conversion de nom long/court, SCLM 86
- conversion de noms longs/court, mises à jour de rsed.envvars pour 88
- corequis, Developer for System z 339
- CRA#ASLM, modèle de RAM SCLM 63
- CRA#CIRX, IRXJCL par opposition à CRAXJCL 72
- CRA#CRAM, modèle de RAM du squelette 63
- CRA#UADD, composants CARMA 52, 65
- CRA#UQRY, composants CARMA 52, 65
- CRA#VCAD, Endevor SCM RAM 65
- CRA#VCAS, Endevor SCM RAM 65
- CRA\$VDEF, composants CARMA 52
- CRA\$VMSG, composants CARMA 52, 65
- CRA#VPDS, modèle de RAM PDS 63
- CRA#VSLM, modèle de RAM SCLM 63
- CRA\$VSTR, composants CARMA 52
- CRAISPRX, fichier de configuration ISPF 47
- CRAISPRX, ISPF alternative CARMA startup 61
- CRANDVRA, personnalisation 69
- CRASERV.properties, Ajustement 60
- CRASRV.properties
  - clist.dsname 54
  - crastart.configuration.file 54
  - crastart.steplib 54
  - crastart.stub 54
  - crastart.syslog 54
  - crastart.tasklib 54
  - crastart.timeout 54
  - port.range 54
  - port.start 54

CRASRV.properties (*suite*)  
 startup.script.name 54  
 CRASRV.properties, ajustement 56, 58, 65, 68  
 CRASTART, CA Endevor SCM RAM -  
 démarrage à l'aide de 68  
 CRASTART, démarrage du serveur  
 CARMA alternatif à l'aide de 57  
 crastart.conf, ajustement 58  
 crastart.conf, démarrage de CARMA  
 crastart 58  
 crastart.endevor.conf, ajustement 69  
 CRASUBCA, ajustement 66  
 CRASUBMT, démarrage par lots  
 CARMA 56  
 CRAXJCL, IRXJCL par opposition à  
 CRAXJCL 72  
 CRD, sécurité du référentiel 259  
 création de LSTRANS.FILE 86  
 cron, nettoyage du répertoire  
 WORKAREA 110

## D

DB2  
 éléments corequis 347  
 Debug Tool  
 éléments corequis 345  
 Debug Tool, IBM, élément corequis 20  
 Définition de fichiers contrôlés par  
 programme z/OS UNIX pour RSE 190  
 Définition de la prise en charge de  
 PassTicket pour RSE 189  
 Définition de la vérification du port  
 d'entrée pour RSE 176  
 définition de ressource, différentes 239  
 Définition des bibliothèques contrôlées  
 par un programme MVS pour RSE 188  
 définition des objectifs, WLM 207  
 Définition du serveur RSE en tant que  
 serveur z/OS UNIX sécurisé 188  
 définitions, prérequis LINKLIST et  
 LPA 20  
 définitions, sécurité 26, 179  
 définitions de port, PROFILE.TCPIP 326  
 définitions de ressource CICS,  
 administrateur 257  
 définitions de ressource CICS,  
 développeur de logiciel 257  
 définitions de ressource différentes 239  
 carte EXEC, JCL serveur 239  
 FEJJCNGF 239  
 SYS1.PARMLIB(ASCHPMxx) 240  
 SYS1.PARMLIB(IEASYSxx) 239  
 SYS1.PARMLIB(IVTPRMxx) 239  
 définitions de ressources essentielles 235  
 rsed.envvars 235  
 SYS1.PARMLIB(BPXPRMxx) 236  
 définitions de sécurité 26, 179  
 définitions de sécurité, Liste de  
 contrôle 180  
 définitions disponibles pour le  
 programme de résolution 321  
 définitions LINKLIST, prérequis 20  
 Définitions locales disponibles pour le  
 programme de résolution 321  
 définitions LPA, prérequis 20  
 démarrage, INETD z/OS UNIX 327  
 démarrage, serveur CARMA  
 alternatif 60  
 démarrage CARMA, soumission par  
 lot 56  
 démarrage du serveur CARMA,  
 alternatif 52, 57, 60  
 Démarrage du serveur CARMA  
 alternatif 60  
 démarrage rapide, option Java  
 (-Xquickstart) 252  
 démon, Lock 23  
 démon, RSE 22  
 démon lock  
 messages de console 134  
 Démon lock 23, 200  
 démon lock, commande Modify 133  
 démon lock, connexion 118  
 démon Lock, LOCKD 111  
 démon Lock (LOCKD) 193  
 démon lock, traçage 149  
 démon REXEC, autorisations des ID  
 utilisateur avec un démarrage par  
 INETD 330  
 Démon RSE 10, 22, 164  
 messages de console 134  
 démon RSE, authentification par 175  
 démon RSE, commande Modify 130  
 démon RSE, commande Start 128  
 démon RSE, RSED 111  
 démon RSE (RSED) 193  
 démon RSE et consignment dans le  
 journal d'audit 167  
 démon SSH, autorisations des ID  
 utilisateur avec un démarrage par  
 INETD 330  
 démond lock, commande Start 128  
 dépendance, Nom d'hôte 315  
 Dépendance au nom d'hôte 315  
 Developer for System z,  
 compréhension 193  
 Developer for System z, présentation du  
 composant  
 représentation graphique 193  
 développement, applications 250  
 Développement d'applications 250  
 diagramme de syntaxe, caractères non  
 alphanumériques et espaces vides 137  
 diagramme de syntaxe, comment  
 lire 136  
 diagramme de syntaxe, de plusieurs  
 lignes 137  
 diagramme de syntaxe, opérandes 136  
 diagramme de syntaxe, sélection de  
 plusieurs opérandes 137  
 diagramme de syntaxe, symboles 136  
 différents fichiers de configuration avec  
 des niveaux de logiciels identiques 278  
 disponibilité, port 115  
 Disponibilité des ports 115  
 données d'audit  
 actions consignées 168  
 données de droits, UNIX z/OS 150  
 droits de mise à jour, administrateur  
 non-système 203

