

IBM Rational Developer for z Systems
Version 9.5.1

Guide de configuration



IBM Rational Developer for z Systems
Version 9.5.1

Guide de configuration



Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 149.

Première édition - décembre 2015

Réf. US : SC27-8577-00

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.

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

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

Cette édition concerne IBM Rational Developer for z Systems version 9.5.1 (numéro de logiciel 5724-T07 ou une partie du numéro de programme 5697-CDT) et toutes les éditions et modifications ultérieures, sauf mention contraire dans les nouvelles éditions.

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

© Copyright IBM Corporation 2015, 2015.

Tables des matières

Figures	vii
--------------------------	------------

Tableaux	ix
---------------------------	-----------

Avis aux lecteurs canadiens.	xi
---	-----------

A propos de ce manuel	xiii
--	-------------

A qui s'adresse ce guide	xiv
Récapitulatif des changements	xiv
Description du contenu du document	xvi
Planification.	xvii
Personnalisation de base	xvii
Common Access Repository Manager (CARMA)	xvii
SCLM Developer Toolkit	xvii
Débogueur intégré	xvii
Analyse de code basée sur l'hôte	xvii
Autres tâches de personnalisation	xviii
Vérification de l'installation	xviii
Définitions de sécurité	xviii
Guide de migration	xviii
Informations de référence sur la configuration hôte	xviii

IBM Rational Developer for z Systems - Guide de la configuration hôte	1
--	----------

Chapitre 1. Planification	3
--	----------

Remarques relatives à la migration	3
Remarques liées à la planification	3
Présentation du produit	3
Compétences requises	4
Temps nécessaire	4
Remarques relatives à la pré-installation	5
ID utilisateur de l'installation	5
Produits requis	6
Ressources requises	6
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	10
Remarques relatives au serveur	11
Remarques préalables au déploiement	11
Liste de contrôle du client	12

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	17

Ajout des tâches démarrées à COMMNDxx	18
Définitions SVC dans IEASVCxx	18
Activation du produit dans IFAPRDxx	18
Définitions LPA dans LPALSTxx	19
Autorisations APF dans PROGxx	20
Définitions LINKLIST dans PROGxx	20
Définitions LINKLIST et LPA prérequis	22
Définitions LINKLIST pour les autres produits	23
Modifications de PROCLIB	23
DBGMGR : Tâche démarrée par le gestionnaire de débogage	23
Procédures de construction à distance ELAXF*	24
Définitions de sécurité	26
rdz.env, le fichier de configuration d'environnement	27
Composants spécifiques	31
Vérification de l'installation	31

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

Configuration requise et liste de contrôle	33
Sélection de la méthode de démarrage du serveur et du gestionnaire RAM actif	34
Démarrage du serveur CARMA	34
CRASTART	34
Soumission par lot	34
Gestionnaire RAM de production	35
CA Endevor RAM SCM	35
Gestionnaire RAM des modules CA Endevor SCM.	35
Exemples de gestionnaire RAM.	35
Gestionnaire RAM PDS	35
Gestionnaire Skeleton	35
Gestionnaire SCLM.	35
Configurations préconfigurées de gestionnaires RAM et de démarrage du serveur	35
Gestionnaire RAM CRASTART avec CA Endevor SCM.	36
Création de fichiers CARMA VSAM	36
Personnalisation CRASRV.properties	36
Personnalisation crastart.endevor.conf	36
(Facultatif) Personnalisation additionnelle du gestionnaire RAM CA Endevor SCM	37
CRASTART avec des exemples de gestionnaires RAM	38
Création de fichiers CARMA VSAM	38
CARMA	38
Exemples de gestionnaire RAM.	38
Personnalisation CRASRV.properties	38
Personnalisation de crastart.conf	39
(Facultatif) Personnalisation additionnelle des gestionnaires RAM personnalisés	39
Soumission par lots avec le gestionnaire RAM CA Endevor SCM	40
Création de fichiers CARMA VSAM	40
Personnalisation CRASRV.properties	40

Personnalisation de CRASUBCA	40	(Facultatif) IRXJCL versus CRAXJCL	66
(Facultatif) Personnalisation additionnelle du gestionnaire RAM CA Endevor SCM	42	Création de CRAXJCL	67
Soumission d'un lot avec des exemples de gestionnaires RAM	42	Chapitre 4. SCLM Developer Toolkit .. 69	
Création des fichiers VSAM	42	Configuration requise et liste de contrôle	69
CARMA	42	Configuration requise	70
Exemples de gestionnaire RAM.	42	Mises à jour d'ISPF.conf pour SCLMDT	70
Personnalisation CRASRV.properties	43	Mises à jour rdz.env pour SCLMDT	71
Personnalisation de CRASUBMT	43	(Facultatif) Conversion de nom long/court	72
(Facultatif) Personnalisation additionnelle des gestionnaires RAM personnalisés	44	Création du fichier LSTRANS.FILE, VSAM de conversion des noms longs/abrévés	72
Informations de configuration CARMA	45	Mises à jour de rdz.env pour la conversion de noms longs/courts	74
CRASRV.properties, interface RSE avec CARMA crastart*.conf, démarrage du serveur CRASTART	45	(Facultatif) Installation et personnalisation d'Ant ..	74
Collecte des fichiers journaux CRASTART ..	51	Mises à jour SCLM pour SCLMDT	75
CRASUB*, démarrage du serveur avec soumission par lots	52	Suppression des anciens fichiers de WORKAREA et de /tmp.	76
Fichiers CARMA VSAM	54	Chapitre 5. Débogueur intégré 77	
CRADEF, fichier de configuration	54	Configuration requise et liste de contrôle	77
CRAMSG, fichier de messages	54	Débogueur intégré et COBOL v4	78
CRASTRS, fichier de chaînes personnalisées	54	Débogueur intégré et autres débogueurs reposant sur Language Environment	79
Gestionnaires RAM (Repository Access Managers) CARMA	54	Débogueur intégré et communication chiffrée	79
Gestionnaire RAM SCM CA Endevor	55	Débogueur intégré et SYSPLEX.	79
Gestionnaire RAM des modules CA Endevor SCM.	55	Paramètres de configuration du débogueur intégré	80
Gestionnaire RAM PDS	55	Mises à jour parmlib du débogueur intégré.	80
Gestionnaire Skeleton	56	Mises à jour de l'appel du superviseur du débogueur intégré	81
Gestionnaire SCLM.	56	Mises à jour TCP/IP du débogueur intégré.	81
CRACFG, CRASCL, CRASHOW et CRATMAP, les fichiers de configuration du gestionnaire RAM de CA Endevor SCM	57	Mises à jour de sécurité du débogueur intégré.	82
CRACFG, interaction du gestionnaire RAM CA Endevor SCM avec le serveur SCM	57	Mises à jour de Language Environment pour le débogueur intégré	82
CRASCL, modèle SCL RAM CA Endevor SCM	57	Mises à jour CICS du débogueur intégré	83
CRASHOW, filtres par défaut RAM CA Endevor SCM	57	Mises à jour IMS du débogueur intégré	84
CRATMAP, mappages d'extension de fichier RAM CA Endevor SCM	58	Mises à jour des procédures mémorisées DB2 pour le débogueur intégré	84
CRANDVRA et CRADYNDA, commandes exec d'allocation RAM CA Endevor SCM	58	Mises à jour ELAXF* du débogueur intégré	85
CRANDVRA, commande exec d'allocation initiale RAM CA Endevor SCM.	58	Chapitre 6. Analyse de code basée sur l'hôte 87	
CRADYNDA, commande exec de réallocation RAM CA Endevor SCM	59	Configuration requise et liste de contrôle	87
Actions par lots du gestionnaire RAM CA Endevor SCM	59	Révision du code	87
CRABCFG, configuration d'action par lots CA Endevor SCM RAM	60	Modification du traitement de révision du code	88
CRABATCA, JCL d'action par lots du RAM CA Endevor SCM	61	Couverture de code.	88
CRABJOBC, carte de travail d'actions par lots du RAM CA Endevor SCM	62	Appel de couverture de code unique	88
CRAALLOC, exec d'allocation du gestionnaire RAM personnalisé	62	Appels de couverture de code multiples.	89
Codes retour CARMA.	63	Rapports générés par la fonction de couverture de code.	89
(Facultatif) Prise en charge de plusieurs RAM	63	Chapitre 7. Autres tâches de personnalisation 91	
Exemple	64	include.conf, inclusions forcées pour l'assistant de contenu C/C++	91
(Facultatif) Exec d'allocation personnalisé	65	Sous-projets z/OS UNIX	92
(Facultatif) Exit utilisateur CARMA	66	Configuration REXEC ou SSH	93
		Prise en charge d'inclusion de précompilateur	93
		Prise en charge de xUnit pour Enterprise COBOL et PL/I.	94
		support Enterprise Service Tools	95

Support de langue bidirectionnelle CICS.	95		IBM Rational Developer for z Systems Host	
Messages IRZ de diagnostic pour Enterprise Service			Utilities, FMID HAKG950	125
Tools	96		Fichiers configurables	126
FEKRNPLI, préprocesseur distant PL/I	97		Migration de la version 9.0 vers la version 9.1 ..	128
Chapitre 8. Vérification de l'installation	99		IBM Rational Developer for z Systems, FMID	
Vérification des tâches démarrées	99		HHOP910	128
DBGMGR : Gestionnaire de débogage	99		Fichiers configurables	130
Vérification des services	99		IBM Rational Developer for z Systems Host	
Initialisation IVP	100		Utilities, FMID HAKG910	134
Connexion à CARMA	100		Fichiers configurables	135
Connexion SCLMDT	101		Migration de la version 8.5 vers la version 9.0 ..	135
Chapitre 9. Définitions de sécurité	103		IBM Rational Developer for z Systems, FMID	
Configuration requise et liste de contrôle	103		HHOP900	135
Activation des paramètres et des classes de sécurité	104		Fichiers configurables	137
Définition des tâches démarrées de Developer for z			IBM Rational Developer for z Systems Host	
Systems	104		Utilities, FMID HAKG900	141
Définition du gestionnaire de débogage en tant que			Fichiers configurables	141
serveur sécurisé z/OS UNIX	105		Chapitre 11. Informations de référence	
Définition des bibliothèques contrôlées par le			sur la configuration hôte	143
programme MVS pour le gestionnaire de débogage	105		Description de Developer for z Systems	143
Définition de l'accès au débogueur intégré.	106		Remarques relatives à la sécurité	143
Définition des profils de fichier	106		Remarques relatives à TCP/IP.	143
Vérification des paramètres de sécurité	107		Remarques à propos de WLM.	143
Chapitre 10. Guide de migration	109		Remarques relatives à l'envoi au client	143
Remarques relatives à la migration	109		Remarques relatives à CICSTS.	143
Sauvegarde des fichiers précédemment			Configuration d'AT-TLS	143
configurés	109		Bibliographie	145
Migration de la version 9.5 vers la version 9.5.1 ..	110		Publications référencées	145
IBM Rational Developer for z Systems, FMID			Publications d'information	146
HHOP951	111		Remarques	149
IBM Rational Developer for z Systems Host			Documentation sur l'interface de programmation	151
Utilities, FMID HAKG951	118		Marques	151
Fichiers configurables.	118		Conditions d'utilisation de la documentation du	
Migration de la version 9.1 vers la version 9.5 ..	119		produit	151
IBM Rational Developer for z Systems, FMID			Licence de copyright	152
HHOP950	119		Marques	152
Fichiers configurables	121			

Figures

1. DBGMGR : Tâche démarrée par le gestionnaire de débogage	24	12. crastart*.conf : démarrage du serveur CARMA en utilisant CRASTART	50
2. rdz.env : fichier de configuration d'environnement	27	13. CRASUB* : démarrage CARMA en utilisant la soumission de lot	53
3. CRASRV.properties : CRASTART avec gestionnaire RAM CA Endevor SCM	36	14. CRACFG - Interaction du gestionnaire RAM CA Endevor SCM avec le serveur SCM	57
4. crastart.endevor.conf : CRASTART avec gestionnaire RAM CA Endevor SCM	37	15. CRASHOW - Filtres par défaut RAM CA Endevor SCM.	57
5. CRASRV.properties : CRASTART avec exemples de gestionnaire RAM	39	16. CRATMAP : Filtres par défaut RAM CA Endevor SCM.	58
6. crastart.conf : CRASTART avec des exemples de gestionnaire RAM	39	17. CRABCFG : Configuration d'actions par lots CA Endevor SCM RAM	61
7. CRASRV.properties : Soumission par lots avec le gestionnaire RAM CA Endevor SCM	40	18. JCL d'action par lots CRABATCA : CA Endevor SCM RAM.	62
8. CRASUBCA : Soumission par lots avec le gestionnaire RAM CA Endevor SCM	41	19. CRABJOBC : carte de travail d'actions par lots du RAM CA Endevor SCM	62
9. CRASRV.properties : soumission d'un lot avec les exemples de gestionnaire RAM.	43	20. Mises à jour d'ISPF.conf pour SCLMDT	71
10. CRASUBMT : soumission d'un lot avec les exemples de gestionnaires RAM	44	21. Mises à jour rdz.env pour SCLMDT	72
11. CRASRV.properties – Fichier de configuration de CARMA	45	22. FLM02LST : JCL de configuration de la conversion de nom long/abrégé	73
		23. include.conf - Inclusions forcées pour l'assistant de contenu C/C++	92

Tableaux

1.	Ressources requises	6	14.	Mise à jour dynamique de l'appel du superviseur du débogueur intégré	81
2.	Ressources optionnelles	6	15.	Programmes de vérification de l'installation pour les services	99
3.	Administrateurs requis pour les tâches requises	7	16.	Variables de configuration de la sécurité	103
4.	Administrateurs nécessaires pour les tâches optionnelles	8	17.	Personnalisations de la version 9.5.1	113
5.	Liste de contrôle du client : Composants obligatoires	13	18.	Personnalisations de z/OS Explorer version 9.5.1.	117
6.	Liste de contrôle du client : Composants facultatifs	13	19.	Personnalisations d'Host Utilities version 9.0	118
7.	Correspondance entre les modules de chargement et les fonctions	20	20.	Personnalisations de la version 9.5	122
8.	Modèles de procédure ELAXF*	25	21.	Personnalisations de Host Utilities version 9.0	126
9.	Liste de contrôle des qualificatifs de haut niveau ELAXF*	26	22.	Personnalisations de la version 9.1.0	130
10.	ELAXF*	26	23.	Personnalisations de Host Utilities version 9.0	135
11.	Reconnexion automatique au gestionnaire de débogage	28	24.	Personnalisations de la version 9.0	138
12.	Codes retour CARMA	63	25.	Personnalisations de Host Utilities version 9.0	142
13.	Liste de contrôle de l'administrateur SCLM	75	26.	Publications référencées	145
			27.	Sites Web référencés	146
			28.	Publications d'information	146

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

A propos de ce manuel

Le présent document décrit la configuration des fonctions d'IBM® Rational Developer for z Systems. Il fournit des instructions pour configurer IBM Rational Developer for z Systems version 9.5.1 sur votre système hôte z/OS.

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

- *IBM Explorer for z/OS* est appelé *z/OS Explorer*.
- *IBM Rational Developer for z Systems* est appelé *Developer for z Systems*.
- *IBM Rational Developer for z Systems Integrated Debugger* est appelé *débogueur intégré*.
- *IBM Rational Developer for z Systems Interface for CA Endeavor® SCM* est appelé *gestionnaire RAM CA Endeavor® SCM* ou simplement *CA Endeavor® SCM RAM*.
- L'abréviation utilisée pour *Common Access Repository Manager* est *CARMA*.
- *Software Configuration and Library Manager Developer Toolkit* est appelé *SCLM Developer Toolkit*, abrégé en *SCLMDT*.
- *IBM z/OS Automated Unit Testing Framework* est appelé *zUnit*.
- *z/OS UNIX System Services* est appelé *z/OS UNIX*.
- *Customer Information Control System Transaction Server* est appelé *CICSTS*, abrégé en *CICS*.

Ce document fait partie d'un groupe de documents qui décrivent la configuration du système hôte Developer for z Systems. Chacun de ces documents s'adresse à des utilisateurs spécifiques. Il n'est pas nécessaire de les lire tous pour effectuer la configuration de Developer for z Systems.

- Le manuel *IBM Rational Developer for z Systems - Guide de configuration hôte* (SC43-2904) décrit de façon détaillée toutes les tâches de planification et de configuration, ainsi que les options (facultatives ou non) et propose divers scénarios.
- Le manuel *IBM Rational Developer for z Systems - Guide de référence de configuration hôte* (SC27-8578) décrit la conception de Developer for z Systems et fournit des informations connexes sur les diverses tâches de configuration de Developer for z Systems, sur les composants z/OS et autres produits (tels que WLM et CICS) associés à Developer for z Systems.

Les informations contenues dans ce document s'appliquent à tous les modules IBM Rational Developer for z Systems version 9.5.

Pour les dernières versions de ce document, voir *IBM Rational Developer for z Systems - Guide de configuration hôte* (SC23-7658) à l'adresse <http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC27-8577>.

Les versions les plus récentes de la documentation complète, y compris les instructions d'installation, les livres blancs, les podcasts et les tutoriels, sont disponibles sur la page de la bibliothèque du site Web d'IBM Rational Developer for z Systems (http://www-01.ibm.com/software/sw-library/en_US/products/Z964267S85716U24/).

A qui s'adresse ce guide

Le présent document s'adresse aux programmeurs système qui souhaitent installer et configurer IBM Rational Developer for z Systems version 9.5.1.

Ce document contient les procédures détaillées nécessaire à la configuration complète du produit, y compris certains scénarios autres que ceux fournis par défaut. Des informations de référence qui peuvent vous aider à planifier et exécuter la configuration se trouvent dans le manuel *IBM Rational Developer for z Systems - Guide de référence de la configuration hôte (SC27-8578)*. Pour utiliser ce manuel, vous devez maîtriser les systèmes hôte z/OS UNIX System Services et MVS.

Récapitulatif des changements

Cette section récapitule les modifications apportées au manuel *IBM Rational Developer for z Systems version 9.5.1 - Guide de configuration hôte*, SC43-2904-00 (mis à jour en décembre 2015).

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

Nouvelles informations :

- Informations de migration vers la version 9.5.1.
- Nouvelles directives et directives supprimées dans `rdz.env`.

Informations supprimées :

Dans la version 9.5.1, les fonctions de moniteur de travaux RSE et JES ont été déplacées d'IBM Rational Developer for z Systems vers un autre produit, IBM Explorer for z/OS. Ce déplacement inclut la documentation connexe.

- Les données propres à RSE ont été supprimées de tous les chapitres.
- Les données propres au moniteur de travaux JES ont été supprimées de tous les chapitres.
- Les données propres au service de commandes TSO ont été supprimées de tous les chapitres.
- Les données push-to-client (envoi au client par commande push) ont été supprimées de tous les chapitres.
- La documentation sur les commandes de l'opérateur a été supprimée.

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 9.5 - Guide de configuration hôte*, SC23-7658-14.

Nouvelles informations :

- Informations de migration vers la version 9.5.1.
- Informations de migration vers la version 9.5.
- Nouvelles directives et directives supprimées dans `rdz.env`.
- Commandes de l'opérateur DEBUG STORAGE et SEND.
- Prise en charge de l'envoi de message.
- Prise en charge de la passerelle ISPF interactive.
- Préprocesseur PL/I.

Informations déplacées :

- Les informations relatives au débogueur intégré font désormais l'objet d'un chapitre distinct.

Informations supprimées :

- Le gestionnaire de déploiement d'application n'est plus mis à disposition ; par conséquent, toutes les informations le concernant ont été supprimées.

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 9.1.1 - Guide de la configuration hôte*, SC11-6285-13.

Nouvelles informations :

- Définition de tâche démarrée DBGMGR modifiée.
- Nouvelle configuration de la sécurité pour le débogueur intégré.
- Nouveau statut rejectLogon pour le pool d'unités d'exécution RSE dans la commande Modify Display Process.
- Nouvelles directives et directives supprimées dans rdz.env.
- Modification des instructions de configuration supplémentaires pour le débogueur intégré en option.

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 9.1 - Guide de configuration hôte*, SC23-7658-12.

Nouvelles informations :

- Informations sur la migration de la version 9.1.0.
- Nouvelles options pour la définition de tâche démarrée RSED.
- Nouvelles directives et directives modifiées dans rdz.env.
- Nouvelles commandes de l'opérateur.
- Nouvelles options pour le débogueur intégré.

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 9.0.1 - Guide de configuration hôte*, SC23-7658-11.

Nouvelles informations :

- Instructions de configuration supplémentaires pour le débogueur intégré en option.

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 9.0.1 - Guide de la configuration hôte*, SC11-6285-10.

Nouvelles informations :

- Informations sur la migration de la version 9.0.1.
- Nouvelles mises à jour PARMLIB facultatives.
- Nouvelle tâche démarrée facultative.
- Nouvelles commandes de l'opérateur.
- Nouvelles directives et directives modifiées dans rdz.env.
- Nouvelles directives dans CRASRV.properties.

- Nouveau fichier de configuration pour le gestionnaire RAM de CARMA CA Endevor SCM.
- Prise en charge d'un exit utilisateur appelé au démarrage de CARMA.

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 9.0 - Guide de la configuration hôte*, SC11-6285-09.

Nouvelles informations :

- Informations de migration vers la version 9.0.
- Nouvelles directives et directives modifiées dans `rdz.env`.
- Nouvelles directives et directives supprimées dans `FEJJCNFG`.
- Nouveaux membres `PROCLIB ELAXF*`.
- Nouvelles commandes d'opérateur `JMON` et `RSED`.
- Nouvelles informations sur l'analyse de code sur l'hôte.

Informations supprimées :

- La tâche démarrée `LOCKD` n'est plus utilisée, c'est pourquoi toutes les informations relatives au démon lock ont été supprimées.
- L'exemple de procédure mémorisée `DB2` a été remplacée par de nouvelles procédures de construction `ELAXF*`. Toutes les informations sur la procédure mémorisée `DB2` ont donc été supprimées.
- Les informations sur la migration des éditions qui ne sont plus prises en charge ont été retirées.

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 8.5.1 - Guide de la configuration hôte*, SC11-6285-08.

Nouvelles informations :

- Nouvelles directives et directives modifiées dans `rdz.env`.
- Prise en charge améliorée pour les actions par lots CA Endevor® SCM RAM.
- Exigences ID utilisateur de l'installation

Ce document reprend certaines informations présentées précédemment dans le manuel *IBM Rational Developer for z Systems version 8.5 - Guide de la configuration hôte*, SC11-6285-07.

Nouvelles informations :

- Nouvelles directives facultatives dans `FEJJCNFG`. Voir `FEJJCNFG`, the JES Job Monitor configuration file.
- Nouvelles directives facultatives dans `rdz.env`.
- Nouveau fichier de configuration facultatif.
- Nouveau composant facultatif.
- Nouveau composant facultatif.
- Nouveau composant facultatif.
- Commandes d'opérateur nouvelles ou optimisées.

Description du contenu du document

Cette section récapitule les informations fournies dans le présent document.

Planification

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

Personnalisation de base

La procédure de personnalisation suivante s'applique à la configuration de base de Developer for z Systems :

- «Configuration personnalisée», à la page 15
- «Modifications de PARMLIB», à la page 16
- «Modifications de PROCLIB», à la page 23
- «Définitions de sécurité», à la page 26
- «rdz.env, le fichier de configuration d'environnement», à la page 27

Common Access Repository Manager (CARMA)

CARMA (Common Access Repository Manager) est une plateforme serveur pour les gestionnaires RAM (Repository Access Managers). Un gestionnaire RAM est une API (Application Programming Interface) pour un gestionnaire SCM (Software Configuration Manager) basé sur un système z/OS. En encapsulant la fonctionnalité SCM dans un gestionnaire RAM, une seule API est disponible pour permettre à un client d'accéder à un gestionnaire SCM pris en charge.

Developer for z Systems fournit des RAM préintégré et des exemples de codes source pour créer votre propre gestionnaire RAM.

IBM Rational Developer for z Systems Interface for CA Endeavor® Software Configuration Manager offre aux clients Developer for z Systems un accès direct à CA Endeavor® SCM.

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 inclus dans ISPF.

SCLM Developer Toolkit dispose d'un plug-in Eclipse qui s'interface avec SCLM et fournit l'accès à tous les processus SCLM pour le développement du code hérité et le support de développement complet Java™ et J2EE sur le poste de travail avec la synchronisation dans SCLM sur le mainframe. Les activités de synchronisation incluent création, l'assemblage et le déploiement du code J2EE depuis le mainframe.

Débogueur intégré

Le composant hôte Integrated Debugger (débogueur intégré) de Developer for z Systems permet aux clients de déboguer diverses applications Language Environment (LE).

Analyse de code basée sur l'hôte

Comme le client Developer for z Systems, l'hôte Developer for z Systems prend en charge l'exécution des outils d'analyse de code, conditionnés sous la forme d'un produit distinct, IBM Rational Developer for z Systems Host Utilities. L'exécution de l'analyse de code sur l'hôte présente l'avantage de permettre son intégration au traitement par lots des tâches quotidiennes.