## E

échec de fichiers, Ouverture MVS 158  
 Echec de l'ouverture de fichiers  
 MVS 158  
 éditeur de définition de ressources CICS  
 (CRD), gestionnaire de déploiement  
 d'application 257  
 ELAXMJCL, modifications de DB2 92  
 ELAXMSAM 92  
 éléments corequis  
 Ant 349  
 Apache Ant 349  
 CICS Transaction Server 345  
 compilateur COBOL 344  
 compilateur PL/I 344  
 DB2 347  
 Debug Tool 345  
 Fault Analyzer 348  
 File Manager 348  
 hôte 341  
 IMS 346  
 Java 2 Technology Edition 341  
 Ported tools 349  
 Rational Team Concert for System  
 z 347  
 REXX 348  
 SDK for z/OS 341  
 z/OS 342  
 éléments corequis z/OS  
 Assembleur de haut niveau 343  
 C/C++ 343  
 Language Environment 344  
 SCLM 343  
 éléments prérequis  
 hôte 339  
 SMP/E 341  
 emplacements des fichiers de vidage,  
 z/OS UNIX 147  
 Emplacements des fichiers de vidage  
 UNIX 147  
 Emplacements des fichiers de vidage  
 z/OS UNIX 147  
 emploi de STEPLIB, Eviter 249  
 Emulateur, Connexion à l'hôte 159  
 Emulateur de connexion à l'hôte 159  
 environnement TSO,  
 Personnalisation 269  
 environnement UNIX, Ordres de  
 recherche dans 317  
 environnement UNIX z/OS, Ordres de  
 recherche dans 317  
 espace adresse, limite de taille 227  
 espace de stockage, analyse  
 d'utilisation 228  
 espace disque, machines virtuelles Java  
 (JVM) 254  
 espaces, diagramme de syntaxe 137  
 espaces vides, diagramme de  
 syntaxe 137  
 ETC.SERVICES  
 alias 324  
 nom\_service 324  
 numéro\_port 324  
 protocole 324  
 Exécution à distance, utilisation 104  
 Exécution de plusieurs instances 277  
 exemple de configuration 244

- exemple de configuration (*suite*)
  - compte de pools d'unités d'exécution 244
  - définition des limites 245
  - détermination des limites minimales 244
- exigences, langage JCL de démarrage 155
- exigences d'accès 9
- exigences liées au langage JCL, démarrage 155
- exigences liées au langage JCL de démarrage 155

## F

- fa.log 140
- Fault Analyzer
  - éléments corequis 348
- FEJJCENFG 165, 245, 280
  - CONSOLE\_NAME 170
  - MAX\_THREADS 239
- FEJJCENFG, fichier de configuration du moniteur de travaux JES 27
- FEJJCENFG, moniteur de travaux JES 178
- FEJJJCL, modifications de PROCLIB changes, moniteur de travaux JES 21
- FEJTSO 27
- FEKAPPC, implémentation APPC 108
- FEKAPPCL, implémentation APPC 108
- FEKAPPCX, implémentation APPC 108
- FEKAPPL 162
- fekfivpi.log 140
- fekfivpi.log, consignation des tests du programme de vérification d'installation 145
- fekfivps.log, consignation des tests du programme de vérification d'installation 145
- FEKFRSRV 109
- FEKLOCKD 23
- FEKLOGS, journal et analyse de configuration 139
- FEKRACF, définitions de sécurité 26, 179
- fekrivp 152
- FEKRSED 22
- FEKSETUP 15, 78, 84, 108
- ffs.log 140
- ffsget.log 140
- ffsput.log 140
- fichiers MVS, échec à l'ouverture 158
- fichier de clés, Création avec RACF 301
- fichier de clés avec keytool, Création 311
- Fichier de configuration de consignation, rsecomm.properties 98
- fichier de configuration de la passerelle client, TSO/ISPF 47
- Fichier de configuration de la passerelle client TSO/ISPF 47
- fichier de configuration du moniteur de travaux JES, FEJJCENFG, 27
- fichier de configuration RSE, rsed.envvars, 32
- fichier temporaire INETD /etc/inetd.pid 327

- fichiers configurables 288, 297
- fichiers configurés précédemment, sauvegarde
  - Version 7.0 283
  - Version 7.1 283
  - Version 7.5 283
- fichiers contrôlés par programme UNIX pour RSE, Définition 190
- fichiers contrôlés par programme z/OS UNIX pour RSE, Définition 190
- fichiers de configuration, Developer for System z 178
- fichiers de configuration, Niveau de logiciels identique, différent 278
- fichiers de configuration, Programme de résolution de base 317
- Fichiers de configuration du programme de résolution de base 317
- Fichiers de vidage 145
- fichiers de vidage, Java 146
- fichiers de vidage, MVS 145
- Fichiers de vidage Java 146
- Fichiers de vidage MVS 145
- fichiers ISPF.conf, utilisation multiple avec configurations 272
- Fichiers ISPF.conf multiples 272
- fichiers journaux
  - .dstoreMemLogging 140
  - .dstoreTrace 140
  - audit.log 140
  - fa.log 140
  - fekfivpi.log 140
  - fekfivps.log 140
  - ffs.log 140
  - ffsget.log 140
  - ffsput.log 140
  - lock.log 140
  - rmt\_class\_loader.cache.jar 140
  - rsecomm.log 140
  - rsedaemon.log 140
  - rseserver.log 140
  - serverlogs.count 140
  - stderr.log 140
  - stdout.log 140
- fichiers journaux du démon RSE
  - audit.log 142
  - rsedaemon.log 142
  - rseserver.log 142
  - serverlogs.count 142
  - stderr.\*.log 142
  - stdout.\*.log 142
- fichiers journaux du pool d'unités d'exécution RSE
  - audit.log 142
  - rsedaemon.log 142
  - rseserver.log 142
  - serverlogs.count 142
  - stderr.\*.log 142
  - stdout.\*.log 142
- fichiers spoule, Accès conditionnel aux 171
- File Manager
  - éléments corequis 348
- flux de connexion 198
- représentation graphique 198
- Flux du démon lock
  - représentation graphique 200

- FMID HCMA710 296
- FMID HHOP710 296
- FMID HHOP750 292
- FMID HHOP760 286
- FMIEXT.properties
  - enabled 103
  - fmlistenport 103
- fonction de trace, moniteur de travaux JES 148
- fonction de trace du moniteur de travaux JES 148
- fragments, Syntaxe 137
- Fragments de syntaxe 137

## G

- gestion, Charge de travail 251
- Gestion de la charge de travail 251
- Gestionnaire, Message de pipeline 79
- Gestionnaire de configuration logicielle (SCM pour Software Configuration Manager) 64
- gestionnaire de déploiement d'application (ADM) 193
- gestionnaire de déploiement d'application, éditeur de définition de ressource CICS 257
- gestionnaire de déploiement d'application, personnalisation 257
- Gestionnaire de déploiement d'application, personnalisation 75
- gestionnaire de déploiement d'application, sécurité 259
- gestionnaire de déploiement d'application, serveur de définition de ressource CICS 257
- Gestionnaire de message, Pipeline 79
- Gestionnaire de message de pipeline 79
- groupes de propriétés, basés sur l'hôte 100
- groupes de propriétés basés sur l'hôte 100
- gskkyman, Création d'une base de données de clés avec 308

## H

- HCMA710 296
- HHOP710 296
- HHOP750 292
- HHOP760 286
- hôte
  - éléments prérequis 339
- hôte, éléments corequis 341
- hôte, éléments prérequis 339
- hôtez/OS
  - éléments prérequis 339

## I

- IBM Debug Tool, élément corequis 20
- ID de transaction du serveur CRD, personnalisation 78
- ID utilisateur et mot de passe utilisable une seule fois 162



- Identification des incidents liés à la configuration 139
- IEASYSxx 246
  - MAXUSER 158, 239
- IMS
  - éléments corequis 346
- incidents liés à la configuration,
  - Identification 139
- incidents relatifs aux éléments requis, connus 158
- INETD, configuration 323
- INETD, exigences Developer for System z 330
- INETD, paramètres
  - environnementaux 330
- INETD, sécurité 329
- inetd.conf 323
  - nom\_service 323
  - protocole 323
  - server\_program 323
  - server\_program\_arguments 323
  - socket\_type 323
  - userid 323
  - wait\_flag 323
- informations de configuration, ordres de recherche de 316
- initialisation IVP
  - ivpinit 114
- Installation et configuration des composants de l'hôte 3
- intégration, gestionnaire de fichiers 102
- Intégration du gestionnaire de fichiers 102
- interface, RESTful par opposition au service Web 77
- interface de service Web 258
  - ADMI 79
  - ADMR 79
  - ADMS 79
- interface de service Web, serveur CRD
  - utilisation de 79
- interface de service Web par opposition à l'interface RESTful 77
- interface RESTful 258
  - ADMI 78
  - ADMR 78
  - ADMS 78
- interface RESTful, serveur CRD
  - utilisation de 77
- interface RESTful par opposition à l'interface de service Web 77, 258
- interface RSE avec CARMA 53
- Interpréteur de commandes sécurisé (SSH pour Secure Shell), utilisation 104
- interrogation d'une liste de révocation de certificat (CRL)
  - rsed.envvars 174
  - variables d'environnement CRL 174
- IRXJCL, IRXJCL par opposition à CRAXJCL 72
- ISP.SISPLOAD
  - ISPF TSO/ISPF, passerelle client 188
- ISPF, Utilisation de plusieurs commandes
  - exec d'allocation 271
- ISPF.conf 47
  - allocjob 48
  - ISPF\_timeout 48

- ISPF.conf (*suite*)
  - isplib 48
  - isplib 48
  - isplib 48
  - isplib 48
  - isplib 48
  - sysproc 48
- ISPF.conf, Ajustement 61
- ISPF.conf, mises à jour pour
  - SCLMDT 84
- ISPF.conf, Personnalisation de base 270
- ISPF TSO/ISPF, passerelle client
  - ISP.SISPLOAD 188
- IVP
  - fekfivpa 120
  - fekfivpd 117
  - fekfivpi 118
  - fekfivpj 117
  - fekfivpl 118
  - fekfivpr 122
  - fekfivps 120
  - fekfivpz 123
- IVP, initialisation 114
- IVP, services de base et facultatifs
  - fekfivpa 113
  - fekfivpd 113
  - fekfivpi 113
  - fekfivpj 113
  - fekfivpl 113
  - fekfivpr 113
  - fekfivps 113
  - fekfivpt 113
  - fekfivpz 113
  - scripts IVP 113
- IVTPRMxx
  - ECSA MAX 239
  - FIXED MAX 239

## J

- J2EE 83, 88
- Java 83, 88
- Java, limite de taille de pile 227
- Java 2 Technology Edition
  - éléments corequis 341
- JAVA\_DUMP\_TDUMP\_PATTERN 146
- JCL de transaction APPC,
  - Personnalisation de base 272
- JCL de transaction APPC, Utilisation
  - d'une commande exec d'allocation 273
- JCL de transaction APPC, Utilisation des
  - profils ISPF existants avec 273
- JES JMON
  - GEN\_CONSOLE\_NAME 170
- JES JMON, FEJJCENFG
  - \_BPXK\_SETIBMOPT
    - \_TRANSPORT 28
  - APPLID 28
  - AUTHMETHOD 28
  - CODEPAGE 28
  - CONCHAR 28
  - CONSOLE\_NAME 28
  - GEN\_CONSOLE\_NAME 28
  - HOST\_CODEPAGE 28
  - LIMIT\_COMMANDS 28
  - LIMIT\_VIEW 28
  - LISTEN\_QUEUE\_LENGTH 28