Les outils d'analyse de code suivants sont disponibles sur l'hôte :

- Révision du code (Code Review) : En utilisant des règles associées à différents niveaux de sécurité, Code Review analyse le code source et signale les violations de règle.
- Couverture de code (Code Coverage) : Analyse un programme en cours et génère un rapport sur les lignes exécutées par rapport au nombre total de lignes exécutables.

Autres tâches de personnalisation

La présente section regroupe diverses tâches de personnalisation facultatives. Pour configurer le service requis, suivez les instructions de la section appropriée.

Personnalisations dans les fichiers de configuration Developer for z Systems :

- include.conf, inclusions forcées pour l'assistant de contenu C/C++

Personnalisations associées à Developer for z Systems ou pour d'autres produits :

- Sous-projets z/OS UNIX
- Prise en charge d'inclusion de précompilateur
- Prise en charge de xUnit pour Enterprise COBOL et PL/I
- Prise en charge d'Enterprise Service Tools
- Support de langue bidirectionnelle CICS
- Messages IRZ de diagnostic pour Enterprise Service Tools
- FEKRNPLI, préprocesseur distant PL/I

Vérification de l'installation

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

Définitions de sécurité

La présente section décrit les définitions de sécurité obligatoires et facultatives avec des exemples de commande RACF.

Guide de migration

La présente section met en évidence les changements apportés à l'installation et la configuration par rapport aux précédentes éditions du produit. Elle fournit également des instructions générales pour la migration de cette édition.

Informations de référence sur la configuration hôte

La présente section résume les informations contenues dans le document *Rational Developer for z Systems - Guide de référence de la configuration hôte (SC27-8578)*.

IBM Rational Developer for z Systems - Guide de la configuration hôte

Chapitre 1. Planification

Utilisez les informations du présent chapitre ainsi que celles figurant dans la section relative à la configuration logicielle requise pour Developer for z Systems qui présente les éléments requis et corequis afin de planifier l'installation et le déploiement de Developer for z Systems. Les sujets suivants sont décrits :

- «Remarques relatives à la migration»
- «Remarques liées à la planification»
- «Remarques relatives à la pré-installation», à la page 5
- «Préparation de la configuration», à la page 9
- «Remarques préalables au déploiement», à la page 11
- «Liste de contrôle du client», à la page 12

Pour la liste complète des éléments de configuration logicielle et matérielle requis pour Developer for z Systems, notamment les éléments requis et corequis, voir l'onglet **System Requirements** sur la page du produit Developer for z Systems (<http://www-03.ibm.com/software/products/en/developerforsystemz/>).

Remarques relatives à la migration

Le Chapitre 10, «Guide de migration», à la page 109 décrit les changements de configuration et d'installation par rapport aux précédentes éditions du produit. Utilisez ces informations pour planifier votre migration vers l'édition en cours de Developer for z Systems.

Remarque :

- Dans la version 9.5.1, Developer for z Systems est devenu un produit qui s'installe par-dessus IBM Explorer for z/OS. Vérifiez que vous disposez d'une version prise en charge d'IBM Developer for z/OS.
- Si vous disposez d'une version antérieure de IBM Rational Developer for System z, sauvegardez les fichiers personnalisés associés avant d'installer cette version d'IBM Rational Developer for z Systems. Pour une présentation des fichiers à personnaliser, voir Chapitre 10, «Guide de migration», à la page 109.
- Si vous prévoyez d'exécuter plusieurs instances de Developer for z Systems, voir "Exécution de plusieurs instances" dans le manuel *IBM Rational Developer for z Systems Guide de référence de la configuration hôte (SC27-8578)*.

Remarques liées à la planification

Présentation du produit

Developer for z Systems est composé d'un client, installé sur l'ordinateur personnel de l'utilisateur, et d'un serveur, installé sur un ou plusieurs systèmes hôte. Le client et l'hôte sont tous les deux installés sur IBM Explorer for z/OS. La présente documentation contient des informations pour un hôte z/OS.

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 du système 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 z Systems améliore la fonctionnalité d'accès de base fournie par z/OS Explorer. Developer for z Systems peut, par exemple, interagir avec les sous-systèmes et les autres logiciels d'application du système hôte, tels que CICS et les gestionnaires de configuration logicielle (SCM - Software Configuration Managers), si Developer for z Systems est configuré de façon appropriée et si ces produits co-requis sont disponibles.

Pour plus d'informations sur Developer for z Systems, son mode d'interaction avec votre système et les produits prérequis ou corequis, voir le manuel *Rational Developer for z Systems - Guide de référence de la configuration hôte* (SC27-8578). Le manuel *IBM Explorer for z/OS - Guide de référence de la configuration hôte* (SC27-8438) fournit des informations similaires pour z/OS Explorer, qui est un prérequis pour Developer for z Systems.

Pour des informations complémentaires sur les fonctionnalités offertes par Developer for z Systems, consultez le site Web Developer for z Systems à l'adresse <http://www-03.ibm.com/software/products/en/developersystemz/>, ou contactez votre interlocuteur IBM habituel.

Compétences requises

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

La configuration de Developer for z Systems 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 requis pour les tâches de personnalisation facultatives et obligatoires.

Temps nécessaire

Le temps nécessaire à l'installation et à la configuration des composants du système hôte Developer for z Systems varie en fonction de différents facteurs tels que :

- la configuration z/OS UNIX et TCP/IP en cours
- la disponibilité du logiciel prérequis et de la maintenance
- la disponibilité d'un utilisateur qui a installé correctement le client pour tester l'installation et signaler les problèmes qui peuvent apparaître

L'expérience a montré qu'un à deux jours sont nécessaires pour mener à bien le processus d'installation et de configuration du système hôte Developer for z Systems, ce processus n'incluant pas l'installation et la configuration d'IBM Explorer for z/OS, qui est un produit prérequis. 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

Pour des instructions détaillées sur l'installation SMP/E du produit, voir le document *Répertoire de programme pour IBM Rational Developer for z Systems* (GI11-7314).

Les serveurs Developer for z Systems reconnaissent leur propre système, pas un SYSPLEX. Si vous utilisez les serveurs dans un SYSPLEX, vous devez vous assurer que les données demandées par les utilisateurs finaux (fichiers, sortie de travaux, fichiers z/OS UNIX) sont disponibles sur le système sur lequel Developer for z Systems est installé. Voir «Remarques préalables au déploiement», à la page 11 pour le clonage de Developer for z Systems sur d'autres systèmes.

Pour exécuter plusieurs instances de Developer for z Systems sur un système hôte unique, voir "Exécution de plusieurs instances" dans le manuel *Rational Developer for z Systems Guide de référence de la configuration hôte* (SC27-8578).

Le système de fichiers (HFS ou zFS) dans lequel Developer for z Systems 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 z Systems de créer l'environnement de sécurité de l'utilisateur et rejette les demandes de connexion du client. Cela est également vrai pour les systèmes de fichiers hébergeant les binaires z/OS Explorer, Java et z/OS UNIX.

ID utilisateur de l'installation

L'ID utilisateur utilisé pour installer Developer for z Systems, ou de la maintenance, doit avoir 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 groupe par défaut de l'ID utilisateur nécessite un GID.
- UID=0 ou autorisation READ dans le profil BPX.SUPERUSER de la classe FACILITY.
- Si les profils BPX.FILEATTR.APF ou BPX.FILEATTR.PROGCTL sont définis dans la classe FACILITY, accès READ à ces profils.
- Des droits d'accès READ, WRITE et EXECUTE au répertoire /tmp (ou un répertoire référencé dans la variable d'environnement TMPDIR).

Produits requis

Developer for z Systems a une liste de logiciels prérequis qui doivent être installés et opérationnels pour que le produit fonctionne. Ce chapitre contient également la liste des éléments logiciels corequis pour prendre en charge les fonctions de Developer for z Systems. Ces éléments requis doivent être installés et opérationnels au moment de l'exécution pour que les fonctions correspondantes opèrent selon leur conception.

Pour la liste complète des éléments de configuration logicielle requis pour Developer for z Systems, notamment les éléments requis et corequis, voir l'onglet **System Requirements** sur le site Web de Developer for z Systems (<http://www-03.ibm.com/software/products/en/developersystemz/>).

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 éléments requis principaux pour une configuration de base sont les suivants :

- z/OS 1.12 ou version suivante
- IBM Explorer for z/OS 3.0
- Edition de service la plus récente de Java 6.0 ou versions supérieures (31 ou 64 bits)

Ressources requises

Developer for z Systems nécessite d'allouer les ressources système répertoriées dans le tableau 1. Les ressources répertoriées dans le tableau 2 sont requises pour des services en option. Prévoyez la disponibilité de ces ressources car, en fonction des règles en vigueur sur votre site, l'obtention des logiciels peut prendre du temps.

Tableau 1. Ressources requises

Ressource	Valeur par défaut	Information
Procédures de construction MVS	ELAXF*	«Modifications de PROCLIB», à la page 23

Tableau 2. Ressources optionnelles

Ressource	Valeur par défaut	Information
Démarrage du système avec CLPA	non applicable	Chapitre 5, «Débogueur intégré», à la page 77
Tâche démarrée	DBGMR	Chapitre 5, «Débogueur intégré», à la page 77
Fichier LINKLIST	FEL.SFELAUTH et FEL.SFELLOAD	<ul style="list-style-type: none">• Chapitre 4, «SCLM Developer Toolkit», à la page 69• Chapitre 5, «Débogueur intégré», à la page 77

Tableau 2. Ressources optionnelles (suite)

Ressource	Valeur par défaut	Information
Fichier LPA	FEL.SFELLPA	<ul style="list-style-type: none"> • «Définitions LPA dans LPALSTxx», à la page 19 • Chapitre 3, «Common Access Repository Manager (CARMA)», à la page 33 • Chapitre 5, «Débogueur intégré», à la page 77
Profils de sécurité	AQE.**	<ul style="list-style-type: none"> • Chapitre 5, «Débogueur intégré», à la page 77
plage de ports pour une utilisation limitée au système hôte	tout port disponible est utilisé	<ul style="list-style-type: none"> • Chapitre 3, «Common Access Repository Manager (CARMA)», à la page 33 • Chapitre 6, «Analyse de code basée sur l'hôte», à la page 87
plage de ports pour une utilisation limitée au système hôte	5336	Chapitre 5, «Débogueur intégré», à la page 77
port pour les communications client-hôte	<ul style="list-style-type: none"> • 5335 pour le débogueur intégré 	<ul style="list-style-type: none"> • Chapitre 5, «Débogueur intégré», à la page 77
mise à jour de la définition système CICS	valeurs multiples	<ul style="list-style-type: none"> • Chapitre 5, «Débogueur intégré», à la page 77
mise à jour du JCL CICS	<ul style="list-style-type: none"> • FEL.SFELLOAD • FEL.SFELAUTH 	<ul style="list-style-type: none"> • «Support de langue bidirectionnelle CICS», à la page 95 • Chapitre 5, «Débogueur intégré», à la page 77

La configuration de Developer for z Systems 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 et le tableau 4, à la page 8 répertorient les administrateurs requis pour les tâches de personnalisation facultatives et obligatoires.

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

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"> • Définition des profils de fichier • Définition de fichiers contrôlés par un programme • Définition de droits pour la soumission de travaux xxx* • Définition de profils CICS pour le débogage • Ajout d'un certificat pour la communication chiffrée • Définition des tâches démarrées • Définition des profils de serveur z/OS UNIX • Définition de profils de débogage • Définition de profils pour l'envoi de messages 	<ul style="list-style-type: none"> • "Remarques relatives à la sécurité" dans <i>Guide de référence de la configuration hôte</i> (SC11-6869)
TCP/IP	Définition de nouveaux ports TCP/IP	"Ports TCP/IP" dans <i>Guide de référence de la configuration hôte</i> (SC27-8578)
SCLM	<ul style="list-style-type: none"> • Définition de traducteurs de langage SCLM pour la prise en charge de Java EE • Définition de types SCLM pour la prise en charge de Java EE 	Chapitre 4, «SCLM Developer Toolkit», à la page 69
CICS TS	<ul style="list-style-type: none"> • Mise à jour du JCL de la région CICS • Mise à jour de la définition système de la région CICS • Définition de groupe CICS • Définition de noms de transaction CICS • Définition d'un programme dans CICS • Définition du débogueur dans CICS 	<ul style="list-style-type: none"> • «Support de langue bidirectionnelle CICS», à la page 95 • Chapitre 5, «Débogueur intégré», à la page 77
WLM	<ul style="list-style-type: none"> • Affectation d'objectifs aux tâches Developer for z Systems 	<ul style="list-style-type: none"> • "Remarques relatives à WLM" dans le <i>Guide de référence de la configuration hôte</i> (SC27-8578) (SC11-6869)

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

Administrateur	Tâche	Information
LDAP	Définition de groupes pour l'envoi au client	“Considérations push-to-client” dans le <i>Guide de référence de la configuration hôte (SC27-8578)</i>

Préparation de la configuration

Pour plus d'informations sur Developer for z Systems, son mode d'interaction avec votre système et les produits prérequis ou corequis, voir le manuel *Rational Developer for z Systems - Guide de référence de la configuration hôte (SC27-8578)*. Le manuel *IBM Explorer for z/OS - Guide de référence de la configuration hôte (SC27-8438)* fournit des informations similaires pour z/OS Explorer, qui est un prérequis pour Developer for z Systems. Ces informations peuvent vous aider à créer une configuration qui répond à vos besoins en cours et d'évolutivité ultérieurs.

Gestion de la charge de travail

Contrairement aux applications z/OS classiques, Developer for z Systems n'est pas une application monolithique qui peut être identifiée facilement au niveau du Workload Manager (WLM). Les différents composants qui constituent Developer for z Systems interagissent pour offrir au client un accès aux services et données du système hôte. Pour planifier votre configuration WLM, voir "Remarques relatives à WLM" dans le manuel *Guide de référence de la configuration hôte (SC27-8578)*.

Remarque : Developer for z Systems 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. Les 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 z Systems.

Utilisation des ressources et limites du système

Developer for z Systems 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. Pour évaluer l'utilisation des ressources clé afin de planifier votre configuration système, voir "Remarques relatives à l'optimisation" dans le manuel *Guide de référence de la configuration hôte (SC27-8578)*. Developer for z Systems peut s'exécuter en mode 31 bits ou 64 bits, en modifiant considérablement les limitations de ressources de stockage.

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 qui peuvent être omises sont répertoriées ici :

- Tous les utilisateurs de Developer for z Systems doivent disposer de droits READ et EXECUTE dans les répertoires Java.

- 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 le système hôte.

Remarques relatives à l'ID utilisateur

L'ID d'un utilisateur de Developer for z Systems doit comporter 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 mégaotets) 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, afin 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
```

- Droits d'accès READ et EXECUTE aux répertoires et fichiers d'installation et de configuration de z/OS Explorer and Developer for z Systems, par défaut /usr/lpp/ibm/zexpl/*, /etc/zexpl/*, /var/rdz/*, /usr/lpp/ibm/rdz/* et /var/rdz/*.
- Droits d'accès READ, WRITE et EXECUTE au répertoire WORKAREA de z/OS Explorer, par défaut /var/zexpl/WORKAREA, et au répertoire des journaux d'utilisateur, par défaut /var/zexpl/logs.
- Droits d'accès READ aux fichiers d'installation de z/OS Explorer, par défaut FEK.SFEK* et FEL.SFEL*.
- Droits d'accès READ, WRITE et EXECUTE au répertoire /tmp ou à un répertoire référencé dans la variable d'environnement TMPDIR.

Remarques relatives au serveur

z/OS Explorer et Developer for z Systems se composent de plusieurs 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. Il n'existe pas d'ordre de démarrage spécifique. Seule exigence : les serveurs doivent être sous tension et en cours d'exécution avant que le premier utilisateur ne tente de se connecter. Pour être efficaces, les mécanismes de sécurité utilisés par les serveurs et les services z/OS Explorer et Developer for z Systems doivent résider dans des fichiers et des systèmes de fichiers sécurisés. Cela implique que seuls les administrateurs système habilités doivent pouvoir mettre à jour les bibliothèques de programmes et les fichiers de configuration.

- Le gestionnaire de débogage (DBGMR) fournit des services relatifs au débogage.
- Le moniteur de travaux JES (JMON) offre tous les services liés à JES. JMON fait partie de z/OS Explorer.
- L'Explorateur de systèmes distants (RSE) fournit des services de base, comme la connexion du client au système hôte et le démarrage d'autres serveurs pour des services spécifiques. RSE fait partie de z/OS Explorer. 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.

Comme documenté dans "Ports TCP/IP" dans le manuel *Guide de référence de la configuration hôte (SC27-8578)*, certains systèmes hôte, et donc leurs ports, doivent être accessibles au client en connexion et être définis dans le pare-feu qui protège le système hôte. Tous les autres ports utilisés par Developer for z Systems reçoivent un trafic réservé à l'hôte. La liste ci-après répertorie les ports requis pour les communications externes dans une configuration de base du produit Developer for z Systems.

- 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 disponibles à une plage définie.

Remarques préalables au déploiement

Developer for z Systems 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 dans un autre emplacement, il doit remplacer son équivalent dans la liste ci-après.

Remarque : La liste suivante ne couvre pas les besoins liés au déploiement des logiciels prérequis et corequis (y compris z/OS Explorer).

Developer for z Systems

- FEL.SFELAUTH(*)
- FEL.SFELLMOD(*)

- FEL.SFELLOAD(*)
- FEL.SFELLPA(*)
- FEL.SFELPROC(*)
- FEL.#CUST.CNTL(*)
- FEL.#CUST.PARMLIB(*)
- FEL.#CUST.PROCLIB(*)
- /usr/lpp/ibm/rdz/*
- /var/rdz/*
- définitions, fichiers et répertoires issus des travaux de personnalisation dans FEL.#CUST.JCL

Utilitaires de l'hôte Developer for z Systems

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

Remarque :

- FEL et /usr/lpp/ibm/rdz correspondent au qualificatif de haut niveau et au chemin d'accès utilisés lors de l'installation de Developer for z Systems. FEL.#CUST, /etc/zexpl et /var/rdz désignent les emplacements par défaut utilisés au cours de la personnalisation du produit (/etc/zexpl étant le répertoire de configuration z/OS Explorer).
 - AKG et /usr/lpp/ibm/rdzutil correspondent au qualificatif de haut niveau et au chemin d'accès utilisés lors de l'installation des Developer for z Systems Host Utilities.
 - Il est recommandé d'installer Developer for z Systems dans un système de fichiers privé (HFS ou zFS), si possible combiné avec z/OS Explorer, pour faciliter le déploiement des composants z/OS UNIX du produit. Si vous ne pouvez pas utiliser un système de fichiers privé, utilisez un outil d'archivage, tel que la commande z/OS UNIX tar, pour transférer les répertoires z/OS UNIX d'un système à un autre. Cette méthode est destinée à préserver les attributs (tels que le contrôle par programme) des fichiers et répertoires Developer for z Systems.
- Pour plus d'informations sur les exemples de commandes suivantes relatives à l'archivage et la restauration du répertoire d'installation Developer for z Systems, voir le manuel *UNIX System Services Command Reference* (SA22-7802).
- Archivage : `cd /SYS1/usr/lpp/ibm/rdz; tar -cSf /u/userid/rdz.tar`
 - Restauration : `cd /SYS2/usr/lpp/ibm/rdz; tar -xSf /u/userid/rdz.tar`

Liste de contrôle du client

Les utilisateurs du client Developer for z Systems doivent connaître les résultats de certaines personnalisations du système 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, à la page 13 répertorie les résultats requis pour les étapes de personnalisation obligatoires. Le tableau 6, à la page 13 répertorie les résultats requis des étapes de personnalisation facultatives.

Tableau 5. Liste de contrôle du client : Composants obligatoires

Personnalisation	Valeur
(prérequis) Numéro de port TCP/IP du démon RSE. La valeur par défaut est 4035.	
Ce port est défini durant la configuration de z/OS Explorer.	

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. La valeur par défaut est FEL.#CUST.PROCLIB. Voir les annotations sur JCLLIB dans «Procédures de construction à distance ELAXF*», à la page 24.	
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 24.	
Emplacement de la procédure AKGCR si elle ne se trouve pas dans une bibliothèque de procédure système. La valeur par défaut est AKG.#CUST.PROCLIB. Voir les annotations sur JCLLIB dans «Révision du code», à la page 87.	
Emplacement de la procédure AKGCC si elle ne se trouve pas dans une bibliothèque de procédure système. La valeur par défaut est AKG.#CUST.PROCLIB. Voir les annotations sur JCLLIB dans «Couverture de code», à la page 88.	
Emplacement de l'instruction d'exécutable de préprocesseur d'inclusion FEKRNPLI. La valeur par défaut est FEL.#CUST.CNTL. Voir «Prise en charge d'inclusion de précompilateur», à la page 93.	
Emplacement des modules de chargement du débogueur s'ils ne se trouvent pas dans LINKLIST. La valeur par défaut est FEL.SFELAUTH. Voir Chapitre 5, «Débogueur intégré», à la page 77	
Emplacement des modules de chargement de test d'unité s'ils ne se trouvent pas dans LINKLIST ou STEPLIB de rdz.env. La valeur par défaut est FEL.SFELLOAD. Voir «Prise en charge de xUnit pour Enterprise COBOL et PL/I», à la page 94.	
Emplacement de la procédure AZUZUNIT si elle ne se trouve pas dans une bibliothèque de procédure système. La valeur par défaut est FEL.#CUST.PROCLIB. Voir les annotations sur JCLLIB dans «Prise en charge de xUnit pour Enterprise COBOL et PL/I», à la page 94.	
Emplacement des fichiers XML exemple *.xsd et *.xsl utilisés pour le formatage de sortie du test d'unité. Les valeurs par défaut sont /usr/lpp/ibm/rdz/samples/zunit/xsd et /usr/lpp/ibm/rdz/samples/zunit/xsl. Voir «Prise en charge de xUnit pour Enterprise COBOL et PL/I», à la page 94.	

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

Personnalisation	Valeur
(corequis) Numéro de port TN3270 de Host Connect Emulator. La valeur par défaut est 23. Voir "Ports TCP/IP" dans <i>Guide de référence de la configuration hôte</i> (SC27-8578).	
(corequis) Numéro de port REXEC ou SSH, dont les valeurs par défaut sont 512 ou 22. Voir «Sous-projets z/OS UNIX», à la page 92.	
(corequis) Numéro de port du serveur Debug Tool (pas de valeur par défaut). Voir (Facultatif) Prise en charge du débogage de DB2 et IMS.	
Emplacement de l'exemple de bibliothèque SFELSAMP pour les exemples de gestionnaire RAM CARMA. La valeur par défaut est FEL.SFELSAMP. Voir le manuel <i>CARMA Developer's Guide</i> (SC23-7660).	
Emplacement du JCL CRA#ASLM pour les allocations de fichier RAM SCLM CARMA. La valeur par défaut est FEL.#CUST.JCL. Voir les annotations concernant CRA#ASLM dans «Gestionnaire SCLM», à la page 56.	

Chapitre 2. Personnalisation de base

Les étapes de personnalisation suivantes sont communes aux divers services Developer for z Systems. Reportez-vous aux chapitres relatifs aux composants spécifiques 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 LPA
- fichiers avec des droits APF
- 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
- (Facultatif) Démarrage du système pour activer un appel du superviseur (SVC)

Pour vérifier l'installation et commencer à utiliser Developer for z Systems 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 z Systems. Pour plus d'informations, voir «Configuration personnalisée».
2. Mettez à jour les limites système z/OS UNIX ainsi que l'enregistrement du produit, lancez des tâches démarrées et définissez des fichiers LINKLIST avec des droits APF et, éventuellement, des appels de superviseur (SVC) et des fichiers LPA. Pour plus d'informations, voir «Modifications de PARMLIB», à la page 16.
3. Créez des procédures de tâche démarrée, et des procédures de compilation et de liaison. Pour plus d'informations, voir «Modifications de PROCLIB», à la page 23.
4. Mettez à jour les définitions de sécurité. Pour plus d'informations, voir «Définitions de sécurité», à la page 26.
5. Personnalisez les fichiers de configuration Developer for z Systems. Pour plus d'informations, voir :
 - «rdz.env, le fichier de configuration d'environnement», à la page 27

Configuration personnalisée

Developer for z Systems contient plusieurs exemples de fichier de configuration et de JCL. Pour que vos personnalisations ne soient pas remplacées lors de l'application de la maintenance, copiez tous ces membres et fichiers z/OS UNIX dans un emplacement différent, puis personnalisez la copie.

Certaines fonctions de Developer for z Systems requièrent 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, FELSETUP, est fourni pour créer les copies et les répertoires requis.

Pour créer des copies personnalisables des fichiers de configuration et du JCL de configuration, et pour créer les répertoires z/OS UNIX requis, personnalisez et soumettez l'exemple de membre FELSETUP dans le fichier FEL.SFELSAMP. La procédure de personnalisation requise est décrite dans ce membre.

Dans ce travail vous devez effectuer les tâches suivantes :

- Créez FEL.#CUST.PARMLIB et remplissez-le avec les exemples de fichiers de configuration.
- Créez FEL.#CUST.PROCLIB et remplissez-le avec des exemples de membres SYS1.PROCLIB.
- Créez FEL.#CUST.JCL et remplissez-le avec un exemple de code JCL de configuration.
- Créez FEL.#CUST.CNTL et remplissez-le avec des exemples de scripts de démarrage de serveur.
- Créez FEL.#CUST.ASM et remplissez-le avec un exemple de code source en assembleur.
- Créez FEL.#CUST.SQL et remplissez-le avec les exemples de fichiers de commandes SQL.
- Remplissez le répertoire de configuration z/OS Explorer, /etc/zexpl/*, avec les exemples de fichiers de configuration.
- Créez /var/rdz/* comme répertoires de travail pour plusieurs fonctions Developer for z Systems et alimentez ces répertoires avec des exemples de fichier.

Remarque :

- Les procédures de configuration de ce manuel utilisent les emplacements membre/fichier créés par le travail FELSETUP, sauf indication contraire. Les exemples d'origine, qui ne peuvent pas être mis à jour, se trouvent dans FEL.SFELSAMP et /usr/lpp/ibm/rdz/samples/.
- Pour savoir quels exemples de membres sont copiés vers quel fichier, et pour plus d'informations sur les répertoires créés, leur masque de contrôle des données de droits et l'emplacement où les différents exemples de fichiers sont copiés, voir les commentaires dans FEL.SFELSAMP(FELSETUP).
- Pour faciliter la migration d'une configuration existante, les commentaires de FEL.SFELSAMP(FELSETUP) documentent également les changements entre les différentes versions de Rational Developer for z Systems.
- Pour conserver tous les fichiers z/OS Explorer et Developer for z Systems z/OS UNIX dans le même système de fichiers (HFS ou zFS), mais également placer les fichiers de configuration dans /etc/zexpl, 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/ibm/rdz/cust) et de définir un lien symbolique (/etc/zexpl) vers ce dernier :

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

Modifications de PARMLIB

Les modifications PARMLIB suivantes sont documentées dans cette section :

- «Définition des limites z/OS UNIX dans BPXPRMxx», à la page 17
- «Ajout des tâches démarrées à COMMNDxx», à la page 18
- «Définitions SVC dans IEASVCxx», à la page 18
- «Activation du produit dans IFAPRDxx», à la page 18

- «Définitions LPA dans LPALSTxx», à la page 19
- «Autorisations APF dans PROGxx», à la page 20
- «Définitions LINKLIST dans PROGxx», à la page 20
- «Définitions LINKLIST et LPA prérequis», à la page 22
- «Définitions LINKLIST pour les autres produits», à la page 23

Pour plus d'informations sur les définitions PARMLIB répertoriées dans les sections suivantes, voir le manuel *MVS Initialization and Tuning Reference* (SA22-7592). Pour plus d'informations sur les exemples de commande de la console, voir le manuel *MVS System Commands* (SA22-7627).

Définition des limites z/OS UNIX dans BPXPRMxx

SCLM Developer Toolkit démarre simultanément plusieurs processus appartenant aux utilisateurs durant certaines activités utilisateur.

Définissez OMVS=xx dans le membre parmlib IEASYSxx pour indiquer quel membre parmlib BPXPRMxx doit être utilisé lors du démarrage du système.

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, car il doit être actif pour chaque client utilisant Developer for z Systems.

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 MAXPROCUSER=50

Remarque :

- La valeur MAXPROCUSER suggérée ici 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).
- Lors de l'installation SMP/E de Developer for z Systems, vous avez été conseillé de placer le code dans un système de fichiers distinct (zFS de HFS) et de mettre à jour BPXPRMxx pour monter ce système de fichiers durant l'IPL du système. Voici une répétition de l'exemple de commande de montage si cette mise à jour est encore à effectuer :