- JES JMON, FEJJCENFG (*suite*)
  - MAX\_DATASETS 28
  - MAX\_THREADS 28
  - SERV\_PORT 28
  - SUBMITMETHOD 28
  - TIMEOUT 28
  - TIMEOUT\_INTERVAL 28
  - TSO\_TEMPLATE 28
  - TZ 28
- JMON 21, 186, 280
  - fekfivpj 117
- JMON, moniteur de travaux JES 111
- Journal d'intégration de Fault Analyzer
  - fa.log 143
  - rsecomm.log 143
- Journal d'intégration de File Manager
  - rsecomm.log 143
- journal du service Commandes TSO 144
- Journal et analyse de configuration à l'aide de FEKLOGS 139
- Journalisation, CARMA 144
- journalisation, démon RSE 141
- Journalisation, intégration de Fault Analyzerlogging 143
- Journalisation, intégration de File Manager 143
- journalisation, moniteur de travaux JES 141
- journalisation, pool d'unités
  - d'exécution 141
- journalisation, SCLM Developer Toolkit 144
- journalisation, transaction APPC 144
- Journalisation, utilisateur RSE 142
- Journalisation CARMA
  - rsecomm.log 144
- Journalisation de la transaction
  - APPC 144
- journalisation des ressources d'installation
  - CICS 259
- Journalisation du moniteur de travaux JES 141
- journalisation du pool d'unités
  - d'exécution 141
- Journalisation pour l'utilisateur, RSE 142
- Journalisation pour l'utilisateur RSE
  - .dstoreMemLogging 142
  - .dstoreTrace 142
  - ffs.log 142
  - ffsget.log 142
  - ffsput.log 142
  - lock.log 142
  - rmt\_class\_loader.cache.jar 142
  - rsecomm.log 142
  - stderr.log 142
  - stdout.log 142
- Journalisation pour le Common Access Repository Manager 144
- Journalisation pour le démon RSE 141
- JVM, partage de classes entre 252

## K

- keytool, Création d'un fichier de clés
  - avec 311

## L

- L'Explorateur de systèmes distants 11
- libération d'un verrou
  - RSE , commande d'annulation de modification 201
- LIMIT\_COMMANDS 169
- LIMIT\_VIEW 171
- limitation de communication externe, ports spécifiés 163
- limitations d'exécution, actions sur les travaux 170
- limite de taille, espace adresse 227
- limite de taille de pile, Java 227
- limite de taille du cache, machines virtuelles Java (JVM) 253
- limites, Système 158
- limites, z/OS UNIX définies dans BPXPRMxx 16
- Limites du système 158
- limites z/OS UNIX définies dans BPXPRMxx 16
- LINKLIST, définitions pour d'autres produits 21
- lire un diagramme de syntaxe, comment 136
- liste de contrôle, administrateur SCLM 90
- liste de contrôle, client 13
- liste de contrôle, configuration requise 15
- liste de contrôle, transaction APPC 107
- Liste de contrôle de l'administrateur SCLM 90
- liste de contrôle de la configuration requise 15
- liste de contrôle de transaction, APPC 107
- Liste de contrôle de transaction APPC 107
- liste de contrôle des qualificatifs, ELAXF\* 26
- Liste de contrôle des qualificatifs de haut niveau ELAXF\* 26
- Liste de contrôle du client 13
- liste de révocation de certificat (CRL), interrogation
  - rsed.envvars 174
  - variables d'environnement CRL 174
- lock.log 140
- LOCKD 10
- LOCKD, démon Lock 111
- logiciel de sécurité, authentification par 174
- logiciels, configuration nécessaire pour les produits et 9
- logiciels et produits requis, configuration de 9
- LPALSTxx 250
- LPALSTxx, définitions LPA dans 18
- LSTRANS.FILE, création 86

## M

- machines virtuelles Java (JVM), partage de classes entre 252
- messages, console 134

- messages, utilitaire d'administration 266
- Messages d'erreur de diagnostic IRZ 95
- messages d'erreurs, diagnostic IRZ 95
- messages de console 134
  - Moniteur de travaux JES 134
- Messages de console
  - démon lock 134
  - Démon RSE 134
  - Serveur de pool d'unités d'exécution RSE 134
- messages de l'utilitaire d'administration 266
- méthode d'accès, Utilisation de la passerelle client TSO/ISPF 270
- méthode d'accès APPC, utilisation 272
- méthode d'accès par passerelle client, Utilisation de TSO/ISPF 270
- méthode d'accès par passerelle client TSO/ISPF, Utilisation 270
- méthode d'accès virtuelle en télétraitement 332
- méthode de connexion RSE alternative 105
- Méthodes, Authentification 161
- Méthodes d'accès, TSO 269
- Méthodes d'accès TSO 269
- Méthodes d'authentification 161
- migration 283
- migration, version 7.5 vers 7.6 286
- mis à jour SCLM pour SCLMDT 89
- mode de passe utilisable une seule fois et ID utilisateur 162
- modifications, DB2 92
- modifications, PROCLIB 92
- modifications, Workload Manager 91
- modifications d'APPC, Activation 335
- modifications de DB2 92
- modifications de PROCLIB 92
- modifications de Workload Manager 91
- Modifications entre la version 7.0 et la version 7.1 296
- Moniteur de travaux JES, commande Modify 129
- Moniteur de travaux JES, commande Start 127
- moniteur de travaux JES, FEJCNFG 178
- moniteur de travaux JES (JMON) 193
- mots de passe et ID utilisateur 162