```
MOUNT FILESYSTEM('#dsn')
  MOUNTPPOINT('-PathPrefix-usr/lpp/rdz')
  MODE(RDWR) /* can be MODE(READ) */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
/* TYPE(HFS) */ /* HFS, auto. extent */
```

- Lors de l'installation SMP/E de Developer for z Systems Host Utilities, il vous a été conseillé de placer le code dans un système de fichiers distinct (zFS de HFS) et de mettre à jour BPXPRMxx pour monter ce système de fichiers durant l'IPL du système. Voici une répétition de l'exemple de commande de montage si cette mise à jour est encore à effectuer :

```
MOUNT FILESYSTEM('#dsn')
  MOUNTPPOINT('-PathPrefix-usr/lpp/rdzutil')
  MODE(RDWR) /* can be MODE(READ) */
  TYPE(ZFS) PARM('AGGRGROW') /* zFS, with extents */
/* TYPE(HFS) */ /* HFS, auto. extent */
```

Ajout des tâches démarrées à COMMNDxx

Ajoutez des commandes de démarrage pour les serveurs Developer for z Systems à SYS1.PARMLIB(COMMANDxx) de façon à les démarrer automatiquement lors du prochain démarrage du système. Définissez CMD=xx dans le membre parmlib IEASYSxx pour indiquer quel membre parmlib COMMNDxx doit être utilisé lors du démarrage du système.

Le débogueur intégré facultatif requiert que le serveur Developer for z Systems DBGMR soit actif sur votre système.

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 DBGMR

Remarque : Il n'existe pas d'ordre de démarrage spécifique pour les serveurs. Seule exigence : les serveurs doivent être sous tension et en cours d'exécution avant que le premier utilisateur ne tente de se connecter.

Définitions SVC dans IEASVCxx

Le débogueur intégré facultatif nécessite qu'un appel du superviseur Developer for z Systems soit défini pour votre système.

Les appels de superviseur définis par l'installation le sont dans SYS1.PARMLIB(IEASVCxx) et requièrent l'activation d'un démarrage du système. Le module de chargement associé doit être chargé dans la zone permanente de programme au démarrage du système. Notez cependant que le débogueur intégré vous permet de procéder à une installation ou à une mise à jour dynamique à l'aide de la tâche démarrée du gestionnaire de débogage. Définissez SVC=xx dans le membre parmlib IEASYSxx pour indiquer quel membre parmlib IEASVCxx doit être utilisé lors du démarrage du système.

Spécifiez le code suivant dans IEASVCxx pour définir l'appel de superviseur Developer for z Systems :

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

Le numéro d'appel du superviseur 251 est le numéro par défaut, mais toute valeur comprise dans la plage de 200 à 255 imposée par z/OS peut être utilisée. Le débogueur intégré va détecter quel appel du superviseur est utilisé.

Remarque : Les versions de Developer for z Systems antérieures à la version 9.1.1 utilisaient un autre SVC, AQESVC01, qui n'est plus utilisé. Il peut être supprimé si votre système hôte utilise uniquement Developer for z Systems version 9.1.1 ou une version ultérieure.

Activation du produit dans IFAPRDxx

Si vous avez acheté Developer for z Systems avec le code produit 5697-CDT, IBM Enterprise COBOL Suite for z/OS versions 1.1 ou supérieures, et que vous ne l'avez pas encore fait pour un autre programme du code produit 5697-CDT, incluez une entrée au membre IFAPRDxx parmlib pour autoriser les programmes connexes. Définissez PROD=xx dans le membre IEASYSxx parmlib pour spécifier quel membre IFAPRDxx parmlib utilisé lors du démarrage du système.

Pour définir Enterprise COBOL Suite for z/OS (code produit 5697-CDT), dans IFAPRDxx, spécifiez :

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

Sinon, si vous le souhaitez, et si vous avez acheté Developer for z Systems séparément (PAS avec Enterprise COBOL Suite for z/OS), vous pouvez inclure une entrée dans le membre IFAPRDxx parmlib pour Developer for z Systems à l'aide du code produit autonome 5724-T07.

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

Une fois le membre IFAPRDxx parmlib mis à jour, il peut être activé de manière dynamique (jusqu'au prochain démarrage système) à l'aide de la commande de console suivante :

```
SET PROD=xx
```

Remarque : Developer for z Systems enregistre les fonctions suivantes :

- RDZ-RSED (pour Developer for z Systems)
- RDZ-CC (pour la couverture de code dans Developer for z Systems Host Utilities)
- RDZ-CR (pour la révision du code dans Developer for z Systems Host Utilities)

Définitions LPA dans LPALSTxx

Le service facultatif CARMA (Common Access Repository Manager) prend en charge différentes méthodes de démarrage de serveur pour le serveur CARMA. La méthode de démarrage CRASTART startup a besoin que les modules de la bibliothèque de chargement FEL.SFELLPA se trouvent dans la zone permanente de programme (LPA - Link Pack Area).

Les fichiers LPA sont définis dans SYS1.PARMLIB(LPALSTxx). Définissez LPA=xx dans le membre parmlib IEASYSxx pour indiquer quel membre parmlib LPALSTxx doit être utilisé lors du démarrage du système.

Les définitions LPA peuvent être définies de manière dynamique (jusqu'au prochain démarrage système) à l'aide de la commande de console suivante :

- SETPROG LPA,ADD,DSN=FEL.SFELLPA,MASK=*

Remarque :

- Les fichiers répertoriés dans LPALSTxx doivent être catalogués dans le catalogue maître ou dans un catalogue utilisateur identifié dans le membre LPALSTxx.
- L'ajout d'un fichier à LPALSTxx requiert l'activation d'un démarrage du système avec CLPA (create LPA).
- Toutes les bibliothèques chargées dans LPA sont automatiquement considérées comme étant compatible APF et contrôlées par programme. Assurez-vous de détenir les contrôles de sécurité adaptés en place pour ces bibliothèques.

- Si vous choisissez de ne pas placer une bibliothèque conçue pour le placement LPA dans LPA et d'utiliser LINKLIST ou STEPLIB à la place, assurez-vous d'avoir défini l'autorisation APF et l'état du contrôle de programmes.

Autorisations APF dans PROGxx

Pour que le gestionnaire de débogage facultatif fonctionne, les modules des bibliothèques de chargement FEL.SFELAUTH et FEL.SFELLPA doivent avoir des droits APF. Notez que le module FEL.SFELLPA doit être dans STEPLIB pour le gestionnaire de débogage et nécessite donc une autorisation APF explicite.

Pour que le service SCLM Developer Toolkit facultatif fonctionne, la bibliothèque d'exécution REXX (REXX.*.SEAGLPA) doit avoir des droits APF.

Par défaut, les droits APF sont définis dans SYS1.PARMLIB(PROGxx). Définissez PROG=xx dans le membre parmlib IEASYSxx pour indiquer quel membre parmlib PROGxx doit être utilisé lors du démarrage du système.

Les droits APF peuvent être définis 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=FEL.SFELAUTH,SMS
- SETPROG APF,ADD,DSN=FEL.SFELLPA,SMS
- SETPROG APF,ADD,DSN=REXX.V1R4M0.SEAGLPA,VOL=volser

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 utilisé dans l'exemple précédent.
- Les bibliothèques LPA, telles que REXX.*.SEAGLPA, ont automatiquement des droits APF lorsqu'elles se trouvent dans la zone LPA et elles ne requièrent donc pas de définitions explicites.
- Certains des produits prérequis et corequis (z/OS Explorer, par exemple) requièrent également des droits APF. Pour plus d'informations, consultez les guides de personnalisation produit associés.

Définitions LINKLIST dans PROGxx

Les définitions LINKLIST pour Developer for z Systems peuvent être regroupées en trois catégories :

- Bibliothèques de chargement Developer for z Systems, nécessaires aux fonctions Developer for z Systems. Ces définitions sont décrites dans la présente section.
- Bibliothèques de chargement d'éléments prérequis nécessaires aux fonctions Developer for z Systems. Ces définitions sont présentées dans «Définitions LINKLIST et LPA prérequis», à la page 22.
- Bibliothèques de chargement Developer for z Systems nécessaires à d'autres produits. Ces définitions sont présentées dans «Définitions LINKLIST pour les autres produits», à la page 23.

Tableau 7. Correspondance entre les modules de chargement et les fonctions

Bibliothèque de chargement	Modules de chargement	Utilisation	STEPLIB
FEL.SFELAUTH	AQE* et CEE*	Chapitre 5, «Débogueur intégré», à la page 77	Procédure ELAXFGO, ou CICS

Tableau 7. Correspondance entre les modules de chargement et les fonctions (suite)

Bibliothèque de chargement	Modules de chargement	Utilisation	STEPLIB
FEL.SFELLMOD	IRZ* et IIRZ*	«Messages IRZ de diagnostic pour Enterprise Service Tools», à la page 96	Lot CICS, IMS ou MVS
FEL.SFELLOAD	AZU* et IAZU*	«Prise en charge de xUnit pour Enterprise COBOL et PL/I», à la page 94	rdz.env ou lot MVS
	BWB*	Chapitre 4, «SCLM Developer Toolkit», à la page 69	rdz.env
	CRA*	Chapitre 3, «Common Access Repository Manager (CARMA)», à la page 33	CRASUB* ou crastart*.conf
	ELAX*	«Procédures de construction à distance ELAXF*», à la page 24 (retour d'informations sur les erreurs et préprocesseur d'inclusion)	Procédures ELAXF*
	FEJB*	«Support de langue bidirectionnelle CICS», à la page 95	CICS
FEL.SFELLPA	CRA*	Chapitre 3, «Common Access Repository Manager (CARMA)», à la page 33	CRASRV.properties
	AQE*	Chapitre 5, «Débogueur intégré», à la page 77	non applicable (zone LPA obligatoire)

Pour que les services Developer for z Systems répertoriés fonctionnent, tous les modules documentés dans le tableau 7, à la page 20 qui sont liés au service doivent être mis à disposition via STEPLIB ou LINKLIST (ou la zone permanente de programme). Notez que la bibliothèque SFELLMOD n'est pas utilisée par Developer for z Systems lui-même, mais par le code généré par Developer for z Systems. Reportez-vous à la colonne STEPLIB dans le tableau 7, à la page 20 si vous choisissez d'utiliser STEPLIB pour savoir où effectuer la définition de STEPLIB (ou DFHRPL pour CICS). Néanmoins, vous devez tenir compte des considérations suivantes :

- L'utilisation de STEPLIB dans z/OS UNIX a un impact négatif sur les performances.
- Si une bibliothèque STEPLIB est dotée de droits 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 dans un JCL ne sont pas transmises aux processus z/OS UNIX démarrés par JCL.

Les fichiers LINKLIST sont définis dans SYS1.PARMLIB(PROGxx), si votre site se conforme aux recommandations IBM. Définissez PROG=xx dans le membre parmlib IEASYSxx pour indiquer quel membre parmlib PROGxx doit être utilisé lors du démarrage du système.

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

- LNKST ADD NAME(`nom_liste`) DSNAME(FEL.SFELAUTH) VOLUME(`num_série_volume`)
- LNKST ADD NAME(`nom_liste`) DSNAME(FEL.SFELLOAD)

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, `num_série-volume` étant le volume dans lequel réside le fichier s'il n'est pas placé dans le catalogue maître :

1. LNKST DEFINE,NAME=LLTMP,COPYFROM=CURRENT
2. LNKST ADD NAME=LLTMP,DSN=FEL.SFELAUTH,VOL=`num_série_volume`
3. LNKST ADD NAME=LLTMP,DSN=FEL.SFELLOAD
4. LNKST ACTIVATE,NAME=LLTMP

Définitions LINKLIST et LPA prérequis

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 z Systems (z/OS Explorer, par exemple) :

- Bibliothèque de chargement système
 - SYS1.LINKLIB
- Environnement d'exécution Language Environment
 - CEE.SCEERUN
 - CEE.SCEERUN2
- Bibliothèque de classes DLL de C++
 - CBC.SCLBDLL
- Passerelle client TSO/ISPF d'ISPF (pour SCLMDT)
 - ISP.SISPLoad
 - ISP.SISPLPA
- Bibliothèque d'exécution REXX (pour SCLMDT)
 - REXX.*.SEAGLPA
- Bibliothèque de chargement système (pour le débogueur intégré)
 - SYS1.MIGLIB
- Bibliothèque de chargement système (pour le débogueur intégré sur z/OS version 1.13 et ultérieures)
 - SYS1.SIEAMIGE
- Bibliothèque de chargement système (pour le test d'unité Enterprise COBOL et PL/I)
 - SYS1.CSSLIB
 - SYS1.SIXML0D1

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 utilisé dans l'exemple précédent.
- Toutes les bibliothèques chargées dans LPA sont automatiquement considérées comme étant compatible APF et contrôlées par programme. Assurez-vous de détenir les contrôles de sécurité adaptés en place pour ces bibliothèques.

- Les bibliothèques conçues pour le positionnement LSA, telles que REXX.*.SEAGLPA, peuvent requérir des autorisations de contrôle par programme ou APF supplémentaires si elles sont accessibles via LINKLIST ou STEPLIB.
- Certains des produits prérequis et corequis (z/OS Explorer, par exemple) requièrent également des définitions STEPLIB ou LINKLIST/LPALIB. Pour plus d'informations, consultez les guides de personnalisation produit associés.

Par défaut, les fichiers LINKLIST sont définis dans SYS1.PARMLIB(PROGxx). Les fichiers LPA sont définis dans SYS1.PARMLIB(LPALSTxx).

Si vous décidez d'utiliser STEPLIB, vous devez définir les bibliothèques non disponibles via LINKLIST/LPALIB dans la directive STEPLIB du fichier de configuration RSE, rdz.env. 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 dispose de droits APF, toutes les autres bibliothèques doivent également être autorisées. Les bibliothèques perdent leur autorisation APF lorsqu'elles sont mélangées avec des bibliothèques non autorisées dans STEPLIB.
- Les bibliothèques ajoutées à STEPLIB DD dans un JCL ne sont pas transmises aux processus z/OS UNIX démarrés par JCL.

Définitions LINKLIST pour les autres produits

Le client Developer for z Systems contient un composant de génération de code appelé Enterprise Service Tools. Pour que le code généré émette des messages d'erreur de diagnostic, tous les modules IRZM* et IIRZ* de la bibliothèque de chargement FEL.SFELMOD doivent être disponibles par l'intermédiaire de STEPLIB ou de LINKLIST.

Par défaut, les fichiers LINKLIST sont définis dans SYS1.PARMLIB(PROGxx).

Si vous décidez d'utiliser STEPLIB, vous devez définir les bibliothèques non disponibles via LINKLIST dans la directive STEPLIB de la tâche qui exécute le code (IMS ou travail par lots). Toutefois, si une bibliothèque STEPLIB dispose de droits APF, toutes les autres bibliothèques doivent également être autorisées. Les bibliothèques perdent leur autorisation APF lorsqu'elles sont mélangées avec des bibliothèques non autorisées dans STEPLIB.

Modifications de PROCLIB

Les modifications PROCLIB suivantes sont documentées dans cette section :

- «DBGMGR : Tâche démarrée par le gestionnaire de débogage»
- «Procédures de construction à distance ELAXF*», à la page 24

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

DBGMGR : Tâche démarrée par le gestionnaire de débogage

Personnalisez l'exemple FEL.#CUST.PROCLIB(DBGMGR) de membre de tâche démarrée, comme décrit dans le membre puis copiez-le dans SYS1.PROCLIB. Comme indiqué dans l'exemple de code suivant, indiquez les informations suivantes :

- Le décalage horaire (par défaut, EST5DST)
- Le port utilisé pour les communications externes (client-hôte) (par défaut, 5335)
- Le port utilisé pour les communications internes (limitées au système hôte) (par défaut, 5336)
- Numéro SVC utilisé par Integrated Debugger, par défaut 251
- Le qualificatif de haut niveau de la bibliothèque de chargement, par défaut FEL

```

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

```

Figure 1. DBGMGR : Tâche démarrée par le gestionnaire de débogage

Remarque :

- Il s'agit d'une tâche démarrée facultative. Elle est utilisée par la fonction de débogueur intégré de Developer for z Systems. Pour plus d'informations, voir Chapitre 5, «Débogueur intégré», à la page 77.
- Cet exemple de fichier JCL, initialement nommé FEL.SFELSAMP(AQJCL) est renommé FEL.#CUST.PROCLIB(DBGMGR) lors de l'étape «Configuration personnalisée», à la page 15.
- Pour connaître les objectifs Workload Manager (WLM) conseillés pour cette tâche, voir les considérations relatives à WLM dans le *Guide de référence de la configuration hôte* (SC27-8578).
- Si le numéro SVC Integrated Debugger est déjà chargé, le numéro SVC spécifié ici est ignoré et le numéro SVC actif est utilisé.

Procédures de construction à distance ELAXF*

Developer for z Systems 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 8, à la page 25.

Tableau 8. 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 de pré-traitement DB2 des instructions SQL EXEC imbriquées dans des programmes COBOL.
ELAXFCOT	Modèle de procédure pour l'exécution d'une traduction CICS des instructions CICS EXEC imbriquées dans des programmes COBOL.
ELAXFCPC	Exemple de procédure pour la compilation C.
ELAXFCPP	Exemple de procédure pour la compilation 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 de pré-traitement DB2 des instructions SQL EXEC imbriquées 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	Exemple de procédure pour la compilation PL/I et traductions CICS et DB2 intégrées.
ELAXFPP1	Exemple de procédure pour la compilation PL/I avec des instructions du préprocesseur SCM (-INC et ++INCLUDE).
ELAXFSP	Exemple de procédure pour enregistrer une procédure mémorisée dans DB2.
ELAXFSQL	Exemple de procédure pour appeler SQL.
ELAXFTSO	Exemple de procédure pour l'exécution et le débogage du code DB2 généré 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 incluses dans le client Developer for z Systems. Si le nom d'une procédure ou le nom d'une étape de la procédure est changé, le fichier des propriétés correspondant sur tous les clients doit également être mis à jour. Il est recommandé de ne pas changer les noms de la procédure et de l'étape.

Personnalisez les exemples de membres de procédure de génération, FEL.#CUST.PROCLIB(ELAXF*), comme décrit dans les membres, et copiez-les dans

SYS1.PROCLIB. Indiquez les qualificatifs de haut niveau appropriés des différentes bibliothèques de produits, comme décrit dans le tableau 9.

Tableau 9. Liste de contrôle des qualificatifs de haut niveau ELAXF*

Produit	Valeur par défaut HLQ	Valeur
Developer for z Systems	FEK	
CICS	CICSTS52.CICS	
DB2	DSNA11	
IMS	IMS	
COBOL	IGY.V5R2M0	
PL/I	PLI.V4R2M0	
C/C++	CBC	
LE	CEE	
système LINKLIB	SYS1	
système MACLIB	SYS1	

Certaines procédures ELAXF* référencent des noms de fichier qui n'ont pas de qualificatifs de bas niveau. Il s'agit par exemple de la bibliothèque d'exécution DB2, qui contient les utilitaires DB2 compilés par votre administrateur DB2. Utilisez le tableau 10 pour mapper les noms de fichier par défaut aux noms utilisés dans votre site.

Tableau 10. ELAXF*. Liste de contrôle de noms de fichier qualifiés complets

Produit	DSN par défaut	Valeur
Developer for z Systems - Exemples SQL	FEL.#CUST.SQL	
Bibliothèque d'exécution DB2	DSNA11.RUNLIB.LOAD	

Si les procédures ELAXF* ne peuvent pas être copiées dans une bibliothèque de procédures système, demandez aux utilisateurs de Developer for z Systems 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=(FEL.#CUST.PROCLIB)
```

Définitions de sécurité

Pour créer les définitions de sécurité de Developer for z Systems, personnalisez et soumettez les exemples de membres FELRACF et AQERACF. L'utilisateur qui soumet le travail doit disposer des privilèges d'administrateur de la sécurité (RACF SPECIAL, par exemple).

FELRACF et AQERACF se trouvent dans FEL.#CUST.JCL sauf si vous avez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

La liste suivante des définitions associées à la sécurité de Developer for z Systems est décrite en détail dans Chapitre 9, «Définitions de sécurité», à la page 103.

- Activation des paramètres et des classes de sécurité

- Définition du gestionnaire de débogage en tant que serveur sécurisé z/OS UNIX
- Définition de l'accès au débogueur intégré
- Définition des profils de fichier

rdz.env, le fichier de configuration d'environnement

Les processus serveur RSE z/OS Explorer (démon RSE, pool d'unités d'exécution RSE et serveur RSE) utilisent les définitions figurant dans le fichier `rdz.env` pour prendre connaissance des variables d'environnement Developer for z Systems.

`rdz.env` se trouve dans `/etc/zexpl/`, sauf si vous spécifiez un emplacement différent lors de la personnalisation et de la soumission du travail `FEL.SFELSAMP(FELSETUP)`. 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 `rdz.env` suivant, qui doit être personnalisé pour correspondre à votre environnement système. Les valeurs par défaut sont fournies pour toutes les variables qui ne sont pas explicitement spécifiées. La syntaxe du fichier est conforme aux règles de syntaxe de shell z/OS UNIX standard. Par exemple, les lignes de commentaire commencent par un signe dièse (#) lorsque vous utilisez une page de codes US et les espaces autour du signe égal (=) ne sont pas autorisés.