## N

- netstat 153
- netstat, configuration TCP/IP 116
- nettoyage du répertoire, WORKAREA skulker 110
- nettoyage du répertoire WORKAREA skulker 110
- niveau de logiciels, identique dans des fichiers de configuration différents 278
- niveaux de logiciels identiques avec des fichiers de configuration différents 278
- nom d'unité logique partenaire, service Commandes TSO 336
- nom de transaction, alternatif FEKFRSRV pour le service Commandes TSO 336
- Nombre d'espaces adresses 217
- Nombre d'unités d'exécution 223

- Nombre de processus 220
- noms d'hôte, application à Developer for System z 319
- notes de migration, utilitaire d'administration 265

## O

- objectifs, définition dans WLM 207
- opérande, sélection multiple dans un diagramme de syntaxe 137
- Opérandes, diagramme de syntaxe 136
- option Java Xquickstart 252
- options de configuration alternatives, APPC et VTAM 336
- options de configuration alternatives APPC 336
- options de configuration alternatives VTAM 336
- ordre de recherche, environnement MVS natif 325
- ordre de recherche, environnement z/OS UNIX 325
- Ordres de recherche, environnement UNIX z/OS 317
- ordres de recherche d'informations de configuration 316

## P

- paramètres de démarrage avec \_RSE\_CMDSERV\_OPTS, Définition supplémentaire Java 47
- paramètres de démarrage avec \_RSE\_JVAOPTS, Définition supplémentaire Java 42
- paramètres de démarrage Java avec \_RSE\_CMDSERV\_OPTS, Définition 47
- paramètres de démarrage Java avec \_RSE\_JVAOPTS, définition 42
- paramètres de sécurité, vérification 191
- paramètres environnementaux, INETD 330
- paramètres et classes, Activation de la sécurité 181
- paramètres et classes de sécurité, Activation 181
- PARMLIB, modifications 16
- partage de classe, activation dans des machines virtuelles Java (JVM) 253
- partage de classes entre machines virtuelles Java (JVM) 252
- PassTickets, utilisation 166
- performances du contrôle d'autorisations d'accès, amélioration 251
- personnalisation - ISPF.conf, 270
- personnalisation, de base 15
- personnalisation, SCLM Developer Toolkit 83
- personnalisation d'ID de transaction du serveur CRD 78
- personnalisation de base 15
- Personnalisation de l'environnement TSO 269
- personnalisation du gestionnaire de déploiement d'application 257

- perspective de projets z/OS 101
- pile Java, limite de taille 227
- pipeline,sécurité 259
- plage de ports disponible pour RSE,
  - Définition 41
- plateformes prises en charge 3
- plusieurs commandes exec d'allocation,
  - TSO/ISPF 271
- plusieurs instances, Exécution 277
- port d'entrée, vérification 164, 176
- Ported tools
  - éléments corequis 349
- PORTRANGE 42, 154
- ports, CARMA et TCP/IP 166
- ports, REXEC 105
- ports, TCP/IP 164
- ports, TCP/IP réservés 153
- ports spécifiques, limitation de
  - communication externe à des 163
- ports TCP/IP 164
- Ports TCP/IP, représentation
  - graphique 164
- Ports TCP/IP, réservés 153
- Ports TCP/IP réservés 153
- préparation, pré-configuration 9
- préparation de la configuration 9
- prérequis, Developer for System z 339
- présentation du composant, Developer
  - for System z
    - représentation graphique 193
- principal par opposition à non-principal,
  - régions de connexion 258
- Prise en charge de la langue
  - bidirectionnelle 94
- Prise en charge de la langue
  - bidirectionnelle CICS 94
- procédure mémorisée, DB2 91
- procédure mémorisée DB2 91
- procédures de construction à distance,
  - ELAXF\* 25
- procédures de construction à distance
  - ELAXF\* 25
- procédures ELAXF\*, Modèle
  - ELAXFADT 25
  - ELAXFASM 25
  - ELAXFBMS 25
  - ELAXFCOC 25
  - ELAXFCOP 25
  - ELAXFCOT 25
  - ELAXFCP1 25
  - ELAXFCPC 25
  - ELAXFCPP 25
  - ELAXFDCL 25
  - ELAXFGO 25
  - ELAXFLNK 25
  - ELAXFMFS 25
  - ELAXFPL1 25
  - ELAXFPLP 25
  - ELAXFPLT 25
  - ELAXFPP1 25
  - ELAXFTSO 25
  - ELAXFUOP 25
- PROCLIB, modifications 21
- produit prérequis 5
- produits, prérequis 5
- produits et logiciels, configuration
  - nécessaire 9

- profil de sécurité, Limitations
  - stockées 155
- PROFILE.TCPIP, définitions de port 326
- profils, Définition de fichier 182
- profils de fichier, Définition 182
- profils ISPF (JCL de transaction
  - APPC) 273
- profils ISPF, Utilisation de l'existant 270
- profils ISPF existants (JCL de transaction
  - APPC), Utilisation 273
- programme de résolution, Définitions
  - locales disponibles pour 321
- programme de résolution TCP/IP,
  - adresse hôte non résolue
    - lock.log 321
- programmes de résolution,
  - Présentation 316
- programmes de vérification de
  - l'installation 113
- PROGxx, autorisations APF 18
- PROGxx, définitions LINKLIST 19
- projectcfg.properties 101
- PROJECT-HOME 102
- WSED-VERSION 102
- projets, basés sur l'hôte 101
- projets basés sur l'hôte 101
- propertiescfg.properties 100
- DEFAULT-VALUES 101
- ENABLED 101
- PROPERTY-GROUP 101
- RDZ-VERSION 101
- propriétaires de tâches 196
- protection, ressources 261
- protection d'application pour RSE,
  - Définition 189
- publications, Référencées 351
- Publications référencées 351