Remarque : Pour que vos modifications soient prises en charge, vous devez redémarrer la tâche démarrée z/OS Explorer RSED.

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

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

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

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

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

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

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

Figure 2. `rdz.env` : fichier de configuration d'environnement

Les définitions suivantes sont facultatives. Si vous les omettez, ce sont les valeurs par défaut qui sont utilisées.

RDZ_HOME

Répertoire principal Developer for z Systems. La valeur par défaut est /usr/lpp/ibm/rdz. Supprimez la mise en commentaire et modifiez cette valeur en fonction de votre installation Developer for z Systems.

RDZ_HLQ

Qualificatif de haut niveau utilisé pour installer Developer for z Systems. La valeur par défaut est FEK. Supprimez la mise en commentaire et modifiez cette valeur pour qu'elle corresponde à l'emplacement de vos fichiers Developer for z Systems.

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

Reconnexion automatique au gestionnaire de débogage. La valeur par défaut est 1, ce qui implique que, lorsqu'une connexion avec le serveur de gestionnaire de débogage n'est pas établie ou est perdue, le logiciel de fouille de données de débogage va tenter une fois de se reconnecter au gestionnaire de débogage. Supprimez la mise en commentaire et indiquez une autre valeur pour limiter la fréquence à laquelle le logiciel de fouille de données de débogage tente de se reconnecter au gestionnaire de débogage.

Tableau 11. Reconnexion automatique au gestionnaire de débogage

debug.miner.autoreconnect	Comportement de reconnexion
-1	Ne pas reconnecter
0 (valeur par défaut)	Tentative de reconnexion toutes les minutes jusqu'à aboutissement
1-86400	Tentative de reconnexion dans la limite de la durée spécifiée. La valeur maximale, 86400, est égale à 24 heures.

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

Méthode alternative pour la définition TCP/IP de localhost. Le logiciel de fouille de données de débogage tente de se connecter au gestionnaire de débogage à l'aide de la spécification localhost. Cette tentative échoue si localhost ne résout pas l'adresse de bouclage local (127.0.0.1 pour IVPv4, ::1 pour IPv6). Supprimez la mise en commentaire et indiquez l'adresse de bouclage local si nécessaire.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dinclude.c=/etc/zexpl/include.conf"

Cette variable pointe vers un nom de fichier z/OS UNIX complet contenant une liste d'inclusions forcées pour l'assistant de contenu en code C. Une inclusion forcée signifie qu'un fichier, un répertoire ou un membre de fichier est analysé lors de l'exécution d'une opération d'assistant de contenu, même si ce fichier ou membre a été inclus dans le code source à l'aide d'une directive de précompilateur. Pour spécifier le nom du fichier de configuration, supprimez la mise en commentaire et effectuez une personnalisation.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -Dinclude.cpp=/etc/zexpl/include.conf"

Cette variable pointe vers un nom de fichier z/OS UNIX complet contenant une liste d'inclusions forcées pour l'assistant de contenu en code C++. Une inclusion forcée signifie qu'un fichier, un répertoire ou un membre de fichier est analysé lors de l'exécution d'une opération d'assistant de contenu, même si ce fichier ou membre a été inclus dans le

code source à l'aide d'une directive de précompilateur. Pour spécifier le nom du fichier de configuration, supprimez la mise en commentaire et effectuez une personnalisation.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"

Intervalle de nettoyage des fichiers d'en-tête C/C++ non utilisés, en millisecondes. La valeur par défaut est de 60000, c'est-à-dire 1 minute. Pour modifier l'intervalle de nettoyage, supprimez la mise en commentaire et personnalisez. La valeur 0 empêche la mise en cache des fichiers d'en-tête C/C++, ce qui réduit les performances de l'assistant de contenu distant dans l'éditeur.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DRIS_BUFFER=8"

Taille de mémoire tampon, exprimée en mégaoctets, utilisée lors de la création d'index distante. La valeur par défaut est 8 Mo. Pour modifier la taille de mémoire tampon, supprimez la mise en commentaire et personnalisez. Les valeurs valides sont des nombres entiers compris entre 1 et 2000 (inclus). Une mémoire tampon plus volumineuse permet d'accélérer la création d'index, mais utilise une partie plus importante du segment de mémoire Java du pool d'unités d'exécution. La mémoire tampon est automatiquement vidée dans l'index si elle est pleine avant la fin de la création d'index.

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS

-DDISABLE_REMOTE_INDEX_SEARCH=true"

Désactivation de l'option de menu de recherche dans des index distants sur le client. La valeur par défaut est false. Pour empêcher les utilisateurs de créer des index à distance pour les fichiers du système hôte, supprimez la mise en commentaire et indiquez true.

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS

-DDISABLE_DELETE_IN_SUBPROJECT=true"

Désactivation de l'option de menu Supprimer dans le contenu contextuel des sous-projets z/OS. La valeur par défaut est false. Pour empêcher les utilisateurs d'utiliser l'option de menu Supprimer dans le menu contextuel de sous-projets z/OS, supprimez la mise en commentaire et indiquez true.

STEPLIB

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

Vous pouvez ignorer la nécessité d'avoir les bibliothèques prérequis dans LINKLIST/LPALIB en supprimant la mise en commentaire d'une ou plusieurs des instructions STEPLIB suivantes et en les personnalisant. Pour plus d'informations sur l'utilisation des bibliothèques dans la liste suivante, voir «Modifications de PARMLIB», à la page 16 :

```
# SCLM Developer Toolkit
STEPLIB=$STEPLIB:$RSE_HLQ.SFELAUTH:$RSE_HLQ.SFELLOAD
# Prise en charge zUnit, xUnit pour Enterprise COBOL et PL/I
STEPLIB=$STEPLIB:$RSE_HLQ.SFELLOAD:SYS1.CSSLIB:SYS1.SIXML0D1
```

Remarque :

- L'utilisation de STEPLIB dans z/OS UNIX a un impact négatif sur les performances.
- Si une bibliothèque STEPLIB dispose de droits APF, toutes les autres bibliothèques doivent également être autorisées. 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_UBLD_DD

Spécifie les instructions de définition de données qui seront utilisées lors de la génération du JCL des générations utilisateur IBM Rational Team Concert à partir d'un client Developer for z Systems qui appellent des commandes TSO ou ISPF. Par défaut, Developer for z Systems utilise les définitions du fichier ISPF.conf, qui est référencé par CGI_ISPCONF dans rdz.env. Annulez la mise en commentaire de cette définition et modifiez-la pour utiliser les définitions de données du fichier spécifié, qui doivent respecter les règles de syntaxe spécifiées dans *ISPF.conf*, le fichier de configuration de la passerelle client TSO/ISPF d'ISPF. Il s'agit d'une directive facultative.

RSE_UBLD_STEPLIB

Spécifie l'instruction STEPLIB qui sera utilisée lors de la génération du JCL des générations utilisateur IBM Rational Team Concert à partir d'un client Developer for z Systems qui appellent des commandes TSO ou ISPF. Par défaut, Developer for z Systems utilise la définition STEPLIB dans rdz.env. Supprimez la mise en commentaire pour utiliser la définition STEPLIB spécifiée. Il s'agit d'une directive facultative.

_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 uniquement utilisée lorsque SCLMDT est utilisé.

Remarque : SCLMDT ajoute /CONFIG et /CONFIG/PROJECT au chemin spécifié dans SCLMDT_CNF_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. La valeur par défaut est FEL.#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 uniquement utilisée 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 uniquement utilisée lorsque la prise en charge de la génération Java EE est utilisée avec SCLM Developer Toolkit.

Composants spécifiques

Developer for z Systems se compose de diverses fonctions non connexes, chacune ayant ses propres tâches de personnalisation. Suivez les instructions de la section adéquate pour configurer le service requis.

Personnalisations des composants autonomes de Developer for z Systems :

- Chapitre 3, «Common Access Repository Manager (CARMA)», à la page 33
- Chapitre 4, «SCLM Developer Toolkit», à la page 69
- Chapitre 5, «Débogueur intégré», à la page 77
- Chapitre 6, «Analyse de code basée sur l'hôte», à la page 87

Personnalisations des fichiers de configuration Developer for z Systems :

- «include.conf, inclusions forcées pour l'assistant de contenu C/C++», à la page 91

Personnalisations associées à Developer for z Systems ou pour d'autres produits :

- «Sous-projets z/OS UNIX», à la page 92
- «Prise en charge d'inclusion de précompilateur», à la page 93
- «Prise en charge de xUnit pour Enterprise COBOL et PL/I», à la page 94
- «support Enterprise Service Tools», à la page 95
- «Support de langue bidirectionnelle CICS», à la page 95
- «Messages IRZ de diagnostic pour Enterprise Service Tools», à la page 96
- «FEKRNPLI, préprocesseur distant PL/I», à la page 97

Vérification de l'installation

Vous trouverez la description détaillée des différents programmes de vérification d'installation (IVP pour Installation Verification Program) dans Chapitre 8, «Vérification de l'installation», à la page 99.

Chapitre 3. Common Access Repository Manager (CARMA)

CARMA (Common Access Repository Manager) est une plateforme serveur pour les gestionnaires RAM (Repository Access Managers). Un gestionnaire RAM est une API (Application Programming Interface) pour un gestionnaire SCM (Software Configuration Manager) z/OS. En encapsulant la fonctionnalité SCM dans un gestionnaire RAM, une seule API est disponible pour permettre à un client d'accéder à un gestionnaire SCM pris en charge.

Developer for z Systems fournit des RAM préintégré et des exemples de codes source pour créer votre propre gestionnaire RAM.

Les SCM basés sur des systèmes hôte ont besoin d'espaces adresse d'utilisateur unique pour accéder à leurs services, ce qui nécessite que CARMA démarre un serveur CARMA par utilisateur. Il est impossible de créer un serveur unique prenant en charge plusieurs utilisateurs.

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 ou les ressources suivantes :

- (Facultatif) Plage de ports TCP/IP pour les communications internes
- (Facultatif) Règle de sécurité afin d'autoriser les développeurs à mettre à jour les fichiers CARMA VSAM
- (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, effectuez les tâches ci-après. Sauf indication contraire, toutes les tâches sont obligatoires.

1. Sélectionnez une méthode pour démarrer CARMA et les gestionnaires à activer. Plusieurs combinaisons de gestionnaires RAM et de démarrage du serveur sont disponibles comme configurations préconfigurées. Pour plus d'informations, voir «Sélection de la méthode de démarrage du serveur et du gestionnaire RAM actif», à la page 34.
2. Créez des fichiers CARMA VSAM. Pour plus d'informations, voir «Fichiers CARMA VSAM», à la page 54 et «Gestionnaires RAM (Repository Access Managers) CARMA», à la page 54.
3. Effectuez la personnalisation initiale des fichiers de configuration RSE pour communiquer avec CARMA. La personnalisation complète dépend de la méthode choisie pour démarrer CARMA. Pour plus d'informations, voir «CRASRV.properties, interface RSE avec CARMA», à la page 45.
4. Selon la méthode de démarrage CARMA choisie et les gestionnaires RAM, effectuez la personnalisation nécessaire des fichiers de configuration associés. Pour plus d'informations, voir :
 - «crastart*.conf, démarrage du serveur CRASTART», à la page 48
 - «CRASUB*, démarrage du serveur avec soumission par lots», à la page 52
5. Personnalisez éventuellement les membres de configuration spécifiques à CA Endevor® SCM. Pour plus d'informations, voir «CRACFG, CRASCL,

CRASHOW et CRATMAP, les fichiers de configuration du gestionnaire RAM de CA Endevor® SCM», à la page 57 et «Actions par lots du gestionnaire RAM CA Endevor® SCM», à la page 59.

6. Mettez éventuellement à jour la commande exec d'allocation de fichier. Pour plus d'informations, voir «CRANDVRA, commande exec d'allocation initiale RAM CA Endevor® SCM», à la page 58, «CRAALLOC, exec d'allocation du gestionnaire RAM personnalisé», à la page 62 et «(Facultatif) Exec d'allocation personnalisé», à la page 65.
7. Créez éventuellement un exit utilisateur de démarrage. Pour plus de détails, voir (Facultatif) Exit utilisateur CARMA.
8. Créez éventuellement CRAXJCL en remplacement de IRXJCL. Pour plus d'informations, voir «(Facultatif) IRXJCL versus CRAXJCL», à la page 66.

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

Sélection de la méthode de démarrage du serveur et du gestionnaire RAM actif

Developer for z Systems prend en charge différentes méthodes de démarrage du serveur CARMA. Developer for z Systems fournit également plusieurs gestionnaires RAM (Repository Access Managers) qui peuvent être répartis dans deux groupes : gestionnaires de production RAM et exemples de gestionnaires RAM. Cette publication décrit plusieurs combinaisons de gestionnaires RAM et méthodes de démarrage du serveur. Chacun des scénarios de configuration décrits est disponible sous la forme d'une installation préconfigurée.

Démarrage du serveur CARMA

Developer for z Systems prend en charge différentes méthodes de démarrage du serveur CARMA. Chaque méthode présente des avantages et des inconvénients.

CRASTART

La méthode "CRASTART" démarre le serveur CARMA comme sous-tâche dans 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.

Soumission par lot

La méthode de soumission par lot démarre le serveur CARMA en envoyant un travail. Il s'agit de la méthode par défaut utilisée dans les fichiers de configuration fournis. Avantage de cette méthode : 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.

Gestionnaire RAM de production

Les gestionnaires de production RAM sont des gestionnaires RAM complètement fonctionnels préconfigurés qui peuvent être utilisés pour accéder à un gestionnaire SCM dans un environnement de production.

CA Endevor® RAM SCM

IBM Rational Developer for z Systems Interface for CA Endevor® Software Configuration Manager offre aux clients Developer for z Systems un accès direct à CA Endevor® SCM.

Gestionnaire RAM des modules CA Endevor® SCM

Le gestionnaire RAM CA Endevor® SCM permet aux clients Developer for z Systems d'accéder directement aux modules CA Endevor® SCM.

Exemples de gestionnaire RAM

Des exemples de gestionnaire RAM permettent de tester la configuration de votre environnement CARMA et servent d'exemples pour le développement de votre propre RAM. Le code source est inclus.

Avertissement : N'utilisez pas les exemples de RAM fournis dans un environnement de production.

Gestionnaire RAM PDS

Le gestionnaire RAM PDS donne une liste de fichiers analogues aux fichiers MVS -> **Mes fichiers** de la vue Systèmes distants.

Gestionnaire Skeleton

Le gestionnaire RAM Skeleton fournit une infrastructure fonctionnelle que vous pouvez utiliser comme point de départ pour développer votre propre gestionnaire RAM.

Gestionnaire SCLM

Le gestionnaire RAM SCLM fournit une entrée de base dans SCLM, le gestionnaire de configuration de logiciel d'ISPF. Le gestionnaire RAM SCLM n'est pas activé par défaut.

Configurations préconfigurées de gestionnaires RAM et de démarrage du serveur

Plusieurs combinaisons de gestionnaires RAM et de démarrage du serveur sont disponibles comme configurations préconfigurées. Les scénarios répertoriés ont besoin uniquement d'une personnalisation mineure par rapport à votre environnement.

- «Gestionnaire RAM CRASTART avec CA Endevor® SCM», à la page 36
- «CRASTART avec des exemples de gestionnaires RAM», à la page 38
- «Soumission par lots avec le gestionnaire RAM CA Endevor® SCM», à la page 40
- «Soumission d'un lot avec des exemples de gestionnaires RAM», à la page 42

Pour plus d'informations sur les étapes de chaque scénario, voir «Informations de configuration CARMA», à la page 45.

Il est possible d'ajouter un gestionnaire RAM à toute installation CARMA, immédiatement ou ultérieurement. Voir «(Facultatif) Prise en charge de plusieurs RAM», à la page 63 pour plus d'informations sur l'ajout d'un gestionnaire RAM à une installation existante.

Gestionnaire RAM CRASTART avec CA Endevor® SCM

Les informations de cette section expliquent comment configurer CARMA avec les spécifications suivantes :

- Démarrage du serveur : méthode CRASTART. Cette méthode requiert la présence de CRASTART dans la zone permanente de programme.
- Gestionnaire RAM : gestionnaire RAM CA Endevor® SCM.

Cette étape de personnalisation peut être omise si vous voulez utiliser l'un des autres scénarios avec des spécifications différentes.

Création de fichiers CARMA VSAM

Pour définir et remplir les fichiers VSAM associés à CARMA, personnalisez et envoyez les travaux JCL suivants. Pour des instructions de personnalisation, voir la documentation du membre. Les fichiers VSAM existants sont remplacés.

Pour plus de détails sur cette étape, voir «Fichiers CARMA VSAM», à la page 54.

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

Personnalisation CRASRV.properties

Le serveur RSE utilise les paramètres de /etc/zexpl/CRASRV.properties pour démarrer et se connecter au serveur CARMA. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Pour que les changements apportés prennent effet, redémarrez la tâche démarrée RSED.

Lorsque vous utilisez les emplacements de fichier par défaut, vous devez remplacer uniquement la valeur de la directive `clist.dsname` par `*CRASTART` et la valeur de `crastart.configuration.file` par `crastart.endevor.conf`. Pour plus d'informations sur les différentes directives, voir «CRASRV.properties, interface RSE avec CARMA», à la page 45.

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

Figure 3. CRASRV.properties : CRASTART avec gestionnaire RAM CA Endevor® SCM

Personnalisation crastart.endevor.conf

CRASTART utilise les définitions situées dans /etc/zexpl/crastart.endevor.conf pour créer un environnement TSO/ISPF valide pour démarrer un serveur CARMA. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

Pour des instructions de personnalisation, voir la documentation du fichier. Pour plus d'informations sur la méthode de démarrage CRASTART, voir «crastart*.conf, démarrage du serveur CRASTART», à la page 48.

Remarque : Compte tenu des limitations de largeur de page, certaines lignes de l'exemple suivant sont renvoyées à la ligne. Toutes les lignes qui sont mises en retrait doivent être ajoutées à la fin de la ligne précédente.


```

* DD used by RAM
TYPEMAP = FEL.#CUST.PARMLIB(CRATMAP)
SHOWVIEW= FEL.#CUST.PARMLIB(CRASHOW)
CRACFG = FEL.#CUST.PARMLIB(CRACFG)
* uncomment CRABCFG and CRABSKEL to use batch actions
*CRABCFG = FEL.#CUST.PARMLIB(CRABCFG)
*CRABSKEL= FEL.#CUST.CNTL
* uncomment and provide correct DSN to use Package Ship
*APIHJC = #shiphjc
CONLIB = CA.NDVR.CSIQLOAD
-COMMAND=ALLOC FI(JCLOUT)  SYSOUT(A) WRITER(INTRDR) RECFM(F) LRECL(80)
  BLKSIZE(80)
-COMMAND=ALLOC FI(EXT1ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT2ELM) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(EXT1DEP) NEW DELETE DSORG(PS) RECFM(V,B) LRECL(4096)
  BLKSIZE(27998) SPACE(5,5) TRACKS UNIT(SYSALLDA)
C1EXMSG= SYSOUT(H)
C1MSG1 = SYSOUT(H)
MSG3FILE= DUMMY

* DD used by CARMA server (CRASERV)
* pay attention to APF authorizations when using TASKLIB
TASKLIB = FEL.SFELLOD,CA.NDVR.CSIQAUTH,CA.NDVR.CSIQAUTU
CRADEF = FEL.#CUST.CRADEF
CRAMSG = FEL.#CUST.CRAMSG
CRASTRS = FEL.#CUST.CRASTRS
CARMALOG = SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (via NDVRC1)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

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

* CRANDVRA

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

```

Figure 4. *crastart.endevor.conf* : *CRASTART* avec gestionnaire RAM CA Endevor® SCM

(Facultatif) Personnalisation additionnelle du gestionnaire RAM CA Endevor® SCM

Le gestionnaire RAM CA Endevor® SCM dispose de composants supplémentaires que vous pouvez personnaliser si nécessaire.

- Le traitement du démarrage CARMA comporte un exit utilisateur facultatif. Pour plus d'informations, voir (Facultatif) Exit utilisateur CARMA.

- Le gestionnaire RAM CA Endevor® SCM dispose de plusieurs fichiers de configuration FEL.#CUST.PARMLIB(CRA*) que vous pouvez personnaliser. Pour plus d'informations, voir «CRACFG, CRASCL, CRASHOW et CRATMAP, les fichiers de configuration du gestionnaire RAM de CA Endevor® SCM», à la page 57.
- Le gestionnaire RAM CA Endevor® SCM dispose d'un exécutable d'allocation, FEL.SFELPROC(CRANDVRA), que vous pouvez personnaliser. Pour plus d'informations, voir «CRANDVRA, commande exec d'allocation initiale RAM CA Endevor® SCM», à la page 58.
- Le gestionnaire RAM CA Endevor® SCM prend en charge les actions CA Endevor® SCM en mode de traitement par lots. Les actions par lots requièrent un fichier de configuration, FEL.#CUST.PARMLIB(CRABCFG), et un JCL de squelette, FEL.#CUST.CNTL(CRABATCA), qui doit être personnalisé. Pour plus d'informations, voir «Actions par lots du gestionnaire RAM CA Endevor® SCM», à la page 59.

CRASTART avec des exemples de gestionnaires RAM

Les informations de cette section expliquent comment configurer CARMA avec les spécifications suivantes :

- Démarrage du serveur : méthode CRASTART. Cette méthode requiert la présence de CRASTART dans la zone permanente de programme.
- Gestionnaire RAM : exemples de gestionnaire RAM, qui ne sont pas destinés à une utilisation à des fins de production.

Cette étape de personnalisation peut être ignorée si vous voulez utiliser l'un des autres scénarios avec des spécifications différentes.

Création de fichiers CARMA VSAM

Personnalisez et envoyez les travaux JCL suivants pour définir et remplir les fichiers VSAM associés à CARMA. Pour des instructions de personnalisation, voir la documentation du membre. Les fichiers VSAM existants sont remplacés.

Pour plus de détails sur cette étape, voir «Fichiers CARMA VSAM», à la page 54 et «Gestionnaires RAM (Repository Access Managers) CARMA», à la page 54.

CARMA

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

Exemples de gestionnaire RAM

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

Personnalisation CRASRV.properties

Le serveur RSE utilise les paramètres de /etc/zexpl/CRASRV.properties pour démarrer et se connecter au serveur CARMA. Vous pouvez modifier le fichier à l'aide de la commande TSO **0EDIT**. Pour que les changements apportés prennent effet, vous devez redémarrer la tâche démarrée RSED.

Lorsque vous utilisez les emplacements de fichier par défaut, vous devez remplacer uniquement la valeur de la directive clist.dsname par *CRASTART. Pour plus d'informations sur les différentes directives, voir «CRASRV.properties,

interface RSE avec CARMA», à la page 45.

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

Figure 5. CRASRV.properties : CRASTART avec exemples de gestionnaire RAM

Personnalisation de crastart.conf

CRASTART utilise les définitions situées dans /etc/zexpl/crastart.conf pour créer un environnement TSO/ISPF valide pour démarrer un serveur CARMA. Vous pouvez modifier le fichier à l'aide de la commande TSO **OEDIT**. Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

Pour des instructions de personnalisation, voir la documentation du fichier. Pour plus d'informations sur la méthode de démarrage CRASTART, voir «crastart*.conf, démarrage du serveur CRASTART», à la page 48.

```
* DD used by RAM
CRARAM1 = FEL.#CUST.CRARAM1                                * PDS RAM
* DD used by CARMA server (CRASERV)
TASKLIB = FEL.SFELLOAD
CRADEF  = FEL.#CUST.CRADEF
CRAMSG  = FEL.#CUST.CRAMSG
CASTRS  = FEL.#CUST.CASTRS
CARMALOG = SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (ISPSTART)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

* DD used by TSO (IKJEFT01)
SYSPROC = #hlq.SFELPROC                                * CRAALLOC
SYSTSIN  = DUMMY
SYTSPRT  = SYSOUT(H)

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

Figure 6. crastart.conf : CRASTART avec des exemples de gestionnaire RAM

Remarque : Compte tenu des limitations de largeur de page, certaines lignes de l'exemple sont renvoyées à la ligne. Toutes les lignes qui sont mises en retrait doivent être ajoutées à la fin de la ligne précédente.

(Facultatif) Personnalisation supplémentaire des gestionnaires RAM personnalisés

Les gestionnaires RAM personnalisés possèdent des composants supplémentaires que vous pouvez personnaliser si nécessaire.

- Le traitement du démarrage CARMA comporte un exit utilisateur facultatif. Pour plus d'informations, voir (Facultatif) Exit utilisateur CARMA.

- Le démarrage du gestionnaire RAM personnalisé dispose d'un exécutable d'allocation, `FEL.SFELPROC(CRAALLOC)`, que vous pouvez personnaliser. Pour plus d'informations, voir `CRAALLOC`, exec d'allocation du gestionnaire RAM personnalisé.

Soumission par lots avec le gestionnaire RAM CA Endevor® SCM

Les informations de cette section expliquent comment configurer CARMA avec les spécifications suivantes :

- Démarrage du serveur : méthode de soumission par lots. Cette méthode nécessite des initiateurs JES.
- Gestionnaire RAM : gestionnaire RAM CA Endevor® SCM.

Cette étape de personnalisation peut être omise si vous voulez utiliser l'un des autres scénarios avec des spécifications différentes.

Création de fichiers CARMA VSAM

Personnalisez et envoyez les travaux JCL suivants pour définir et remplir les fichiers VSAM associés à CARMA. Pour des instructions de personnalisation, voir la documentation du membre. Les fichiers VSAM existants sont remplacés.

Pour plus de détails sur cette étape, voir «Fichiers CARMA VSAM», à la page 54.

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

Personnalisation CRASRV.properties

Le serveur RSE utilise les paramètres de `/etc/zexpl/CRASRV.properties` pour démarrer et se connecter au serveur CARMA. Vous pouvez modifier le fichier à l'aide de la commande TSO **EDIT**. Pour que les changements apportés prennent effet, vous devez redémarrer la tâche démarrée RSED.

Lorsque vous utilisez les emplacements de fichier par défaut, vous devez remplacer uniquement la valeur de la directive `clist.dsname` par `FEL.#CUST.CNTL(CRASUBCA)`. Pour plus d'informations sur les différentes directives, voir «CRASRV.properties, interface RSE avec CARMA», à la page 45.

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

Figure 7. CRASRV.properties : Soumission par lots avec le gestionnaire RAM CA Endevor® SCM

Personnalisation de CRASUBCA

La liste de commandes `FEL.#CUST.CNTL(CRASUBCA)` et le JCL intégré envoient un serveur CARMA. Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

Pour des instructions de personnalisation, voir la documentation du membre. Pour plus d'informations sur la méthode de démarrage de soumission par lots, voir «CRASUB*, démarrage du serveur avec soumission par lots», à la page 52.