## R

- RACF
  - permis 183
- RACF, Création d'un fichier de clés
  - avec 301
- Rational Team Concert for System z
  - éléments corequis 347
- référentiel, CRD 76
- référentiel, manifeste 81
- référentiel de CRD 76, 177
- référentiel des manifestes 81
- région de connexion CICS
  - non-primaire 81
- région de connexion CICS primaire 77, 80
- région de connexion primaire, CICS 77
- région gérant le Web 258
- régions de connexion, CICS
  - secondaires 78
- régions de connexion, principal par
  - opposition à non-principal 258
- régions de connexion CICS
  - secondaires 78
- régions de connexion secondaires,
  - CICS 78
- règles de classification, WLM 206
- remarques, ID utilisateur 9
- remarques, Performances 249

- remarques, Pré-installation 4
- remarques, prédéploiement 12
- remarques, Sécurité 161
- remarques, serveur 10
- Remarques à propos de CICSTS 257
- remarques à propos de WLM xvi, 205
- remarques relatives à l'ID, Utilisateur 9
- remarques relatives à l'ID utilisateur 9
- remarques relatives à l'optimisation 215
- remarques relatives à l'utilisation
  - d'APPC 109
- remarques relatives à la migration 283
- Remarques relatives à la migration 3
- Remarques relatives à la
  - pré-installation 4
- Remarques relatives à la sécurité< 161
- remarques relatives au
  - prédéploiement 12
- remarques relatives au serveur 10
- Remarques relatives aux
  - performances 249
- Repository Access Manager, activation du
  - SCM de CA Endevor 64
- Repository Access Manager, Activation
  - du squelette 63
- Repository Access Manager, Activation
  - PDS 63
- Repository Access Manager, Activation
  - SCLM 63
- Repository Access Manager (RAM) 64
- Repository Access Manager (RAM),
  - Activation des modèles 62
- Repository Access Manager du squelette,
  - Activation 63
- Repository Access Manager PDS,
  - Activation 63
- Repository Access Manager SCLM,
  - Activation 63
- réseau, contrôle 243
- ressources, protection 261
- ressources, requises 5
- ressources requises 5
- REXEC, utilisation 104
- REXEC, vérification de la connexion 122
- REXX
  - éléments corequis 348
- rmt\_class\_loader\_cache.jar 140
- RSE 11
- RSE , Définition de PORTRANGE 41
- RSE, contrôle 240
- RSE, Définition d'un serveur z/OS UNIX
  - sécurisé 188
- RSE, Définition de fichiers contrôlés par
  - programme z/OS UNIX pour 190
- RSE, définition de la protection
  - d'application pour 189
- RSE, Définition de la vérification du port
  - d'entrée pour 176
- RSE, définition des bibliothèques
  - contrôlées par programme MVS 188
- RSE, Définition du support PassTicket
  - pour 189
- RSE, méthode de connexion
  - alternative 105
- RSE, rsed.envvars
  - \_RSE\_JVAOPTS 178
- RSE, ssl.properties 179

- RSE, traçage 148
- RSE comme application Java
  - représentation graphique 195
- rsecomm.log 140
  - Journal d'intégration de File Manager 143
  - SCLM Developer Toolkit, journalisation 144
- rsecomm.properties 148
  - debug\_level 98
  - log\_location 98
  - server.version 98
- rsecomm.properties, 98
- RSED 22
- RSED, démon RSE 111
- rsed.envvars 130, 234, 278
  - \_BPX\_SHAREAS 40
  - \_BPX\_SPAWN\_SCRIPT 40
  - \_BPXK\_SETIBMOPT
  - \_TRANSPORT 38
  - \_CEE\_DMPTARG 36
  - \_CEE\_RUNOPTS 40
  - \_CMDSERV\_BASE\_HOME 37
  - \_CMDSERV\_CONF\_HOME 37, 272
  - \_CMDSERV\_WORK\_HOME 37
  - \_FEKFSCMD\_PARTNER\_LU\_ 38
  - \_FEKFSCMD\_TP\_NAME\_ 38
  - &ISPROF=&SYSUID..ISPPROF= 47
  - \_RSE\_CMDSERV\_OPTS 40
  - \_RSE\_DAEMON\_CLASS 40
  - \_RSE\_HOST\_CODEPAGE 36
  - \_RSE\_JAVAOPTS 40, 146, 269
  - \_RSE\_LOCKD\_CLASS 40
  - \_RSE\_LOCKD\_PORT 36
  - \_RSE\_POOL\_SERVER\_CLASS 40
  - \_RSE\_PORTRANGE 38, 163
  - \_RSE\_SERVER\_CLASS 40
  - \_RSE\_SERVER\_TIMEOUT 40
  - \_SCLMDT\_TRANTABLE 38
  - ANT\_HOME 38
  - CGI\_DTWORk 40
  - CLASSPATH 40
  - DAPPLID 46
  - Daudit.cycle 44
  - Daudit.retention.period 44
  - Ddaemon.log 42
  - Ddeny.nonzero.port 45
  - DDENY\_PASSWORD 46
  - DDSTORE\_IDLE\_SHUTDOWN
  - \_TIMEOUT 46
  - DDSTORE\_LOG\_DIRECTORY 42
  - DDSTORE\_MEMLOGGING\_ON 46
  - DDSTORE\_TRACING\_ON 46
  - Denable.audit.log 44
  - Denable.automount 44
  - Denable.certificate.mapping 44
  - Denable.port.of.entry 44
  - Denable.standard.log 44
  - DHIDE\_ZOS\_UNIX 46
  - Dipv6 44
  - Dkeep.last.log 44
  - Dmaximum.clients 42, 235
  - Dmaximum.threadpool.process 44, 235
  - Dmaximum.threads 42, 235
  - Dminimum.threadpool.process 42, 235