```

PROC 8 CRAPRM1 CRAPRM2 CRAPRM3 CRAPRM4 CRAPRM5 CRAPRM6 CRAPRM7 CRAPRM8
SUBMIT * END($$)
//CRA&PORT JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*
//RUN      EXEC PGM=IKJEFT01,DYNAMNBR=125,REGION=0M,TIME=NOLIMIT
//*
//* DD used by RAM
//TYPEMAP DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRATMAP)
//SHOWVIEW DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRASHOW)
//CRACFG DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRACFG)
//* uncomment CRABCFG and CRABSKEL to use batch actions
//*CRABCFG DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRABCFG)
//*CRABSKEL DD DISP=SHR,DSN=FEL.#CUST.CNTL
//* uncomment and provide correct DSN to use Package Ship
//*APIHJC DD DISP=SHR,DSN=#shipphjc
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD
//JCLOUT DD SYSOUT=(A,INTRDR),DCB=(LRECL=80,RECFM=F,BLKSIZE=80)
//EXT1ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT2ELM DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//EXT1DEP DD DISP=(NEW,DELETE),UNIT=SYSALLDA,
//          RECFM=VB,LRECL=4096,BLKSIZE=27998,SPACE=(TRK,(5,5))
//C1MSG1 DD SYSOUT(H)
//C1EXMSG DD SYSOUT(H)
//MSG3FILE DD DUMMY
//*
//* DD used by CARMA server (CRASERV)
//* pay attention to APF authorizations when using STEPLIB
//STEPLIB DD DISP=SHR,DSN=FEL.SFELLOAD
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//          DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
//CRADEF DD DISP=SHR,DSN=FEL.#CUST.CRADEF
//CRAMSG DD DISP=SHR,DSN=FEL.#CUST.CRAMSG
//CRASTRS DD DISP=SHR,DSN=FEL.#CUST.CRASTRS
//CARMALOG DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//*
//* DD used by ISPF (via NDVRC1)
//ISPPROF DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(1,1,5))
//ISPCTL0 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPCTL1 DD DISP=(NEW,DELETE,DELETE),UNIT=SYSALLDA,
//          LRECL=80,RECFM=FB,SPACE=(TRK,(5,5))
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSENU
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//*
//* DD used by TSO (IKJEFT01)
//SYSPROC DD DISP=SHR,DSN=FEL.SFELPROC
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
%CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1 &CRAPRM2 &STR(+))
&CRAPRM3 &STR(+)
&CRAPRM4 &STR(+)
&CRAPRM5 &STR(+)
&CRAPRM6 &STR(+)
&CRAPRM7 &STR(+)
&CRAPRM8 &STR(+) )
$$
EXIT CODE(0)

```

Figure 8. CRASUBCA : Soumission par lots avec le gestionnaire RAM CA Endevor® SCM

(Facultatif) Personnalisation additionnelle du gestionnaire RAM CA Endevor® SCM

Le gestionnaire RAM CA Endevor® SCM dispose de composants supplémentaires que vous pouvez personnaliser si nécessaire.

- Le traitement du démarrage CARMA comporte un exit utilisateur facultatif. Pour plus d'informations, voir «(Facultatif) Exit utilisateur CARMA», à la page 66.
- Le gestionnaire RAM CA Endevor® SCM comporte plusieurs fichiers de configuration, FEL.#CUST.PARMLIB(CRACFG), FEL.#CUST.PARMLIB(CRASHOW) et FEL.#CUST.PARMLIB(CRATMAP), que vous pouvez personnaliser. Pour plus d'informations, voir «CRACFG, CRASCL, CRASHOW et CRATMAP, les fichiers de configuration du gestionnaire RAM de CA Endevor® SCM», à la page 57.
- Le gestionnaire RAM CA Endevor® SCM dispose d'un exécutable d'allocation, FEL.SFELPROC(CRANDVRA), que vous pouvez personnaliser. Pour plus d'informations, voir «CRANDVRA, commande exec d'allocation initiale RAM CA Endevor® SCM», à la page 58.
- Le gestionnaire RAM CA Endevor® SCM prend en charge les actions CA Endevor® SCM en mode de traitement par lots. Les actions par lots requièrent un fichier de configuration, FEL.#CUST.PARMLIB(CRABCFG), un JCL de squelette, FEL.#CUST.CNTL(CRABATCA), et une carte de travail par défaut facultative, FEL.#CUST.CNTL(CRABJOBC), qui doivent être personnalisés. Pour plus d'informations, voir «Actions par lots du gestionnaire RAM CA Endevor® SCM», à la page 59.

Soumission d'un lot avec des exemples de gestionnaires RAM

Les informations de cette section expliquent comment configurer CARMA avec les spécifications suivantes :

- Démarrage du serveur : méthode de soumission par lots, qui nécessite des initiateurs JES
- Gestionnaire RAM : exemples de gestionnaire RAM, qui ne sont pas destinés à une utilisation à des fins de production

Cette étape de personnalisation peut être omise si vous voulez utiliser l'un des autres scénarios avec des spécifications différentes.

Création des fichiers VSAM

Personnalisez et envoyez les travaux JCL suivants pour définir et remplir les fichiers VSAM associés à CARMA. Pour des instructions de personnalisation, voir la documentation du membre. Les fichiers VSAM existants sont remplacés.

Pour plus de détails sur cette étape, voir «Fichiers CARMA VSAM», à la page 54 et «Gestionnaires RAM (Repository Access Managers) CARMA», à la page 54.

CARMA

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

Exemples de gestionnaire RAM

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

Personnalisation CRASRV.properties

Le serveur RSE utilise les paramètres de `/etc/zexpl/CRASRV.properties` pour démarrer et se connecter au serveur CARMA. Vous pouvez modifier le fichier à l'aide de la commande TSO **0EDIT**. Pour que les changements apportés prennent effet, vous devez redémarrer la tâche démarrée RSED.

Lorsque vous utilisez les emplacements de fichier par défaut, vous devez remplacer uniquement la valeur de la directive `clist.dsname` par `FEL.#CUST.CNTL(CRASUBMT)`. Pour plus d'informations sur les différentes directives, voir «CRASRV.properties, interface RSE avec CARMA», à la page 45.

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

Figure 9. CRASRV.properties : soumission d'un lot avec les exemples de gestionnaire RAM

Personnalisation de CRASUBMT

La liste de commandes `FEL.#CUST.CNTL(CRASUBMT)` et le JCL intégré envoient un serveur CARMA. Les modifications sont appliquées pour tous les serveurs CARMA démarrés après la mise à jour.

Pour des instructions de personnalisation, voir la documentation du membre. Pour plus d'informations sur la méthode de démarrage de soumission par lots, voir «CRASUB*, démarrage du serveur avec soumission par lots», à la page 52.

```

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

```

Figure 10. CRASUBMT : soumission d'un lot avec les exemples de gestionnaires RAM

(Facultatif) Personnalisation additionnelle des gestionnaires RAM personnalisés

Les gestionnaires RAM personnalisés possèdent des composants supplémentaires que vous pouvez personnaliser si nécessaire.

- Le traitement du démarrage CARMA comporte un exit utilisateur facultatif. Pour plus d'informations, voir (Facultatif) Exit utilisateur CARMA.
- Le démarrage du gestionnaire RAM personnalisé dispose d'un exécutable d'allocation, FEL.SFELPROC(CRAALLOC), que vous pouvez personnaliser. Pour plus d'informations, voir CRAALLOC, exec d'allocation du gestionnaire RAM personnalisé.

Informations de configuration CARMA

Les différents scénarios documentés dans cette publication ont de nombreux fichiers de configuration CARMA en commun. Les informations de ces fichiers de configuration sont documentés ici et référencés dans les scénarios.

CRASRV.properties, interface RSE avec CARMA

Le serveur CARMA fournit une interface API standard pour d'autres produits utilisant des systèmes hôte pour accéder à un ou plusieurs gestionnaires de configuration de logiciel (SCM). Cependant, il ne fournit pas de méthodes permettant la communication directe avec un ordinateur client. Pour cette communication, 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.

CRASRV.properties se trouve dans /etc/zexpl/, sauf si vous spécifiez un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). 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 : Pour que les changements apportés prennent effet, vous devez redémarrer la tâche démarrée RSED.

```
| # CRASRV.properties - CARMA configuration options
| #
| clist.dsname=''
| crastart.configuration.file=crastart.conf
| #port.start=0
| #port.range=100
| #user.exit='FEL.SFELSAMP(CRAEXIT) '
| #startup.script.name=carma.startup.rex
| #crastart.stub=CRASTART
| #crastart.syslog=Partial
| #crastart.timeout=420
| #crastart.steplib=FEL.SFELLPA
| #crastart.tasklib=TASKLIB
```

Figure 11. CRASRV.properties – Fichier de configuration de CARMA

clist.dsname

Définit la méthode de démarrage du serveur CARMA. Pour plus d'informations sur les différentes méthodes de démarrage, voir «Sélection de la méthode de démarrage du serveur et du gestionnaire RAM actif», à la page 34.

- *CRASTART indique que le serveur CARMA doit être démarré en tant que sous-tâche au sein de RSE à l'aide de CRASTART. Si vous spécifiez *CRASTART, vous devez également spécifier les directives crastart.* ou utiliser leurs valeurs par défaut.
- *ISPF indique que le serveur CARMA doit être démarré à l'aide de la passerelle client TSO/ISPF d'ISPF. Cette méthode de démarrage est obsolète.
- Toute autre valeur définit l'emplacement du CLIST CRASUBMT en utilisant les conventions de dénomination de style TSO. Lorsque des 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'ID utilisateur du 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 une chaîne nulle qui indique que CARMA n'est pas configuré.

crastart.configuration.file

Indique le nom du fichier de configuration CRASTART. La valeur par défaut est `crastart.conf`. Ce fichier indique les attributions de fichier et les appels de programme nécessaires pour démarrer un serveur CARMA. Cette directive est utilisée uniquement si la directive `clist.dsname` a la valeur `*CRASTART`. Le nom de fichier peut être indiqué de plusieurs manières :

- Chaîne nulle, ce qui signifie que la variable n'est pas spécifiée. La valeur par défaut est utilisée.
- Uniquement un nom de fichier, c'est-à-dire la méthode par défaut. CARMA parcourt votre répertoire de configuration (par défaut `/etc/zexp1`) pour rechercher le fichier.
- Chemin relatif, à savoir le répertoire et le nom du fichier, sans barre oblique (`/`) au début. CARMA ajoute votre répertoire de configuration (par défaut `/etc/zexp1/`) au chemin fourni pour que le chemin devienne absolu.
- Chemin absolu, à savoir le répertoire et le nom du fichier avec une barre oblique (`/`) au début. CARMA utilise l'emplacement de fichier spécifié.

#port.start

Lorsque le paramètre `port.start` a pour valeur 0 (zéro), CARMA utilise un port éphémère pour communiquer avec le serveur RSE. Dans ce scénario, TCP/IP affecte un numéro de port aléatoire libre. Lorsque le paramètre `port.start` a une valeur différente de zéro, il est interprété comme point de départ d'une plage de ports utilisée pour la communication entre CARMA et le serveur RSE, auquel cas la variable `port.range` doit également être définie. Le port par défaut est 0. Pour spécifier le début de la plage de ports, supprimez la mise en commentaire et effectuez la personnalisation. La communication sur ce port concerne uniquement le système 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" dans le document *Guide de référence de la configuration hôte (SC27-8578)*.

#port.range

Plage de ports, commençant par `port.start`, qui est utilisée pour la communication CARMA si la valeur du paramètre `port.start` est différente de zéro. La valeur par défaut est 100. Pour spécifier la taille de la plage de ports, supprimez la mise en commentaire et effectuez la personnalisation. Par exemple, lorsque le paramètre `port.start` a pour valeur 5227 et le paramètre `port.range` a pour valeur 100, les ports compris entre 5227 et 5326 (tous deux inclus) peuvent être utilisés par CARMA. Chaque connexion CARMA utilise un port exclusivement. La spécification d'une plage de ports limite donc le nombre maximal de sessions CARMA simultanées.

#user.exit

Définit le code spécifié par l'utilisateur à exécuter au démarrage de CARMA. Annulez la mise en commentaire de ce paramètre et spécifiez le nom de fichier du code à exécuter.

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'ID utilisateur du client, et non du préfixe TSO. Le second nom de fichier requiert que les utilisateurs CARMA gèrent leur propre code d'exit.

Un exemple d'exit utilisateur est fourni sous le nom FEL.SFELSAMP(CRAEXIT). Cet exemple décrit également les arguments de démarrage transmis à l'exit utilisateur. Pour plus d'informations, voir (Facultatif) Exit utilisateur CARMA.

startup.script.name

Définit le script de démarrage CARMA. La valeur par défaut est carma.startup.rex. Cette commande EXEC REXX déclenche le démarrage d'un serveur CARMA. Le nom de fichier peut être indiqué de plusieurs manières :

- Chaîne nulle, ce qui signifie que la variable n'est pas spécifiée. Dans ce cas, la valeur par défaut est utilisée.
- Uniquement un nom de fichier, c'est-à-dire la méthode par défaut. CARMA recherche les répertoires dans la variable d'environnement PATH pour trouver le fichier. Le répertoire contenant les exécutables Developer for z Systems (/usr/lpp/ibm/rdz/bin par défaut) est automatiquement ajouté à la variable d'environnement PATH.
- Chemin relatif, à savoir le répertoire et le nom du fichier, sans barre oblique (/) au début. CARMA ajoute votre répertoire de configuration (par défaut /etc/zexpl/) au chemin fourni pour que le chemin devienne absolu.
- Chemin absolu, à savoir le répertoire et le nom du fichier avec une barre oblique (/) au début. CARMA utilise l'emplacement de fichier spécifié.

#crastart.stub

Module de remplacement z/OS UNIX pour l'appel de CRASTART. La valeur par défaut est CRASTART. Ce module de remplacement rend le module de chargement CRASTART basé sur MVS disponible pour les processus z/OS UNIX. Pour indiquer un chemin spécifique, supprimez la mise en commentaire et appliquez la personnalisation. Cette directive est utilisée uniquement si la directive clist.dsname a la valeur *CRASTART. Le nom de fichier peut être indiqué de plusieurs manières :

- Chaîne nulle, ce qui signifie que la variable n'est pas spécifiée. La valeur par défaut est utilisée.
- Uniquement un nom de fichier, c'est-à-dire la méthode par défaut. CARMA recherche les répertoires dans la variable d'environnement PATH pour trouver le fichier. Le répertoire contenant les exécutables Developer for z Systems (/usr/lpp/ibm/rdz/bin par défaut) est automatiquement ajouté à la variable d'environnement PATH.
- Chemin relatif, à savoir le répertoire et le nom du fichier, sans barre oblique (/) au début. CARMA ajoute votre répertoire de configuration (par défaut /etc/zexpl/) au chemin fourni pour que le chemin devienne absolu.
- Chemin absolu, à savoir le répertoire et le nom du fichier avec une barre oblique (/) au début. CARMA utilise l'emplacement de fichier spécifié.

#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 valides sont répertoriées dans le tableau suivant.

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

Pour spécifier le niveau de détail requis pour les messages de journal système, supprimez la mise en commentaire et effectuez la personnalisation. Cette directive est utilisée uniquement si la directive `clist.dsname` a 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). Pour indiquer la valeur de dépassement du délai d'attente requise, supprimez la mise en commentaire et appliquez la personnalisation. Cette directive est utilisée uniquement si la directive `clist.dsname` a la valeur `*CRASTART`.

Remarque : L'arrêt système 522 pour le module CRASERV se produit si le paramètre JWT du membre parmlib SMFPRMxx a une valeur inférieure à la valeur `crastart.timeout` définie dans `CRASRV.properties`. Cette occurrence n'a pas d'incidence sur les opérations CARMA car le serveur est redémarré automatiquement si nécessaire.

#crastart.steplib

L'emplacement du module CRASTART lors d'un accès par l'intermédiaire de la directive STEPLIB de `rdz.env`. La valeur par défaut est `FEL.SFELLPA`. Si le module CRASTART ne peut pas faire partie de LPA ou LINKLIST, supprimez la mise en commentaire et personnalisez la directive. Des problèmes de contrôle de programme et d'APF risquent de se produire si le module CRASTART ne se trouve pas dans la LPA. Cette directive est utilisée uniquement si la directive `clist.dsname` a la valeur `*CRASTART`.

#crastart.tasklib

Autre nom pour le nom DD TASKLIB dans `crastart.conf`. La valeur par défaut est TASKLIB. Si le nom DD TASKLIB possède une signification spéciale pour votre SCM ou RAM et ne peut être utilisé en remplacement de STEPLIB, supprimez la mise en commentaire et personnalisez cette directive. Cette directive est utilisée uniquement si la directive `clist.dsname` a la valeur `*CRASTART`.

crastart*.conf, démarrage du serveur CRASTART

RSE démarre le module de chargement CRASTART, lequel utilise les définitions de `crastart*.conf` pour créer un environnement valide pour exécuter des commandes par lots TSO et ISPF. Rational Developer for z Systems utilise cet environnement pour exécuter le serveur CARMA, CRASERV.

`crastart*.conf` se trouve dans `/etc/zexpl/` si vous n'avez pas défini un emplacement différent lorsque vous avez personnalisé et envoyé le travail

FEL.SFELSAMP(FELSETUP). 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.

Developer for z Systems fournit plusieurs fichiers de configuration crastart*.conf. Chacun de ces exemples de fichiers est préconfiguré pour un scénario de personnalisation donné :

- crastart.endevor.conf est configuré pour le démarrage CRASTART avec le gestionnaire RAM CA Endevor® SCM.
- crastart.conf est configuré pour le démarrage CRASTART avec les exemples de gestionnaires RAM.

D'un point de vue conceptuel, le fichier crastart*.conf a la même fonction qu'un flux de travaux JCL, mais il plus restrictif.

- Les exemples suivants montrent des formats de lignes valides :
 - * comment
 - ddname=dsn1,dsn2,dsn3 * comment
 - ddname=SYSOUT(c) * comment
 - ddname=DUMMY * comment
 - -COMMAND=<any bpxwdyn command> * comment
 - PROGRAM = progname parms * comment

Remarque : La commande **BPXWDYN** est documentée dans le document *Using REXX and z/OS UNIX System Services* (SA22-7806) ; elle permet de créer des constructions d'allocations complexes.

- Toutes les données en entrée sont converties en majuscules.
- La continuation de ligne n'est pas prise en charge.
- Il n'existe aucune limite de longueur de ligne.
- Un ou plusieurs blancs sont autorisés autour du signe égal (=).
- Les allocations DD doivent précéder l'instruction PROGRAM associée.
- Les noms de définition de données (DD) alloués ici sont libérés à la fin de l'exécution du programme. Ils ne s'accumulent pas.
- Les noms DD alloués par les programmes appelés ne sont pas libérés.
- Plusieurs fichiers peuvent être concaténés pour former un nom DD. Les noms de fichier peuvent être séparés par une virgule (,) et la recherche porte sur la concaténation dans l'ordre indiqué.
- Toutes les allocations de fichiers sont effectuées avec DISP=SHR, sauf pour les allocations effectuées à l'aide de -COMMAND.
- Les données linéaires ne sont pas prises en charge. Toutes les données doivent se trouver dans des fichiers catalogués.
- Les variables peuvent être utilisées uniquement à droite du signe égal (=).
- Les variables suivantes sont prises en charge :

&CRAUSER.	ID utilisateur client
&CRADATE.	Date en cours au format Daaaajjj (julien 7 caractères)
&CRATIME.	Heure en cours au format Thhmmss (heures minutes secondes)
&CRAPRM1.	Numéro de port

System symbol	Tout symbole système SYS1.PARMLIB(IEASYMxx)
-<ddname>	Tiret (-) suivi du nom DD déjà défini qui fait office de référence amont *.ddname dans le JCL. La DD d'origine doit être allouée en utilisant 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. Si vous avez besoin du préfixe TSO ou d'une autre variable qui n'est pas disponible, voir «(Facultatif) Exec d'allocation personnalisé», à la page 65.

La figure 12 montre un crastart*.conf de base qui contient les services ISPF.

```
* DD used by RAM

* DD used by CARMA server (CRASERV)
TASKLIB = FEL.SFELLOAD
CRADEF  = FEL.#CUST.CRADEF
CRAMSG  = FEL.#CUST.CRAMSG
CRASTRS = FEL.#CUST.CRASTRS
CARMALOG = SYSOUT(H)
SYSPRINT= SYSOUT(H)

* DD used by ISPF (ISPSTART)
-COMMAND=ALLOC FI(ISPCTL0) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPCTL1) NEW DELETE DSORG(PS) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA)
-COMMAND=ALLOC FI(ISPPROF) NEW DELETE DSORG(PO) RECFM(F,B) LRECL(80)
  BLKSIZE(32720) SPACE(5,5) TRACKS UNIT(SYSALLDA) DIR(5)
ISPTABL = -ISPPROF
ISPTLIB = -ISPPROF,ISP.SISPTENU
ISPMLIB = ISP.SISPMENU
ISPPLIB = ISP.SISPPENU
ISPSLIB = ISP.SISPSENU

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

* CRAALLOC

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

Figure 12. crastart*.conf : démarrage du serveur CARMA en utilisant CRASTART

Remarque :

- Compte tenu des limitations de largeur de page, certaines lignes de l'exemple sont renvoyées à la ligne. Toutes les lignes qui sont mises en retrait doivent être ajoutées à la fin de la ligne précédente.
- Si vous modifiez la ligne PROGRAM, assurez-vous qu'il y ait au moins un espace avant la parenthèse fermante (") de l'instruction PARM() pour simplifier le traitement de la chaîne.
- 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.

- Le nom DD TASKLIB fait office de STEPLIB dans le JCL. Son nom DD doit correspondre à la valeur définie pour crastart.tasklib dans CRASRV.properties qui est décrit dans «CRASRV.properties, interface RSE avec CARMA», à la page 45.
- Des règles APF standard s'appliquent aux allocations TASKLIB. Les bibliothèques perdent leur autorisation APF lorsqu'une bibliothèque autorisée non-APF fait partie de la concaténation.
- L'arrêt système 522 pour le module CRASERV se produit si le paramètre JWT du membre parmlib SMFPRMxx a une valeur inférieure à la valeur crastart.timeout définie dans CRASRV.properties. Ceci n'a pas d'incidence sur les opérations CARMA car le serveur est redémarré automatiquement, si nécessaire.
- Les détails du processus de démarrage CARMA se trouvent dans rsecomm.log à la fin de l'exécution du serveur. Pour plus d'informations sur la définition du niveau de détail de rsecomm.log, voir le *Guide de configuration hôte d'IBM Explorer for z/OS (SC27-8437)*.