- rsed.envvars (suite)
  - Dprocess.cleanup.interval 45
  - Dsingle.logon 45
  - DSTORE\_LOG\_DIRECTORY 144, 148
  - DTSO\_SERVER 46
  - Duser.log 42
  - GSK\_CRL\_SECURITY\_LEVEL 38
  - GSK\_LDAP\_PASSWORD 38
  - GSK\_LDAP\_PORT 38
  - GSK\_LDAP\_SERVER 38
  - GSK\_LDAP\_USER 38
  - JAVA\_HOME 36
  - JAVA\_PROPAGATE 40
  - LANG 36
  - LIBPATH 40
  - PATH 36, 40
  - RSE\_HOME 36
  - RSE\_JAVAOPTS 37
  - RSE\_LIB 40
  - RSE\_SAF\_CLASS 37
  - SCLMDT\_BASE\_HOME 40
  - SCLMDT\_CONF\_HOME 38
  - SCLMDT\_WORK\_HOME 40
  - STEPLIB 36, 37, 38, 172
  - TZ 36
  - Xms 42, 235
  - Xmx 42, 235
- rsed.envvars, fichier de configuration
  - RSE 32
- rsed.envvars, mise à jour pour assurer la coexistence 303
- rsed.envvars, mises à jour pour
  - SCLMDT 85
- rsed.envvars, mises à jour pour une conversion de noms longs/courts 88
- rsedaemon.log 140, 141
- rseserver.log 140, 141

## S

- sauvegarde des fichiers configurés
  - Version 7.0 283
  - Version 7.1 283
  - Version 7.5 283
- SCLM 88
- SCLM, Conversion de nom
  - long/court 86
- SCLM, sécurité 177
- SCLM Developer Toolkit 189
- SCLM Developer Toolkit, journalisation
  - rsecomm.log 144
- SCLM Developer Toolkit, personnalisation 83
- SCLM Developer Toolkit (SCLMDT) 193
- SCLM Developer Toolkit, vérification de la connexion
  - SCLMDT, vérifications 120
- SCLMDT, mises à jour ISPF.conf
  - pour 84
- SCLMDT, mises à jour rsed.envvars
  - pour 85
- script de shell, REXEC/SSH 123
- script de shell ivpinit, définition de variables d'environnement avec 114
- script de shell REXEC/SSH 123
- script de shell SSH 123
- SDSF 84

- Secure Socket Layer, Chiffrement des communications à l'aide de 163
- Secure Socket Layer, Configuration 299
- sécurité, CICSTS 177
- sécurité, Connexion 163
- sécurité, Définition des commandes JES 186
- sécurité, gestionnaire de déploiement d'application (ADM) 259
- sécurité, INETD 329
- sécurité, JES 168
- sécurité, pipeline 259
- sécurité, SCLM 177
- sécurité, transaction 259
- sécurité de l'unité d'exécution dans un serveur RSE
  - PassTickets 166
- sécurité des commandes, Définition
  - JES 186
- sécurité des commandes JES, Définition 186
- Sécurité des connexions 163
- sécurité des unités logiques, VTAM 336
- sécurité du cache, machines virtuelles
  - Java (JVM) 253
- sécurité du référentiel, CRD 259
- sécurité JES 168
- segment, Définition OMVS 182
- segment OMVS, Définition 182
- serverlogs.count 140
- serveur CRD utilisant l'interface RESTful 77
- serveur de définition de ressources CICS (CRD), gestionnaire de déploiement d'application 257
- Serveur de pool d'unités d'exécution RSE
  - messages de console 134
- serveur du moniteur de travaux, JES 21
- serveur du moniteur de travaux JES 21
- Serveur RSE 164
- serveur UNIX, Définition de RSE en tant que 188
- serveur z/OS UNIX, Définition de RSE en tant que 188
- serveur z/OS UNIX sécurisé, Définition de RSE en tant que 188
- service Commandes TSO 193
- Service Commandes TSO 156, 269
- service Commandes TSO, transaction
  - APPC pour 106
- Service SCLM Developer Toolkit 19
- session de Shell, démarrage INETD 328
- SETUID, attribut du système de fichiers 150
- skulker, nettoyage du répertoire
  - WORKAREA 110
- SMP/E
  - éléments prérequis 341
- SMP/E - installation, bit sticky 153
- sorties du système, Limitations
  - forcées 155
- soumission par lot, à l'aide du démarrage du serveur CARMA 56
- soumission par lots, CA Endevor SCM
  - RAM - démarrage à l'aide 65
- Sous-projets z/OS UNIX, actions à distance basées sur l'hôte pour 105



- sous-systèmes pris en charge 4
- SSH, utilisation 104
- SSL, Chiffrement des communications à l'aide de 163
- SSL, configuration 299
- ssl.properties 95
  - daemon\_key\_label 97
  - daemon\_keydb\_file 97
  - daemon\_keydb\_password 97
  - enable\_ssl 97
  - server\_keystore\_file 97
  - server\_keystore\_label 97
  - server\_keystore\_password 97
  - server\_keystore\_type 97
- ssl.properties, activation de SSL en créant un démon RSE 304
- ssl.properties, Activation du protocole SSL via la mise à jour 303
- stderr\*.log 140
- stderr.log 140
- stdout\*.log 140
- stdout.log 140
- STEPLIB, Eviter l'emploi de 249
- structure de répertoire, z/OS UNIX
  - représentation graphique 202
- structure de répertoire z/OS UNIX
  - représentation graphique 202
- support, Enterprise Service Tools 93
- support, EST 93
- support d'authentification du client, ajout de X.509 308
- support Enterprise Service Tools 93
- support EST 93
- support PassTicket pour RSE, Définition 189
- support pour RSE, Définition du PassTicket 189
- Suppression des fichiers antérieurs du répertoire WORKAREA 90
- Symboles, diagramme de syntaxe 136
- Syntaxe, exemple 137
- SYS1.PARMLIB(APPCLPMxx) 333
- SYS1.PARMLIB(ASCHPMxx) 334
- SYS1.PARMLIB(BPXPRMxx) 245
  - MAXASSIZE 155, 185
  - MAXPROCSYS 158
  - MAXPROCUSER 158
  - MAXUIDS 158
- SYS1.PARMLIB(BPXPRMxx), Limitations définies 155
- SYS1.PARMLIB(BPXPRMxx), machines virtuelles Java (JVM) 254
- SYS1.PARMLIB(IEASYSxx) 246
  - MAXUSER 158
- SYSEXEC 48
- sysplex, configuration identique par 277
- SYSPROC 48
- système de fichiers zFS, utilisation 249
- systèmes d'exploitation pris en charge 3
- systèmes de fichiers, zFS 249
- systèmes de fichiers/OS UNIX, contrôle 243

## T

- tables, Conversion 318
- tables, Hôte local 318

- Tables de conversion 318
- Tables de système hôte local 318
- tables hôte, Local 318
- tâche démarrée du moniteur de travaux JES, JMON 21
- tâche démarrée RSE, RSED 22
- tâches, personnalisation facultative 91
- tâches de personnalisation, facultatives 91
- tâches de personnalisation alternatives 91
- tâches démarrées, Définition pour Developer for System z
  - tâches démarrées JMON 185
  - tâches démarrées LOCKD 185
  - tâches démarrées RSED 185
- tâches démarrées, vérification 111
- tâches démarrées, vérification des services 113
- tâches démarrées Developer for System z, Définition 185
- tâches démarrées JES, JMON 111
- taille, Espace adresse 155
- Taille d'espace adresse 155
- taille de pile Java, fixe 251
- Taille de pile Java fixe 251
- taille estimée, instructions 228
- TCP/IP, application à Developer for System z 319
- TCP/IP, Configuration 315
- TCP/IP, Définitions locales disponibles pour le programme de résolution 321
- Test de la connexion de la configuration de l'hôte SSL 305
- tiers et certificat X.509 162
- TN3270 344
- traçage 148
- traçage, CARMA 149
- traçage, démon Lock 149
- traçage, RSE 98, 148
- traçage, suivi des erreurs 149
- traçage de suivi, erreur 149
- Traçage de suivi des erreurs 149
- transaction, sécurité 259
- Transaction APPC
  - identification et résolution des incidents APPC 156
- transaction APPC, Implémentation 108
- transaction APPC, préparation 106
- Transaction APPC et service Commandes TSO 156
- Transaction APPC pour le service Commandes TSO 106
- Transaction du service Commandes TSO, Définition 335
- Transactions APPC, configuration multiple Developer for System z 274
- Transactions APPC multiples 274
- transactions CICS 177
- travaux, Actions conditionnelles sur 168
- TSO/ISPF, personnalisation - ISPF.conf, 270
- TSO/ISPF, utilisation avec configurations multiples 272
- TSO/ISPF, Utilisation d'une commande exec d'allocation 271

- TSO/ISPF, Utilisation de plusieurs commandes exec d'allocation 271
- TSO/ISPF, Utilisation des profils ISPF existants 270
- types de sous-système
  - ASCH 206
  - CICS 206
  - JES 206
  - OMVS 206
  - STC 206

## U

- uchars.settings, Caractères non éditables 103
- Un ID utilisateur et un mot de passe 162
- unités logiques multiples 336
- UNIX z/OS, données de droits 150
- utilisation d'APPC, remarques relatives à 109
- Utilisation d'une commande exec d'allocation 271
- utilisation de l'espace, système de fichiers z/OS UNIX 232
- utilisation de l'espace de stockage 226
- utilisation de l'espace du système de fichiers, z/OS UNIX 232
- utilisation de l'espace du système de fichiers z/OS UNIX 232
- utilisation de la soumission par lot pour un démarrage du serveur CARMA 56
- utilisation de PassTickets 166
- Utilisation de STEPLIB 21
- utilisation des profils ISPF existants 270
- utilisation des ressources, optimisation 215
- utilisation des ressources, présentation 216
- Utilitaire d'administration, notes de migration 265
- Utilitaire d'administration CICS 77
- utilitaire d'administration pour des administrateur CICS
  - fonctions fournies 261
- utilitaires de gestion du cache, machines virtuelles Java (JVM) 254

## V

- validation de l'autorité de certification
  - fichier de clés SAF 173
  - gskkyman 173
  - TRUST, HIGHTRUST 173
- Variable PARM, limitations JCL 24
- variables de modèle de cliché de transaction 146
- vérification de l'installation 111
- vérification de la connexion de la passerelle client TSO/ISPF d'ISPF 118
- Vérification des paramètres de sécurité 191
- Vérification du port d'entrée 164
- vérification du port d'entrée (POE) 176
- vérification du script de shell REXEC/SSH 123

version 7.0 et version 7.1, Modifications  
entre 296  
VSAM 331  
VTAM 332

## W

WLM, règles de classification 206  
WORKAREA, Suppression des fichiers  
antérieurs du répertoire 90  
workload manager 205

## X

X.509, ajout du support d'authentification  
du client 308  
x.509, configuration de  
l'authentification 299  
Xquickstart, option Java 252

## Z

z/OS  
éléments corequis 342  
z/OS, hôte, éléments prérequis 339  
z/OS UNIX, contrôle 241





Numéro de programme : 5724-T07

SC11-6285-03