Collecte des fichiers journaux CRASTART

CRASTART crée un environnement TSO sous la forme d'un processus enfant de RSE qui s'exécute dans un espace adresse différent. Des actions significatives peuvent être nécessaires pour conserver le résultat CARMA envoyé à SYSOUT(*), ce qui complique la collecte des fichiers journaux. Vous pouvez résoudre ce problème en écrivant les fichiers journaux dans un fichier spécifique à l'utilisateur, comme indiqué dans l'exemple d'allocation suivant :

```
-COMMAND=ALLOC FI(CARMALOG) MOD CATALOG DSORG(PS) RECFM(F,B) LRECL(133)
          BLKSIZE(27930) SPACE(5,5) TRACKS UNIT(SYSALLDA)
          DA(&CRAUSER..&SYSNAME..CRA.CARMALOG)
```

Remarque :

- Compte tenu des limitations de largeur de page, certaines lignes de l'exemple sont renvoyées à la ligne. Toutes les lignes qui sont mises en retrait doivent être ajoutées à la fin de la ligne précédente.
- Pour pouvoir créer des fichiers journaux spécifiques à l'utilisateur, ce fichier journal doit être alloué via l'instruction -COMMAND.
- Vous pouvez également allouer les fichiers journaux dans un exec d'allocation pour bénéficier d'une plus grande souplesse ; par exemple, envoyer le fichier uniquement à un fichier pour des utilisateurs spécifiques. Pour plus d'informations sur les exec d'allocation, voir «(Facultatif) Exec d'allocation personnalisé», à la page 65.

Si vous écrivez des fichiers journaux dans SYSOUT, n'oubliez pas que le SYSOUT alloué par les processus z/OS UNIX est traité comme un résultat spécial dans JES. Cela revient au même que SYSOUT alloué par des transactions APPC.

- Alors que le serveur CARMA est encore actif, vous pouvez afficher le résultat à l'aide de la commande **DA** dans SDSF. Pour ce travail, l'ID utilisateur sera suivi d'un chiffre aléatoire comme nom de travail et d'un ID de travail STC. L'utilisateur est le propriétaire de travail.
- Si la sortie a été écrite dans une classe de sortie HOLD, lorsque le serveur CARMA s'arrête, en raison d'une période d'inactivité ou si l'utilisateur met fin à la connexion, vous pouvez afficher le résultat à l'aide des commandes **APPC ON** et **H ALL** dans SDSF. Le nom, l'ID et le propriétaire du travail ne changent pas. Chaque DD présente un fichier spoule distinct, qui n'indique pas de quel DD il s'agit.
- Le gestionnaire de travaux JES peut également afficher la sortie si SEARCHALL=ON est actif dans FEJCNFG et si la sortie réside sur le spoule dans une classe de

| sortie HOLD. Pour plus d'informations sur la directive SEARCHALL, voir le manuel
| *IBM Explorer for z/OS - Guide de configuration hôte (SC27-8437)*..

CRASUB*, démarrage du serveur avec soumission par lots

RSE démarre la liste de commandes (CLIST) CRASUB*, qui à son tour soumet un JCL imbriqué afin de créer un environnement valide pour l'exécution de commandes TSO et ISPF par lots. Rational Developer for z Systems utilise cet environnement pour exécuter le serveur CARMA, CRASERV.

CRASUB se trouve dans FEL.#CUST.CNTL à moins que vous n'ayez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

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

Developer for z Systems fournit plusieurs travaux JCL CRASUB*. Chacun de ces exemples de fichiers est préconfiguré pour un scénario de personnalisation donné :

- CRASUBCA est configuré pour démarrer un traitement par lots avec le gestionnaire RAM CA Endevor® SCM.
- CRASUBMT est configuré pour le démarrer un traitement par lots avec les exemples de gestionnaire RAM.

La figure 13, à la page 53 affiche un squelette CRASUB* de base qui inclut les services ISPF.


```

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

```

Figure 13. CRASUB* : démarrage CARMA en utilisant la soumission de lot

Remarque :

- Si vous modifiez les données SYSTSIN, assurez-vous qu'il y ait au moins un espace avant la parenthèse fermante (") de l'instruction PARM() pour simplifier le traitement de la chaîne.
- 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 annulant la mise en commentaire et en modifiant la ligne SET CRAPRM2=420 dans la liste de commandes CRASUB*. Cette valeur correspond au nombre de secondes que CARMA attend 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 à la fin de l'exécution du serveur. Pour plus d'informations sur la définition du niveau de détail de rsecomm.log, voir le *Guide de configuration hôte d'IBM Explorer for z/OS (SC27-8437)*.

Fichiers CARMA VSAM

Le serveur CARMA doit avoir un accès en LECTURE aux trois fichiers VSAM. Les exemples de membres permettant de créer et remplir ces fichiers VSAM se trouvent dans FEL.#CUST.JCL si vous n'avez pas défini un emplacement différent lorsque vous avez personnalisé et envoyé le travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Remarque :

- Si vous souhaitez fusionner les définitions d'un gestionnaire RAM (personnalisé) dans une configuration VSAM existante, consultez l'exemple de travail FEL.#CUST.JCL(CRA#UADD). Ce travail doit être personnalisé et soumis pour chaque VSAM CARMA modifié. Pour plus d'informations sur la structure d'enregistrement utilisée par les différents fichiers de méthode d'accès VSAM CARMA, voir *Common Access Repository Manager Developer's Guide* (SC23-7660).
- Utilisez l'exemple de travail FEL.#CUST.JCL(CRA#UQRY) pour extraire les définitions actives d'une méthode d'accès VSAM dans un fichier séquentiel.

CRADEF, fichier de configuration

Ce fichier VSAM décrit les fonctions prises en charge par les gestionnaires RAM définis. Les développeurs RAM doivent disposer d'un accès en MISE A JOUR à ce fichier. Le fichier peut être créé par l'un des exemples de travaux suivants :

- CRA\$VCAD remplit le fichier avec des données RAM CA Endevor® SCM.
- CRA\$VDEF remplit le fichier avec des exemples de données RAM.

Les exemples de tâches peuvent être utilisés pour désactiver un gestionnaire RAM défini durant la création de VSAM. Vous pouvez ainsi créer une configuration CARMA personnalisée à l'aide d'un fichier d'entrée principal unique, pouvant être fourni par IBM ou personnalisé par vos développeurs RAM.

CRAMSG, fichier de messages

Ce fichier VSAM contient les messages émis par le serveur CARMA. Le fichier peut être créé par l'un des exemples de travaux suivants :

- CRA\$VMSG remplit le fichier avec des données de serveur génériques.

CRASTRS, fichier de chaînes personnalisées

Ce fichier VSAM contient les messages émis par les gestionnaires RAM définis. Les développeurs RAM doivent disposer d'un accès en MISE A JOUR à ce fichier. Le fichier peut être créé par l'un des exemples de travaux suivants :

- CRA\$VCAS remplit le fichier avec des données RAM CA Endevor® SCM.
- CRA\$VSTR remplit le fichier avec des exemples de données RAM.

Gestionnaires RAM (Repository Access Managers) CARMA

Un gestionnaire RAM (Repository Access Manager) est une API (Application Programming Interface) pour un gestionnaire SCM (Software Configuration Manager) z/OS. De la même manière, Developer for z Systems ou les applications écrites par l'utilisateur peuvent démarrer un serveur CARMA qui charge les gestionnaires RAM et fournit une interface standard pour accéder au gestionnaire SCM pris en charge.

Le serveur CARMA doit pouvoir trouver les modules chargeables RAM via LINKLIST ou STEPLIB/TASKLIB.

Les modules chargeables RAM CRAR* RAM qui sont fournis par Developer for z Systems se trouvent dans FEL.SFELLOAD, l'exemple de code source et les travaux de compilation se trouvant dans FEL.SFELSAMP, sauf si vous avez utilisé un autre qualificatif de haut niveau pendant l'installation SMP/E de Developer for z Systems.

Les sections suivantes contiennent des remarques de personnalisation pour les gestionnaires RAM disponibles avec Developer for z Systems. Les exemples de membres référencés se trouvent dans FEL.#CUST.* si vous n'avez pas défini un emplacement différent lorsque vous avez personnalisé et envoyé l'exemple de travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Pour connaître CARMA de manière plus approfondie et obtenir plus d'informations sur les exemples de RAM et de code source fournis, voir *Common Access Repository Manager Developer's Guide* (SC23-7660).

Gestionnaire RAM SCM CA Endeavor®

- Le gestionnaire RAM CA Endeavor® SCM est un gestionnaire RAM de type production.
- Le gestionnaire RAM CA Endeavor® SCM permet aux clients Developer for z Systems d'accéder directement aux éléments SCM CA Endeavor®.
- Le module chargeable s'appelle CRARNDVR.
- Le gestionnaire RAM CA Endeavor® SCM dispose de nombreux autres paramètres comparé à une configuration CARMA classique. Utilisez l'une des installations préconfigurées qui prennent en charge le gestionnaire RAM SCM CA Endeavor® comme point de départ et personnalisez-la davantage en fonction de vos besoins.
- Le gestionnaire RAM CA Endeavor® SCM dispose de plusieurs fichiers de configuration que vous pouvez personnaliser. Pour plus d'informations, voir «CRACFG, CRASCL, CRASHOW et CRATMAP, les fichiers de configuration du gestionnaire RAM de CA Endeavor® SCM», à la page 57.
- Le gestionnaire RAM CA Endeavor® SCM dispose d'un exécutable d'allocation, FEL.SFELPROC(CRANDVRA), que vous pouvez personnaliser. Pour plus d'informations, voir «CRANDVRA, commande exec d'allocation initiale RAM CA Endeavor® SCM», à la page 58.
- Le gestionnaire RAM CA Endeavor® SCM prend en charge les actions CA Endeavor® SCM en mode de traitement par lots, en arrière-plan. Pour plus d'informations, voir «Actions par lots du gestionnaire RAM CA Endeavor® SCM», à la page 59.

Gestionnaire RAM des modules CA Endeavor® SCM

- Le gestionnaire RAM des modules CA Endeavor® SCM est un gestionnaire RAM de type production.
- Le gestionnaire RAM CA Endeavor® SCM permet aux clients Developer for z Systems d'accéder directement aux modules CA Endeavor® SCM.
- Le module chargeable s'appelle CRARPKGS.
- Le gestionnaire RAM des modules CA Endeavor® SCM ne possède pas de paramètres personnalisables et doit être utilisé en association avec le gestionnaire RAM CA Endeavor® SCM.

Gestionnaire RAM PDS

- Le gestionnaire RAM PDS est un exemple. Ne l'utilisez pas dans un environnement de production.

- Le gestionnaire RAM PDS donne une liste de fichiers analogues aux fichiers **MVS -> Mes fichiers** de la vue Systèmes distants.
- Le module chargeable s'appelle CRARPDS.
- Le gestionnaire RAM PDS requiert la disponibilité des services ISPF.
- Le gestionnaire RAM PDS nécessite qu'un fichier de données VSAM supplémentaire soit alloué à la DD CRARAM1. Ce fichier VSAM peut être alloué et amorcé avec l'exemple de travail FEL.#CUST.JCL(CRA#VPDS). Pour des instructions de personnalisation, voir la documentation du membre.
- Le code source et les travaux de compilation sont disponibles dans FEL.SFELSAMP. Voir *Common Access Repository Manager Developer's Guide* (SC23-7660) pour plus d'informations.

Gestionnaire Skeleton

- Le gestionnaire RAM skeleton est un exemple. Ne l'utilisez pas dans un environnement de production.
- Le gestionnaire RAM Skeleton fournit une infrastructure fonctionnelle que vous pouvez utiliser comme point de départ pour développer votre propre gestionnaire RAM.
- Le module chargeable s'appelle CRARTEST.
- Le code source et les travaux de compilation sont disponibles dans FEL.SFELSAMP. Voir *Common Access Repository Manager Developer's Guide* (SC23-7660) pour plus d'informations.

Gestionnaire SCLM

- Le gestionnaire RAM SCLM est un exemple. Ne l'utilisez pas dans un environnement de production.
- Le gestionnaire RAM SCLM fournit une entrée de base dans SCLM, le gestionnaire de configuration de logiciel d'ISPF. Ce gestionnaire RAM n'est pas activé par défaut.
- Le module chargeable s'appelle CRARSCLM.
- Le gestionnaire RAM SCLM a besoin des services ISPF pour être disponible.
- Le gestionnaire RAM SCLM nécessite qu'un fichier de données VSAM supplémentaire soit alloué à la DD CRARAM2. Ce fichier VSAM peut être alloué et amorcé avec l'exemple de travail FEL.#CUST.JCL(CRA#VSLM). Pour des instructions de personnalisation, voir la documentation du membre.
- Le gestionnaire RAM SCLM a besoin des différents fichiers spécifiques à l'utilisateur pour exister. Personnalisez FEL.#CUST.JCL(CRA#ASLM) pour allouer ces fichiers. Pour des instructions de personnalisation, voir la documentation du membre. Chaque utilisateur doit envoyer CRA#ASLM une fois pour pouvoir utiliser CARMA avec le gestionnaire SCLM. Sinon, une erreur d'attribution est engendrée.
- Le gestionnaire RAM SCLM n'est pas activé par défaut. Pour activer le gestionnaire, il doit être défini dans les fichiers CARMA VSAM référencés par DD CRADEF et CRASTRS. Utilisez l'exemple de travail FEL.#CUST.JCL(CRA#UADD) pour fusionner FEL.SFELVSM2(CRA0SLMD) avec CRADEF et FEL.SFELVSM2(CRA0SLMS) avec CRASTRS. Pour des instructions de personnalisation, voir la documentation du membre.
- Le code source et les travaux de compilation sont disponibles dans FEL.SFELSAMP. Voir *Common Access Repository Manager Developer's Guide* (SC23-7660) pour plus d'informations.

CRACFG, CRASCL, CRASHOW et CRATMAP, les fichiers de configuration du gestionnaire RAM de CA Endevor® SCM

Les composants CARMA RAM CA Endevor SCM suivants peuvent être personnalisés, quelle que soit la méthode de démarrage de serveur que vous avez choisie. Les exemples de membre référencés ci-après se trouvent dans FEL.#CUST.PARMLIB, sauf si vous avez indiqué un emplacement différent lorsque vous avez personnalisé et soumis le travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

CRACFG, interaction du gestionnaire RAM CA Endevor SCM avec le serveur SCM

CRACFG indique comment le gestionnaire RAM CA Endevor SCM interagit avec CA Endevor SCM. Voir la documentation dans le membre pour les instructions de personnalisation si vous souhaitez changer les paramètres par défaut.

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

Figure 14. CRACFG - Interaction du gestionnaire RAM CA Endevor SCM avec le serveur SCM

CRASCL, modèle SCL RAM CA Endevor® SCM

CRASCL est un modèle SCL (Software Control Language) qui permet de limiter les actions et les options qui sont autorisées lorsque des modules CA Endevor® SCM sont traités.

Lorsqu'il est utilisé, seules les actions et options explicitement répertoriées dans le modèle SCL sont autorisées à être spécifiées dans Developer for z Systems Packages Editor. Voir la documentation dans le membre pour les instructions de personnalisation si vous souhaitez changer les paramètres par défaut.

CRASHOW, filtres par défaut RAM CA Endevor® SCM

CRASHOW définit des filtres par défaut pour des environnements CA Endevor® SCM, des systèmes, etc. Voir la documentation dans le membre pour les instructions de personnalisation si vous souhaitez changer les paramètres par défaut.

```
ENV=*
TOENV=
STGID=*
TOSTGID=
SYS=*
SUBSYS=*
ELEM=*
TOELEM=
TYPE=*
#FILTER-DEP=YES
```

Figure 15. CRASHOW - Filtres par défaut RAM CA Endevor® SCM

Remarque : FILTER-DEP n'est pas une variable CA Endevor® SCM courante, mais une variable propre à Developer for z Systems qui contrôle les analyses de dépendance des éléments comportant des références d'encombrement à d'autres emplacements de référentiel CA Endevor® SCM.

CRATMAP, mappages d'extension de fichier RAM CA Endeavor® SCM

CRATMAP substitue un type CA Endeavor® SCM aux mappages d'extension de fichier. Si vous souhaitez changer les valeurs par défaut, voir les instructions de personnalisation de la documentation du membre.

```
# *      = cbl
# COBOL  = cbl
# COPY   = cpy
# ASM    = asm
# MACRO   = asm
# PROCESS = jcl
```

Figure 16. CRATMAP : Filtres par défaut RAM CA Endeavor® SCM

CRANDVRA et CRADYNDA, commandes exec d'allocation RAM CA Endeavor® SCM

Les composants CARMA RAM CA Endeavor SCM suivants peuvent être personnalisés, quelle que soit la méthode de démarrage de serveur que vous avez choisie.

Vous pouvez personnaliser une copie de ces commandes exec REXX d'allocation REXX si certaines valeurs par défaut comme le nom du fichier, ne correspondent pas aux normes établies pour votre site. Les commandes exec se trouvent dans FEL.SFELPROC, sauf si vous avez utilisé un qualificatif de haut niveau différent lors de l'installation SMP/E de Developer for z Systems.

Pour des instructions de personnalisation, voir la documentation du membre. Pour plus d'informations sur les exec d'allocation, voir «(Facultatif) Exec d'allocation personnalisé», à la page 65.

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 SFELPROC dans la définition de données SYSEXEC DD de la méthode de démarrage CARMA choisie par vos soins afin que le nouveau nom de fichier puisse correspondre.

CRANDVRA, commande exec d'allocation initiale RAM CA Endeavor® SCM

Les deux méthodes de démarrage (soumission par lots et CRASTART) appellent la commande exec REXX CRANDVRA pour allouer des fichiers spécifiques de l'utilisateur utilisés par le RAM CA Endeavor® SCM. Les allocations sont effectués dans un exc distinct, car un exec offre plus de souplesse que dans la soumission par lots CRASUBCA JCL et le fichier de configuration CRASTART crastart.endeavor.conf. L'exécutable d'allocation est également chargé d'appeler l'exit utilisateur facultatif.

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

DD	Nom du fichier	Type
ENHCEDIT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDIT	Temporaire
ENHCEDITV	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDITV	Temporaire
C1PRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	Temporaire
SPCLLIST	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.SPCLLIST	Temporaire
PKGSCLS	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.PKGSCLS	Temporaire
CRABJCLO	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRABJCLO	Temporaire
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRAPARM	Temporaire

CRADYNDA, commande exec de réallocation RAM CA Endeavor® SCM

CA Endeavor® SCM peut fonctionner avec des fichiers bloqués variables avec diverses longueurs d'enregistrement et exige que Developer for z Systems utilise des allocations identiques pour certaines actions. Les deux méthodes de soumission par lots et de démarrage CRASTART appellent la commande exec CRADYNDA REXX pour allouer à un fichier de travail les bons attributs.

Actions par lots du gestionnaire RAM CA Endeavor® SCM

En principe, les actions CA Endeavor® SCM telles que "Generate Element" sont exécutées "en ligne", dans l'espace d'adresse du serveur CARMA. Ce comportement peut engendrer des problèmes si vos procédures CA Endeavor® SCM appelle TSO, lequel est déjà actif, ce qui signifie que les DD requis (SYSTSIN et SYSTSPRT, par exemple) son en cours d'utilisation.

Pour résoudre ce problème, le gestionnaire RAM CA Endeavor® SCM prend en charge les "actions par lots". Lorsque les actions par lots sont activées, le gestionnaire RAM CA Endeavor® SCM soumet un travail par lots personnalisable pour exécuter des actions telles que "Generate Element". L'utilisation d'un travail par lots se traduit par l'allocation de DD tels que SYSTSIN et SYSTSPRT par vos procédures CA Endeavor® SCM, le JCL soumis ne demandant pas que le TSO soit actif.

Les actions par lots du gestionnaire RAM CA Endeavor® SCM sont l'équivalent au niveau de Developer for z Systems des actions d'arrière-plan CA Endeavor® SCM.

Lorsqu'une requête est émise pour exécuter une action prise en charge par des actions par lots, le gestionnaire RAM CA Endeavor® SCM vérifie l'existence du DD CRABCFG dans CRASUBCA ou crastart.endeavor.conf, ainsi que la validité de la configuration de DD. Si CRABCFG existe et que la configuration est valide, l'action est exécutée dans le lot. Si CRABCFG n'existe pas, l'action est exécutée en ligne. Les clients Developer for z Systems ont la possibilité de redéfinir ce comportement.

Par exemple :

```
/* uncomment CRABCFG and CRABSKEL to use batch actions
/*CRABCFG DD DISP=SHR,DSN=FEL.#CUST.PARMLIB(CRABCFG)
/*CRABSKEL DD DISP=SHR,DSN=FEL.#CUST.CNTL
```

Remarque :

- L'environnement sans TSO est disponible uniquement pour les actions du gestionnaire RAM CA Endeavor® SCM sélectionnées. Les actions par lots ne prennent pas en charge un environnement sans TSO hors de ce domaine d'application.
- Le fichier de configuration CRABCFG indique les actions CA Endeavor® SCM prises en charge.
- Un exemple de travail fonctionnel, FEL.#CUST.CNTL(CRABATCA), est fourni pour exécuter les actions par lots, mais vous êtes censé personnaliser cet exemple afin de démarrer vos procédures CA Endeavor® SCM en cours.
- Vérifiez que le nombre d'initiateurs JES disponibles est suffisant dans la classe utilisée pour soumettre les JCL d'actions par lots.
- Lors de l'utilisation de JES dans un environnement SYSPLEX, vérifiez que le travail s'exécute sur le système en cours ou que les informations d'exécution sont réacheminées vers le système qui héberge Developer for z Systems, de sorte que le gestionnaire RAM CA Endeavor® SCM puisse en vérifier l'état.
- Le client Developer for z Systems peut fournir une carte de travail personnalisée et des instructions JCL supplémentaires au JCL d'actions par lots avant toute soumission.

CRABCFG, configuration d'action par lots CA Endeavor® SCM RAM

CRABCFG définit les variables de configuration liées aux actions par lots CA Endeavor® SCM RAM.

CRABCFG se trouve dans FEL.#CUST.PARMLIB à moins que vous n'ayez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Consultez l'exemple de fichier CRABCFG 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 commentaires qui suivent une directive et sa valeur affectée sont pris en charge. Les espaces autour du signe égal (=) le sont également. La continuation de ligne n'est pas prise en charge.

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


```

# Location of batch action JCL
SKELETON-DD = CRABSKEL
#
# batch action JCL members within SKELETON-DD
DEFAULT-JOBCARD = CRABJOBC
ADD-ELEMENT     = CRABATCA
DELETE-ELEMENT  = CRABATCA
GENERATE-ELEMENT = CRABATCA
MOVE-ELEMENT    = CRABATCA
RETRIEVE-ELEMENT = CRABATCA
PRINT-ELEMENT   = CRABATCA
PRINT-MEMBER    = CRABATCA
SIGNIN-ELEMENT  = CRABATCA
TRANSFER-ELEMENT = CRABATCA
#
# Command substitution key within batch action JCL
BSTIPT01-KEY = <CRA_BSTIPT01>

```

Figure 17. CRABCFG : Configuration d'actions par lots CA Endeavor® SCM RAM

SKELETON-DD

Nom de l'instruction DD faisant référence à un ou plusieurs fichiers PDS(E) détenant les JCL de squelette d'action par lots. L'exemple de valeur est CRABSKEL. Peut être modifiée si nécessaire. Ce DD doit être défini dans le serveur CARMA dans CRASUBCA ou crastart.endeavor.conf.

DEFAULT-JOBCARD

Nom du membre qui détient une carte de travail par défaut. Si elle n'est pas ignorée par une carte de travail spécifique à un utilisateur et stockée sur un client Developer for z Systems, cette carte de travail par défaut remplace la clé <JOBCARD> dans le code JCL d'un squelette. Peut être modifié si nécessaire.

GENERATE-ELEMENT et autres actions CA Endeavor® SCM

Les noms de clé représentent les actions CA Endeavor® SCM prises en charge par l'action par lots et ne peuvent pas être changés. La valeur attribuée à chaque clé est le nom de membre du JCL de squelette correspondant. L'exemple de valeur est CRABATCA pour toutes les clés. Peut être modifié si nécessaire.

BSTIPT01-KEY

Clé de substitution de la chaîne de commande CA Endeavor® SCM en cours. L'exemple de valeur est <CRA_BSTIPT01>. Peut être modifié si nécessaire. La première occurrence, ne figurant pas dans un commentaire, de cette clé de substitution dans le JCL de squelette est remplacée par la chaîne de commande qui indique à CA Endeavor® SCM d'exécuter l'action demandée en fonction de l'élément demandé.

CRABATCA, JCL d'action par lots du RAM CA Endeavor® SCM

CRABATCA est un exemple de JCL de squelette utilisé pour l'action par lots. Pour changer les valeurs par défaut, voir les instructions de personnalisation de la documentation du membre.

CRABATCA se trouve dans FEL.#CUST.CNTL à moins que vous n'ayez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

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

```

//<JOB CARD>
//*
//CRABATCA EXEC PGM=NDVRC1,DYNAMNBR=1500,REGION=4096K,PARM='C1BM3000'
//STEPLIB DD DISP=SHR,DSN=CA.NDVR.CSIQAUTU
// DD DISP=SHR,DSN=CA.NDVR.CSIQAUTH
//CONLIB DD DISP=SHR,DSN=CA.NDVR.CSIQLOAD
//C1MSG1 DD SYSOUT=*
//C1MSG2 DD SYSOUT=*
//C1PRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYMDUMP DD DUMMY
//SYSIN DD DUMMY
//BSTIPT01 DD *
SET STOPRC 16 .
<CRA_BSTIPT01>
//*

```

Figure 18. JCL d'action par lots CRABATCA : CA Endeavor® SCM RAM

CRABJOBC, carte de travail d'actions par lots du RAM CA Endeavor® SCM

CRABJOBC représente un exemple de carte de travail par défaut utilisé pour le JCL de squelette d'actions par lots qui définit la clé <JOB CARD>. Pour changer les valeurs par défaut, voir les instructions de personnalisation de la documentation du membre.

CRABJOBC se trouve dans FEL.#CUST.CNTL à moins que vous n'ayez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

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

```

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

```

Figure 19. CRABJOBC : carte de travail d'actions par lots du RAM CA Endeavor® SCM

CRAALLOC, exec d'allocation du gestionnaire RAM personnalisé

Les deux méthodes de démarrage (soumission par lots et CRASTART) appellent la commande exec REXX CRAALLOC REXX pour allouer des fichiers spécifiques à l'utilisateur qui peuvent être utilisés par un gestionnaire RAM écrit par l'utilisateur. Les allocations sont effectuées dans un exec distinct, car un exec offre plus de souplesse que dans le JCL de soumission par lots CRASUBMT et le fichier de configuration CRASTART crastart.conf. L'exécutable d'allocation est également chargé d'appeler l'exit utilisateur facultatif.

DD	Nom du fichier	Type
CRAPARM	&SYSPREF.&SYSUID.. &SYSNAME..CRA\$CUST.CRAPARM	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. CRAALLOC se trouve dans FEL.SFELPROC si vous avez utilisé un qualificatif de haut niveau différent pendant l'installation SMP/E de Rational Developer for z Systems.

Pour des instructions de personnalisation, voir la documentation du membre. Pour plus d'informations sur les exec d'allocation, voir (Facultatif) Exec d'allocation personnalisé.

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 SFELPROC dans la définition de données SYSEXEC de la méthode de démarrage CARMA choisie afin que le nouveau nom de fichier puisse correspondre.

Codes retour CARMA

CARMA peut signaler différents codes d'erreur au client ou dans les journaux du système hôte. Les détails fournis avec l'erreur ainsi que les informations contenues dans le tableau 12 vous aident à localiser l'erreur et à la corriger.

Tableau 12. Codes retour CARMA

Plage d'erreur	Type d'erreur
4-99	Erreurs CARMA génériques
100-199	Erreurs RAM génériques
200-399	Erreurs CRASERV (serveur CARMA)
400-499	Erreurs RSE (outil d'exploration CARMA)
500-899	Erreurs spécifiques au gestionnaire RAM
900-999	Erreurs TSO et TCP/IP

Codes retours courants :

- 220 : le serveur CARMA s'arrête en raison d'un délai d'attente d'inactivité. Il ne s'agit pas d'une erreur.
- 990 : le serveur CARMA ne peut pas se connecter au port sur lequel l'outil d'exploration CARMA écoute.

(Facultatif) Prise en charge de plusieurs RAM

CARMA offre la possibilité de définir plusieurs gestionnaires RAM et de 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 sont chargés sur le serveur CARMA.

Parce que les gestionnaires RAM sont actifs comme des plug-in du serveur CARMA, assurez-vous que tous les éléments prérequis, comme les allocations de fichiers, pour chaque RAM sont disponibles dans l'espace adresse du serveur

CARMA. Cette condition requise peut nécessiter des changements dans les exemples de configuration CARMA, comme CRASUBMT ou crastart.conf, qui sont fournis avec Developer for z Systems.

Exemple

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

Définition pour le gestionnaire RAM CA Endevor® SCM :

- FEL.SFELVSM2(CRA0VCAD) : définitions CRADEF
- FEL.SFELVSM2(CRA0VCAS) : définitions CRASTRS
- /etc/zexpl/crastart.endevor.conf : fichier de configuration CRASTART

Définitions pour le gestionnaire RAM PDS :

- FEL.SFELVSM2(CRA0VDEF) : définitions CRADEF
- FEL.SFELVSM2(CRA0VSTR) : définitions CRASTRS
- FEL.#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 gestionnaire RAM PDS à partir des membres SFELVSM2. Ces membres détiennent des informations pour tous les modèles de gestionnaire RAM et pas seulement le RAM PDS.
2. Fusionnez ces données avec les membres SFELVSM2 du gestionnaire RAM CA Endevor® SCM.
3. Créez une liste d'éléments prérequis spécifiques du gestionnaire RAM PDS :
 - DD CRARAM1, lié à FEL.#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 gestionnaires 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 et CRASTRS. La VSAM CRAMSG est spécifique du serveur CARMA et est identique pour les deux gestionnaires RAM.
2. Ajoutez une définition CRARAM1 à crastart.endevor.conf :
CRARAM1 = FEL.#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 gestionnaires RAM.
4. 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 du gestionnaire RAM CA Endevor® SCM.

- %CRANDVRA : exec de préallocation du gestionnaire RAM CA Endevor® SCM, situé dans FEL.SFELPROC, qui alloue des fichiers de travail utilisateur temporaires.
- NDVRC1 : CA Endevor® dorsal 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.

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

(Facultatif) Exec d'allocation personnalisé

Toutes les méthodes de démarrage du serveur CARMA ont des limitations en terme d'allocation de fichier. Par exemple, la substitution de préfixe n'est pas disponible dans JCL ni CRASTART.

Toutefois, en créant un exec appelé après le démarrage de TSO ou d'ISPF, et avant le démarrage de CARMA, vous pouvez utiliser toutes les variables et tous les services disponibles dans TSO ou ISPF pour effectuer les allocations requises.

Rational Developer for z Systems utilise un exécutable d'allocation dans chacune des installations préconfigurées décrites précédemment dans ce chapitre. FEL.SFELPROC(CRANDVRA), l'exécutable d'allocation du gestionnaire RAM de CA Endevor® SCM et FEL.SFELPROC(CRAALLOC), l'exécutable d'allocation des gestionnaires RAM personnalisés. L'exécutable d'allocation alloue des fichiers temporaires catalogués qui possèdent le préfixe TSO de l'utilisateur comme qualificatif de haut niveau. L'exécutable d'allocation est également chargé d'appeler l'exit utilisateur facultatif.

Les instructions de personnalisation sont décrites dans l'exécutable. Il est possible, mais non recommandé, de modifier l'exécutable d'allocation car les personnalisations doivent être recommencées lorsque le service PTF met à jour l'exécutable. Si possible, utilisez à la place l'exit utilisateur CARMA, décrit dans «(Facultatif) Exit utilisateur CARMA», à la page 66.

Remarque :

- Lors de la mise à jour d'un exec d'allocation, veillez à ne pas détruire les allocations déjà effectuées dans le processus de démarrage CARMA par CRASTART ou le JCL de démarrage.
- La sortie générée par l'exec d'allocation est indiquée dans la DD SYSTSPRT du serveur CARMA.

Lors de la mise à jour d'un exec d'allocation, veillez à ne pas détruire les allocations déjà effectuées dans le processus de démarrage CARMA par CRASTART ou le JCL de démarrage.

Les exemples suivants montrent comment appeler un exec d'allocation qui nécessite uniquement TSO.

crastart*.conf

```

SYSPROC = my.exec.library
PROGRAM = IKJEFT01 %myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )

CRASUB*
//SYSPROC DD DISP=SHR,DSN=my.exec.library
//SYSTSIN DD *
%myexec ISPSTART PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2. )
//*

```

(Facultatif) Exit utilisateur CARMA

CARMA prend en charge l'appel d'un exit utilisateur pour permettre l'initialisation spécialisée au démarrage et le nettoyage spécialisé à l'arrêt du serveur CARMA. L'utilisation d'un exit utilisateur réduit la nécessité de modifier l'exécutable d'allocation, qui est géré par le service PTF.

L'exit utilisateur est appelé par l'exécutable d'allocation et exécuté deux fois. L'appel de l'initialisation est effectué après l'allocation des fichiers temporaires, mais avant l'appel du serveur CARMA. L'appel du nettoyage est effectué après l'arrêt du serveur CARMA, mais avant la suppression des fichiers temporaires. Si le premier appel se termine avec le code retour 99 ou un code supérieur, le démarrage de CARMA est interrompu. Cela implique que ni le serveur CARMA ni le deuxième appel de cet exit utilisateur n'est exécuté.

Un exemple d'exit utilisateur est fourni sous le nom FEL.SFELSAMP(CRAEXIT), à moins que vous n'ayez utilisé un qualificatif de haut niveau différent lors de l'installation SMP/E de Developer for z Systems. Cet exemple d'exit utilisateur décrit en détails les arguments de démarrage transmis à l'exit utilisateur :

Argument de démarrage	Description
(STARTUP) (ENDING)	Indique si l'appel de l'exit est antérieur ou postérieur à l'appel du serveur CARMA.
EXIT_RC=rc	Code retour du précédent appel de l'exit. rc est toujours égal à 0 lors de l'appel de (STARTUP).
CARMA_RC=rc	Code retour de l'appel du serveur CARMA. rc est toujours égal à 0 lors de l'appel de (STARTUP).
...	Commande de démarrage et arguments de démarrage du serveur CARMA. Par exemple, ISPSTART PGM(CRASERV) PARM(1312 420 EXIT=CRAEXIT CLIENT=9.0.1)

La sortie générée par l'exit utilisateur est indiquée dans la définition de données SYSTSPRT du serveur CARMA.

(Facultatif) IRXJCL versus CRAXJCL

Si le serveur CARMA est démarré à l'aide de TSO (IKJEFTxx), des problèmes peuvent survenir si vos RAM appellent des services qui à leur tour appellent l'interface de traitement par lots REXX IRXJCL. L'incident peut survenir si 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. Il appelle ensuite IRXJCL pour exécuter le travail réel.

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 sont 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 est écrite 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 devez le personnaliser pour spécifier des attributions spécifiques à utiliser pour SYSTSPRT. L'allocation pour SYSTSIN doit généralement être un fichier factice.

Un exemple de code source d'assembleur et un exemple de travail de compilation/liaison sont disponibles respectivement sous la forme `FEL.#CUST.ASM(CRAXJCL)` et `FEL.#CUST.JCL(CRA#CIRX)`, sauf si vous avez défini un emplacement différent lorsque vous avez personnalisé et envoyé le travail `FEL.SFELSAMP(FELSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Personnalisez le code source en assembleur CRAXJCL selon les besoins en utilisant la documentation fournie dans le membre. Ensuite, personnalisez et soumettez le JCL `CRA#CIRX` pour créer le module de chargement CRAXJCL. Pour des instructions de personnalisation, voir la documentation du membre.

Si nécessaire, vous pouvez renommer IRXJCL. Ajustez alors la source CRAXJCL pour appeler ce nouveau nom pour IRXJCL et le compiler, puis remplacez le nom du module chargeable CRAXJCL par IRXJCL. Il peut s'avérer plus aisé de procéder à cette configuration que de changer tous vos appels en IRXJCL.

Chapitre 4. 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 du système hôte inclus dans ISPF.

SCLM Developer Toolkit dispose d'un plug-in Eclipse qui s'interface avec SCLM et fournit l'accès à tous les processus SCLM pour le développement du code hérité et le support de développement complet Java et Java EE sur le poste de travail avec la synchronisation à SCLM sur le mainframe et notamment la génération, l'assemblage et le déploiement du code Java EE à partir du mainframe.

Configuration requise et liste de contrôle

Vous avez 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 les ressources suivantes :

- Mises à jour APF et LINKLIST
- Définition de traducteurs de langage SCLM pour la prise en charge de Java EE
- Définition de types SCLM pour la prise en charge de Java EE
- (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 «Configuration requise», à la page 70.
2. Personnalisez les fichiers de configuration Developer for z Systems. Pour plus d'informations, voir :
 - «Mises à jour d'ISPF.conf pour SCLMDT», à la page 70
 - «Mises à jour rdz.env pour SCLMDT», à la page 71
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 72.
4. Vous pouvez installer et personnaliser Ant pour utiliser le support de génération Java EE. Pour plus d'informations, voir «(Facultatif) Installation et personnalisation d'Ant», à la page 74.
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 75.
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 anciens fichiers de WORKAREA et de /tmp», à la page 76.

Configuration requise

Pour obtenir la liste complète des éléments logiciels requis pour Developer for z Systems, notamment les éléments requis et corequis, voir l'onglet **System Requirements** sur la page du produit Developer for z Systems (<http://www-03.ibm.com/software/products/en/developerforsystemz/>) qui documente également les spécifications Ant requises pour les générations Java EE dans SCLM Developer Toolkit.

Avertissement : SCLM Developer Toolkit utilise la passerelle ISPF existante et par conséquent, requiert z/OS version 1.8 ou supérieure. De plus, si la passerelle ISPF interactive est utilisée pour le service Commandes TSO, la passerelle ISPF existante ainsi que la passerelle ISPF interactive doivent être configurées.

Comme décrit dans la section «Modifications de PARMLIB», à la page 16, SCLM Developer Toolkit requiert une personnalisation supplémentaire des paramètres système. Ces modifications incluent les éléments suivants :

- (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.SISpload, 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.
- Les paramètres par défaut de la commande TSO **OUTPUT** permettent à un utilisateur d'extraire uniquement les sorties de travail commençant par cet ID utilisateur spécifique. Pour utiliser la fonction **OUTPUT** pleinement, l'exemple d'exit TSO/E IKJEFF53 peut nécessiter des modifications afin qu'un utilisateur puisse extraire une sortie de travail dont il est propriétaire mais qui ne commence pas par son 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/ et /var/zexpl/WORKAREA/. WORKAREA se trouve dans /var/zexpl/, sauf si vous spécifiez un emplacement différent lors de la personnalisation d'IBM Explorer for z/OS.

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.SISPLIB est allouée à la concaténation ISPSLIB dans ISPF.conf. L'utilisation du fichier ISP.SISPSENU est facultative.

ISPF.conf se trouve dans /etc/zexpl/, sauf si vous spécifiez un emplacement différent lors de la personnalisation d'IBM Explorer for z/OS. 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 au système 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 (,). .

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

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

Figure 20. 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 "Personnalisation de l'environnement TSO" dans le manuel *IBM Explorer for z/OS - Guide de la configuration hôte (SC27-8438)*.
- 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 `rdz.env` pour `SCLMDT`

`SCLM Developer Toolkit` utilise certaines directives définies dans `rdz.env` pour localiser les fichiers et les répertoires.

`rdz.env` se trouve dans `/etc/zexpl/`, sauf si vous spécifiez un emplacement différent lors de la personnalisation et de la soumission du travail `FEL.SFELSAMP(FELSETUP)`. Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **0EDIT**.

Remarque : Pour que les changements apportés prennent effet, redémarrez la tâche démarrée `RSED IBM Explorer for z/OS`.

L'exemple de code suivant illustre les directives `SCLMDT` présentes dans le fichier `rdz.env` qui doivent être personnalisées pour correspondre à votre environnement système. Pour plus de détails sur la personnalisation de `rdz.env`, voir «`rdz.env`, le fichier de configuration d'environnement», à la page 27.

```

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

```

Figure 21. Mises à jour rdz.env pour SCLMDT

(Facultatif) Conversion de nom long/court

SCLM Developer Toolkit permet d'enregistrer dans SCLM des fichiers dont le nom est long. Il s'agit des noms comportant plus de 8 caractères ou en casse mixte. Ce stockage 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/abrégés

Pour créer la la VSAM de conversion de noms longs/abrégés, personnalisez et soumettez l'exemple de membre FLM02LST de la bibliothèque SFPPF exemple ISP.SISPSAMP. Les étapes de configuration mentionnées dans la présente publication prévoient que le fichier VSAM s'appelle FEL.#CUST.LSTRANS.FILE, comme illustré dans l'exemple suivant de JCL de configuration.

```

//FLM02LST JOB <job parameters>
/*
/* 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 FEL.#CUST.LSTRANS.FILE to
/*     match your naming convention for the SCLM translate VSAM.
/*
//CREATE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE FEL.#CUST.LSTRANS.FILE
SET MAXCC=0
DEFINE CLUSTER(NAME(FEL.#CUST.LSTRANS.FILE) -
              VOLUMES(*****)) -
              RECORDSIZE(58 2048) -
              SHAREOPTIONS(3 3) -
              CYLINDERS(1 1) -
              KEYS(8 0) -
              INDEXED) -
DATA (NAME(FEL.#CUST.LSTRANS.FILE.DATA)) -
INDEX (NAME(FEL.#CUST.LSTRANS.FILE.INDEX))

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

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

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

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

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

```

Figure 22. FLM02LST : JCL de configuration de la conversion de nom long/abrégé

Remarque : Les utilisateurs doivent disposer d'un accès en MISE A JOUR à ce fichier VSAM.

Mises à jour de rdz.env 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 `rdz.env` pour rechercher le nom de la méthode VSAM de conversion de noms longs/courts.

`rdz.env` se trouve dans `/etc/zexpl/`, sauf si vous spécifiez un emplacement différent lors de la personnalisation et de la soumission du travail `FEL.SFELSAMP(FELSETUP)`. 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 : Pour que les changements apportés prennent effet, redémarrez la tâche démarrée RSED.

(Facultatif) Installation et personnalisation d'Ant

Cette étape est requise uniquement si vous projetez d'utiliser la prise en charge de la génération Java EE 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 z Systems:

- Téléchargez le dernier fichier Ant compressé au format binaire dans le système de fichiers z/OS UNIX. Téléchargez la version `.zip` de ANT.
- 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 exister 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 (par exemple, `/usr/lpp/java/J6.0/bin/jar -xf apache-ant-1.7.1.zip`).
- Convertissez tous les fichiers texte Ant en EBCDIC en exécutant et en personnalisant éventuellement l'exemple de script `/usr/lpp/ibm/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 ouvrez un fichier texte dans le répertoire ANT, tel que `apache-ant-1.7.1/README`. Si le fichier est lisible, la conversion a abouti.
- Utilisez la commande **chmod -R 755 *** pour permettre aux utilisateurs de lire et d'exécuter des fichiers dans le répertoire ANT.

- Pour pouvoir utiliser Ant, vous devez définir les variables d'environnement JAVA_HOME et ANT_HOME.
 - La variable JAVA_HOME (dans rse.env) est requise pour pointer vers le répertoire de base Java, par exemple :
`JAVA_HOME=/usr/lpp/java/IBM/J6.0`
 - La variable ANT_HOME (dans rse.env) est requise pour pointer vers le répertoire de base 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/ibm/rdz/samples/BWBTRANT`
- `cat ./apache-ant-1.7.1/README`
- `chmod -R 755 *`
- `oedit /etc/rdz.env`

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:$PATH
export PATH=/usr/lpp/java/J6.0/bin:$PATH
```

- Pour afficher la version, si le programme a été correctement installé, exécutez `ant -version`.

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 manuel *IBM Rational Developer for z Systems SCLM Developer Toolkit Administrator's Guide* (SC23-9801) :

- Définition de traducteurs de langage pour la prise en charge de Java EE
- Définition de types SCLM pour la prise en charge de Java EE

Pour effectuer des tâches de personnalisation et de définition de projet, l'administrateur SCLM doit connaître plusieurs valeurs personnalisables de Developer for z Systems, tel que décrit dans le tableau 13.

Tableau 13. Liste de contrôle de l'administrateur SCLM

Description	<ul style="list-style-type: none"> • Valeur par défaut • Emplacement de la réponse 	Valeur
Exemple de bibliothèque Developer for z Systems	<ul style="list-style-type: none"> • FEL.SFELSAMV • Installation SMP/E 	

Tableau 13. Liste de contrôle de l'administrateur SCLM (suite)

Description	<ul style="list-style-type: none"> • Valeur par défaut • Emplacement de la réponse 	Valeur
Exemple de répertoire Developer for z Systems	<ul style="list-style-type: none"> • /usr/lpp/ibm/rdz/samples • Installation SMP/E 	
Répertoire bin Java	<ul style="list-style-type: none"> • /usr/lpp/java/J6.0/bin • rse.env - \$JAVA_HOME/bin 	
Répertoire bin Ant	<ul style="list-style-type: none"> • /usr/lpp/Apache/Ant/apache-ant-1.7.1/bin • rdz.env - \$ANT_HOME/bin 	
Répertoire initial WORKAREA	<ul style="list-style-type: none"> • /var/zexpl • rse.env - \$CGI_ISPWORK 	
Répertoire initial de configuration de projet SCLMDT	<ul style="list-style-type: none"> • /var/rdz/sclmdt • rdz.env - \$_SCLMDT_CONF_HOME 	
Méthode d'accès VSAM de conversion de nom long/court	<ul style="list-style-type: none"> • FEL.#CUST.LSTRANS.FILE • rdz.env - \$_SCLMDT_TRANTABLE 	

Suppression des anciens fichiers de WORKAREA et de /tmp

SCLM Developer Toolkit et la passerelle existante ISPF partagent les mêmes répertoires WORKAREA et /tmp qui peuvent nécessiter un nettoyage régulier. 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. En utilisant le démon z/OS UNIX cron, qui exécute des commandes à des dates et à des heures données, vous pouvez définir un outil automatisé qui nettoie périodiquement les répertoires cible. Reportez-vous au document UNIX System Services Command Reference (SA22-7802) pour plus d'informations sur le script skulker et le démon cron.

Chapitre 5. Débogueur intégré

Le composant hôte Integrated Debugger (débogueur intégré) de Developer for z Systems autorise les clients version 9.0.1 (ou version ultérieure) à déboguer diverses applications Language Environment (LE). Le débogueur intégré requiert z/OS version 1.10 ou supérieure.

Reportez-vous à la section "Débogueur intégré" dans le chapitre "Comprendre Developer for z Systems" du *Guide de référence de la configuration hôte (SC27-8578)* pour obtenir une présentation du flux de données du débogueur intégré.

Configuration requise et liste de contrôle

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

- Ajout d'un appel du superviseur (redémarrage du système facultatif)
- LPA mise à jour pour l'appel du superviseur
- Mise à jour LINKLIST
- Autorisation APF
- Définition de la tâche démarrée
- Définition des profils de sécurité et des listes d'accès
- Réservation de ports TCP/IP pour la communication client-hôte et les communications limitées au système hôte
- (Facultatif) Mise à jour du JCL de la région CICS
- (Facultatif) Mise à jour de la définition système CICS
- (Facultatif) Mise à jour du JCL de la région de message IMS
- (Facultatif) Mise à jour du JCL d'espace adresse de procédure mémorisée DB2
- (Facultatif) Mise à jour des procédures de construction à distance ELAXF*

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

1. Le débogueur intégré nécessite que la tâche démarrée facultative DBGMGR soit active (conjointement avec la tâche démarrée obligatoire RSED à partir d'IBM Explorer for z/OS). Pour le JCL de démarrage DBGMGR, voir «DBGMGR : Tâche démarrée par le gestionnaire de débogage», à la page 23.
2. La configuration du débogueur intégré est gérée par les arguments de démarrage de la tâche démarrée DBGMGR. Pour plus d'informations, voir «Paramètres de configuration du débogueur intégré», à la page 80.
3. La tâche démarrée DBGMGR requiert que la bibliothèque FEL.SFELAUTH ait des droits APF. Voir «Mises à jour parmlib du débogueur intégré», à la page 80.
4. Le débogueur intégré utilise un appel du superviseur Developer for z Systems (SVC). Le module de chargement SVC doit être chargé dans la zone permanente de programme (LPA) et peut être activé de manière dynamique. Pour plus d'informations, voir «Mises à jour parmlib du débogueur intégré», à la page 80.

5. Le débogueur intégré doit être accessible dans votre application et requiert les mises à jour LINKLIST ou STEPLIB. Pour plus d'informations, voir «Mises à jour parmlib du débogueur intégré», à la page 80.
6. Le débogueur intégré nécessite que l'ID utilisateur de l'application en cours de débogage comporte un segment OMVS valide. Pour plus d'informations, voir «Mises à jour de sécurité du débogueur intégré», à la page 82.
7. Le débogueur intégré requiert certaines autorisations de sécurité. Pour plus d'informations, voir «Mises à jour de sécurité du débogueur intégré», à la page 82.
8. La tâche démarrée DBGMR requiert des autorisations de sécurité. Pour plus d'informations, voir «Mises à jour de sécurité du débogueur intégré», à la page 82.

L'étape suivante est uniquement requise pour le débogage des transactions CICS :

1. Le débogueur intégré peut déboguer des transactions CICS. Le débogueur intégré et IP CICS Sockets doivent être définis sur CICS. Pour plus d'informations, voir «Mises à jour CICS du débogueur intégré», à la page 83.

Le débogueur intégré peut utiliser un exit utilisateur Language Environment (LE) qui lit les options d'exécution TEST dans un fichier pour un module donné. Cela constitue un avantage pour le débogage du code qui est actif dans des sous-systèmes, comme des transactions IMS et des procédures mémorisées DB2, car il n'est pas possible de fournir dynamiquement des options d'exécution TEST à ce code. Les étapes suivantes sont uniquement requises lorsque l'exit utilisateur LE est utilisé.

1. Créez l'exit utilisateur LE et un environnement LE personnalisé qui appelle cet exit utilisateur. Pour plus d'informations, voir «Mises à jour de Language Environment pour le débogueur intégré», à la page 82.
2. Définissez l'environnement LE personnalisé pour votre transaction IMS. Pour plus d'informations, voir «Mises à jour IMS du débogueur intégré», à la page 84.
3. Définissez l'environnement LE personnalisé pour vos procédures mémorisées DB2. Pour plus d'informations, voir «Mises à jour des procédures mémorisées DB2 pour le débogueur intégré», à la page 84.

Remarque :

- Le débogueur intégré utilise CICS CADP pour fournir les options d'exécution TEST aux transactions CICS. Pour plus d'informations sur CADP, voir la documentation CICS TS. Toutefois, si vous le souhaitez, vous pouvez également utiliser le mécanisme d'exit utilisateur LE.
- L'exit utilisateur LE peut également être utilisé pour les applications régulières. Pour plus d'informations, voir «Mises à jour ELAXF* du débogueur intégré», à la page 85.

Débogueur intégré et COBOL v4

Pour déboguer des programmes écrits en COBOL version 3.4 et version 4, le débogueur intégré doit avoir accès à un fichier de liste (PDS ou PDS/E). Le nom de ce fichier peut être fourni via la variable d'environnement AQE_DBG_V4LIST ou la définition de données AQEV4LST. Si aucune d'elles n'est présente, le débogueur intégré constitue le nom du fichier en remplaçant le dernier qualificatif du fichier de l'exécutable (par exemple, .LOAD) par .LISTING. Discutez avec vos développeurs de la méthode à utiliser sur votre site.

Débogueur intégré et autres débogueurs reposant sur Language Environment

Normalement, un seul débogueur reposant sur Language Environment (LE), comme le débogueur intégré, peut être actif dans une application, une région CICS, une procédure mémorisée DB2 ou une transaction IMS donnée. Un débogueur est de type LE s'il fournit un alias ou un module de chargement CEEVDBG qui doit être disponible pour l'application.

Toutefois, le débogueur intégré peut coexister avec IBM Debug Tool for z/OS s'il est chargé en premier par l'application.

Débogueur intégré et communication chiffrée

Si le client Developer for z Systems utilise le chiffrement pour communiquer avec le démon RSE, le moteur de débogage (basé sur le client) utilise aussi par défaut le chiffrement pour communiquer avec le gestionnaire de débogage (basé sur un hôte). Par défaut, le moteur de débogage utilise les mêmes certificats que le client Developer for z Systems.

Cela implique que la tâche démarrée du démon RSE et la tâche démarrée du gestionnaire de débogage sont censées être configurées à l'identique pour le chiffrement des communications. Les scénarios de remplacement suivants sont disponibles dans les cas où les paramètres de chiffrement du démon RSE et du gestionnaire de débogage sont différents :

- Le moteur de débogage peut, après confirmation, essayer d'utiliser une communication non chiffrée si la session chiffrée échoue.
- Le moteur de débogage peut utiliser d'autres certificats d'authentification que le client Developer for z Systems.

Contrairement au démon RSE, le gestionnaire de débogage ne dispose pas de la prise en charge native de la communication chiffrée. Il fait appel à un service TCP/IP appelé Application Transparent Transport Layer Security (AT-TLS) pour les communications chiffrées. Voir la rubrique "Configuration d'AT-TLS" dans le manuel *IBM Rational Developer for z Systems - Guide de référence de la configuration hôte (SC27-8578)* pour obtenir un guide de configuration pas à pas.

Débogueur intégré et SYSPLEX

Developer for z Systems ne reconnaît pas SYSPLEX et exige donc que le programme à déboguer soit actif sur le même système que celui auquel le client Developer for z Systems est connecté. Cela inclut les sessions de débogage démarrées via un travail par lots.

Dans z/OS 2.1, vous pouvez spécifier SYSAFF=* ou SYSTEM=* sur la carte de travail pour imposer que le travail s'exécute sur le système sur lequel il a été soumis. Sur les systèmes plus anciens, vous devez explicitement spécifier le bon nom de système lorsque vous utilisez JESPLEX pour réunir plusieurs sous-systèmes JES dans un SYSPLEX.

Paramètres de configuration du débogueur intégré

Le débogueur intégré permet de configurer les variables suivantes dans le JCL de démarrage DBGMGR. Pour le JCL de démarrage DBGMGR, voir «DBGMGR : Tâche démarrée par le gestionnaire de débogage», à la page 23.

- Le décalage horaire (par défaut, EST5DST)
- Le port utilisé pour les communications externes (client-hôte) (par défaut, 5335)
- Le port utilisé pour les communications internes (limitées au système hôte) (par défaut, 5336)
- Numéro SVC utilisé par Integrated Debugger, par défaut 251
- Le qualificatif de haut niveau de la bibliothèque de chargement, par défaut FEL

Mises à jour parmlib du débogueur intégré

- La tâche démarrée DBGMGR doit être active pour que les clients Developer for z Systems puissent se connecter à l'hôte. Pour plus de détails sur le démarrage automatique du serveur au démarrage du système, voir «Ajout des tâches démarrées à COMMNDxx», à la page 18.
- La tâche démarrée DBGMGR requiert que la bibliothèque FEL.SFELAUTH ait des droits APF. Pour plus d'informations, voir «Autorisations APF dans PROGxx», à la page 20.
- Language Environment (LE) doit pouvoir appeler le débogueur intégré. La bibliothèque FEL.SFELAUTH doit donc être placée dans l'instruction LINKLIST ou STEPLIB de l'application à déboguer. Pour plus d'informations, voir «Définitions LINKLIST dans PROGxx», à la page 20.

Remarque :

- Lors de l'utilisation de LINKLIST, vérifiez que FEL.SFELAUTH se trouve avant les bibliothèques des autres débogueurs LE contenant le module de chargement CEEVDBG. Par exemple, IBM Debug Tool for z/OS utilise les bibliothèques hlq.SEQA*.
- Pour éviter les conflits, un seul débogueur basé sur LE doit être défini dans LINKLIST.
- Le débogueur intégré peut coexister avec IBM Debug Tool for z/OS s'il est chargé d'abord par l'application.
- Sauf pour la tâche démarrée du gestionnaire de débogage, les modules de chargement du débogueur intégré dans FEL.SFELAUTH ne requièrent pas d'autorisation pour s'exécuter. Les modules de chargement se trouvent ici afin de pouvoir être utilisés dans un environnement autorisé.
- Le débogueur intégré utilise l'API z/OS Binder. Cela implique que SYS1.MIGLIB doit se trouver dans LINKLIST (ou STEPLIB). Pour plus d'informations, voir «Définitions LINKLIST et LPA prérequis», à la page 22.
- Le débogueur intégré utilise l'API z/OS Binder. Cette API est disponible depuis z/OS version 1.10 sous la forme /usr/lib/iewbndd.so et également depuis z/OS version 1.13 sous la forme SYS1.SIEAMIGE(IEWBNDD). Cela implique que pour z/OS version 1.13 et ultérieures, SYS1.SIEAMIGE se trouve dans LINKLIST (ou STEPLIB). Pour plus d'informations, voir «Définitions LINKLIST et LPA prérequis», à la page 22.

Remarque : Si SYS1.SIEAMIGE ne figure pas dans LINKLIST ou STEPLIB sur les systèmes z/OS 1.13 et ultérieurs, le débogueur intégré émettra le message suivant et tentera d'utiliser /usr/lib/iewbndd.so :

CEE3501S The module //IEWBNDD was not found

- Le débogueur intégré nécessite qu'un appel du superviseur Developer for z Systems soit défini pour votre système. Le numéro SVC par défaut est 251. Le module de chargement associé, FEL.SFELLPA(AQESVC03), doit être chargé dans la zone permanente de programme (LPA) et peut être activé de manière dynamique. Pour plus d'informations, voir «Définitions SVC dans IEASVCxx», à la page 18 et «Définitions LPA dans LPALSTxx», à la page 19.

Remarque : Les versions de Developer for z Systems antérieures à la version 9.1.1 utilisaient un autre SVC, AQESVC01, qui n'est plus utilisé. Il peut être supprimé si votre système hôte utilise uniquement Developer for z Systems version 9.1.1 ou une version ultérieure.

Mises à jour de l'appel du superviseur du débogueur intégré

Le débogueur intégré nécessite qu'un appel du superviseur (SVC) de Developer for z Systems, FEL.SFELLPA(AQESVC03), soit défini pour votre système. Dans des circonstances idéales, ce SVC est activé ou mis à jour durant l'IPL du système. Cependant, l'IPL n'est pas toujours une option, de sorte que le débogueur intégré procédera à une installation ou à une mise à jour dynamique à l'aide de la tâche démarrée du gestionnaire de débogage.

Lors du démarrage, la tâche démarrée du gestionnaire de débogage, DBGMGR, vérifie les éléments suivants, puis exécute l'action associée, comme indiqué dans tableau 14 :

- l'appel du superviseur est défini
- la version de l'appel du superviseur actif (si défini) et le nouvel appel du superviseur (dans SFELLPA)
- l'argument de démarrage SVC=svc_number est spécifié, où svc_number correspond au nombre SVC souhaité

Tableau 14. Mise à jour dynamique de l'appel du superviseur du débogueur intégré

	Appel du superviseur défini	Appel du superviseur non défini
Version de la correspondance SVC actif et SVC nouveau	<ul style="list-style-type: none"> • Aucune action, l'appel du superviseur étant déjà actif et à jour • L'argument de démarrage de l'appel du superviseur est ignoré 	<ul style="list-style-type: none"> • Définition dynamique de l'appel du superviseur à l'aide du module de chargement SFEKLPA • L'argument de démarrage de l'appel du superviseur est requis
Les versions des SVC actif et nouveau ne correspondent pas	<ul style="list-style-type: none"> • Mise à jour de l'appel du superviseur à l'aide du module de chargement SFEKLPA • L'argument de démarrage de l'appel du superviseur est ignoré 	<ul style="list-style-type: none"> • Définition dynamique de l'appel du superviseur à l'aide du module de chargement SFEKLPA • L'argument de démarrage de l'appel du superviseur est requis

Mises à jour TCP/IP du débogueur intégré

Le débogueur intégré utilise 2 ports TCP/IP.

- Port pour une communication client-hôte (par défaut, 5335). La communication sur ce port peut être chiffrée.
- Port pour une communication limitée au système hôte (par défaut, 5336).

Mises à jour de sécurité du débogueur intégré

Le débogueur intégré requiert les définitions de sécurité suivantes. Pour plus d'informations, voir Chapitre 9, «Définitions de sécurité», à la page 103.

- Segment OMVS pour l'ID utilisateur qui exécute l'application en cours de débogage (dont l'ID utilisateur de la région CICS)
- Tâche démarrée DBGMGR
- Autorisation BPX.SERVER pour l'ID utilisateur de la tâche démarrée
- Contrôle de programmes pour la bibliothèque de chargement de la tâche démarrée
- Autorisation AQE.AUTHDEBUG.STDPGM pour les utilisateurs déboguant les programmes d'état de programme.
- AQE.AUTHDEBUG.AUTHPGM pour les utilisateurs déboguant les programmes autorisés et les programmes à l'état problème.

Remarque : Pour simplifier la migration à partir d'une configuration Developer for z Systems existante sans le débogueur intégré, l'exemple de JCL FEL.SFELSAMP(AQERACF) avec des commandes RACF est fourni uniquement pour les définitions de sécurité liées au débogueur intégré.

Mises à jour de Language Environment pour le débogueur intégré

Le débogueur intégré peut utiliser un exit utilisateur Language Environment (LE) qui lit les options d'exécution TEST dans un fichier pour un module donné. Cela constitue un avantage pour le débogage du code qui est actif dans des sous-systèmes, comme des transactions IMS et des procédures mémorisées DB2, car il n'est pas possible de fournir dynamiquement des options d'exécution TEST à ce code.

Les étapes suivantes décrivent le flux de processus :

1. LE est invité à exécuter un module.
2. LE appelle l'exit utilisateur avec le nom du module comme argument.
3. L'exit utilisateur ouvre le fichier d'options d'exécution TEST et recherche les options d'exécution pour le module.
4. L'exit utilisateur renvoie les options d'exécution TEST à LE.
5. LE exécute le module avec les options d'exécution TEST spécifiées.

Remarque :

- IBM Debug Tool for z/OS utilise une version étendue du mécanisme d'exit utilisateur décrit ici. Pour prendre en charge la coexistence et éviter toute redondance pour la gestion des options d'exécution TEST, le débogueur intégré utilise les mêmes valeurs par défaut que l'outil Debug Tool pour les options de l'exit utilisateur et le nom de fichier des options d'exécution TEST.
- L'exit utilisateur prend en charge la substitution de variable pour l'ID utilisateur et le nom de module dans le nom de fichier des options d'exécution TEST.
- L'hôte et le client Developer for z Systems doivent gérer le fichier des options d'exécution TEST avec l'interface graphique du client Developer for z Systems.

- Le débogueur intégré utilise CICS CADP pour fournir les options d'exécution TEST aux transactions CICS. Pour plus d'informations sur CADP, voir la documentation CICSTS.

Les étapes suivantes sont uniquement requises lorsque l'exit utilisateur LE est utilisé.

- Personnalisez et soumettez AQED3CXT afin de créer l'exit utilisateur. AQED3CXT se trouve dans FEL.#CUST.JCL, sauf si vous avez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP).
- Personnalisez et soumettez AQED3CEE pour créer les modules de chargement LE personnalisés qui appellent l'exit utilisateur. AQED3CEE se trouve dans FEL.#CUST.JCL, sauf si vous avez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Par défaut, les modules de chargement LE personnalisés (CEE*) sont placés dans FEL.#CUST.LOAD.

Remarque : Pour des raisons de performances, ne placez pas la bibliothèque de chargement qui contient les modules de chargement LE personnalisés dans LINKLIST. L'exit utilisateur sera appelé pour tout module de chargement compatible LE, y compris un logiciel acheté.

Mises à jour CICS du débogueur intégré

Pour déboguer des transactions CICS, le débogueur intégré requiert les mises à jour CICS ci-dessous.

- Mises à jour du paramètre d'initialisation du système CICS (SIT) :
 - Indiquez DEBUGTOOL=YES.
 - Indiquez TCPIP=YES.
 - Indiquez LLACOPY=YES si vous dépendez de LINKLIST pour extraire un module de chargement à partir de la concaténation de définition de données DFHRPL.
- Mises à jour du JCL CICS :
 - Indiquez REGION=0M dans l'instruction EXEC.
 - Définissez la bibliothèque de chargement FEL.SFELAUTH dans l'instruction de définition de données DFHRPL de la région. Si le paramètre SIT LLACOPY=YES est spécifié, la bibliothèque peut également résider dans LINKLIST.
 - Définissez la bibliothèque de chargement SYS1.MIGLIB dans l'instruction de définition de données DFHRPL de la région. Si le paramètre SIT LLACOPY=YES est spécifié, la bibliothèque peut également résider dans LINKLIST.
 - Pour z/OS versions 1.13 et ultérieures, définissez la bibliothèque de chargement SYS1.SIEAMIGE dans l'instruction de définition de données DFHRPL de la région. Si le paramètre SIT LLACOPY=YES est spécifié, la bibliothèque peut également résider dans LINKLIST. Pour plus de détails, voir les informations relatives à l'API z/OS Binder dans «Mises à jour parmlib du débogueur intégré», à la page 80.

Remarque :

- L'ID utilisateur de la région CICS requiert le droit UPDATE sur le profil CSVLLA.dataset dans la classe FACILITY pour le paramètre SIT LLACOPY=YES afin de fonctionner de manière appropriée.
- Pour déboguer des programmes écrits en COBOL v4, le débogueur intégré doit avoir accès à un fichier de liste (PDS ou PDS/E). Le nom de ce fichier

peut être fourni via la variable d'environnement AQE_DBG_V4LIST ou la définition de données AQEV4LST. Si aucune d'elles n'est présente, le débogueur intégré constitue le nom du fichier en remplaçant le dernier qualificatif du fichier de l'exécutable (par exemple, .LOAD) par .LISTING). Demandez à vos développeurs la méthode à utiliser sur votre site.

- Mises à jour de la définition système CICS :

Définissez le débogueur sur une région CICS, comme indiqué dans l'exemple de travail de mise à jour de la définition système CICS AQECSD. AQECSD se trouve dans FEL.#CUST.JCL, sauf si vous avez spécifié un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Remarque :

- Le débogueur intégré requiert l'activation de la communication reposant sur TCP/IP dans votre région CICS. Il est recommandé d'utiliser z/OS Communication Server IP CICS Sockets. Pour des informations sur la configuration de cette fonction, voir le manuel *Communication Server IP CICS Sockets Guide* (SC31-8807).
- Le débogueur intégré utilise CICS CADP pour fournir les options d'exécution TEST aux transactions CICS. Pour plus d'informations sur CADP, voir la documentation CICSTS.

Mises à jour IMS du débogueur intégré

Pour déboguer des transactions IMS, le débogueur intégré requiert les mises à jour IMS ci-dessous :

- Mises à jour du JCL de la région des messages IMS (DFSMPR) :
 - Indiquez REGION=0M dans l'instruction EXEC de la région.
 - Ajoutez la bibliothèque de chargement FEL.SFELAUTH à l'instruction STEPLIB, si elle ne se trouve pas dans LINKLIST. Il n'est pas nécessaire de maintenir l'autorisation APF de tous les ensembles de données STEPLIST, sauf si elle est requise pour d'autres produits que le débogueur intégré.
 - Lorsque vous utilisez l'exit utilisateur Language Environment (LE), ajoutez la bibliothèque de chargement contenant les modules de chargement LE personnalisés (par défaut FEL.#CUST.LOAD) à STEPLIB.

Mises à jour des procédures mémorisées DB2 pour le débogueur intégré

Pour déboguer des procédures mémorisées DB2, le débogueur intégré a besoin des mises à jour DB2 suivantes.

- Mises à jour de la tâche démarrée JCL pour l'espace adresse de procédure mémorisée DB2 :
 - Indiquez REGION=0M dans l'instruction EXEC.
 - Ajoutez la bibliothèque de chargement FEL.SFELAUTH à l'instruction STEPLIB, si elle ne se trouve pas dans LINKLIST. Il n'est pas nécessaire de maintenir l'autorisation APF de tous les ensembles de données STEPLIST, sauf si elle est requise pour d'autres produits que le débogueur intégré.
 - Lorsque vous utilisez l'exit utilisateur Language Environment (LE), ajoutez la bibliothèque de chargement contenant les modules de chargement LE personnalisés (par défaut FEL.#CUST.LOAD) à STEPLIB.

Mises à jour ELAXF* du débogueur intégré

Les mises à jour suivantes sont requises pour pouvoir utiliser le débogueur intégré dans les procédures de construction à distance fournies par Developer for z Systems. Pour plus d'informations sur la fonction et l'emplacement de ces procédures JCL, voir «Procédures de construction à distance ELAXF*», à la page 24.

- Mises à jour des procédures JCL ELAXFDCL, ELAXFG0 et ELAXFTS0 :
 - Indiquez REGION=0M dans l'instruction EXEC.
 - Ajoutez la bibliothèque de chargement FEL.SFELAUTH à l'instruction STEPLIB, si elle ne se trouve pas dans LINKLIST. Il n'est pas nécessaire de maintenir l'autorisation APF de tous les ensembles de données STEPLIST, sauf si elle est requise pour d'autres produits que le débogueur intégré.
 - Lorsque vous utilisez l'exit utilisateur Language Environment (LE), ajoutez la bibliothèque de chargement contenant les modules de chargement LE personnalisés (par défaut FEL.#CUST.LOAD) à STEPLIB.

Chapitre 6. Analyse de code basée sur l'hôte

Comme le client Developer for z Systems, l'hôte Developer for z Systems prend en charge l'exécution des outils d'analyse de code, conditionnés sous la forme d'un produit distinct, IBM Rational Developer for z Systems Host Utilities. L'exécution de l'analyse de code sur l'hôte présente l'avantage de permettre son intégration au traitement par lots des tâches quotidiennes.

Les outils d'analyse de code suivants sont disponibles sur l'hôte :

- Révision du code (Code Review) : En utilisant des règles associées à différents niveaux de sécurité, Code Review analyse le code source et signale les violations de règle.
- Couverture de code (Code Coverage) : Analyse un programme en cours et génère un rapport sur les lignes exécutées par rapport au nombre total de lignes exécutables.

Configuration requise et liste de contrôle

Vous n'avez pas besoin de l'aide d'autres administrateurs pour commencer à utiliser les outils d'analyse de code sur l'hôte dans votre site, mais vous devez réaliser les tâches suivantes. Sauf indication contraire, toutes les tâches sont obligatoires.

1. Installez IBM Rational Developer for z Systems Host Utilities, en suivant les instructions du document *Program Directory for IBM Rational Developer for z Systems Host Utilities* (GI11-7463). Si vous utilisez les valeurs par défaut, le produit est installé avec le qualificatif de haut niveau AKG et le chemin z/OS UNIX /usr/lpp/rdzutil.
2. Créez des copies personnalisables des exemples fournis en personnalisant et en soumettant AKG.SAKGSAMP(AKGSETUP). Ce travail effectue les tâches suivantes :
 - Créez AKG.#CUST.PROCLIB et remplissez-le avec des exemples de membres SYS1.PROCLIB.
 - Créez AKG.#CUST.JCL et remplissez-le avec un exemple de JCL de configuration.

Remarque : IBM Debug Tool for z/OS est un prérequis lorsque vous utilisez le composant de couverture de code.

Révision du code

La révision du code analyse le code source et signale les violations de règle en utilisant des règles associées à différents niveaux de sécurité. L'outil est livré avec des fournisseurs de règles destinés à Cobol et à PL/I, mais vous pouvez en ajouter d'autres.

Developer for z Systems Host Utilities fournit un exemple de procédure, AKGCR, pour simplifier l'appel des services de révision du code en mode de traitement par lots. AKGCR se trouve dans AKG.#CUST.PROCLIB, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail AKG.SAKGSAMP(AKGSETUP).

Personnalisez l'exemple de procédure, `AKG.#CUST.PROCLIB(AKGCR)`, comme indiqué dans le membre, puis copiez-le dans `SYS1.PROCLIB`.

Si la procédure `AKGCR` ne peut pas être copiée dans une bibliothèque de procédure système, demandez aux utilisateurs de Developer for z Systems d'ajouter à leur travail d'appel une carte `JCLLIB` juste après la carte `JOB`.

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

Modification du traitement de révision du code

Developer for z Systems Code Review (révision du code) autorise l'intégration de code tiers au processus de révision. Vous pouvez, par exemple, ajouter un fournisseur de règles pour analyser du code C/C++, ou étendre le fournisseur de règles Cobol pour qu'il reconnaisse les conventions de codification d'un site particulier.

La fonction de révision du code réalisée sur l'hôte est un processus Eclipse, comme le client Developer for z Systems. Ainsi, les améliorations apportées à la révision du code sur le client par votre équipe chargée du support au développement sont réutilisables sur l'hôte.

Les améliorations sont conditionnées sous la forme de plug-in ou de fonctions Eclipse. Pour les activer, vous devez les mettre à disposition du code existant, selon les instructions du travail de configuration `AKGCRADD`. `AKGCRADD` se trouve dans `AKG.#CUST.JCL`, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail `AKG.SAKGSAMP(AKGSETUP)`.

Couverture de code

La couverture de code analyse un programme en cours et génère un rapport sur les lignes exécutées par rapport au nombre total de lignes exécutables. Notez que la couverture de code établit, en utilisant un port éphémère, une connexion TCP/IP à IBM Debug Tool for z/OS ou au moteur de débogage du débogueur intégré d'un client Developer for z Systems.

Developer for z Systems Host Utilities permet d'appeler la couverture de code en mode de traitement par lots de deux manières : un exemple de procédure `JCL` pour traiter une exécution de programme unique, et un ensemble de scripts pour démarrer et arrêter un collecteur de couverture de code actif en permanence, capable de traiter plusieurs exécutions de programme.

Appel de couverture de code unique

L'exemple de procédure `AKGCC` fournit une méthode pour démarrer un collecteur de couverture de code, lui faire analyser une exécution de programme unique, arrêter le collecteur et archiver les résultats pour une utilisation ultérieure.

`AKGCC` se trouve dans `AKG.#CUST.PROCLIB`, sauf si vous avez indiqué un autre emplacement lorsque vous avez personnalisé et soumis le travail `AKG.SAKGSAMP(AKGSETUP)`.

Personnalisez l'exemple de procédure, `AKG.#CUST.PROCLIB(AKGCC)`, comme indiqué dans le membre, puis copiez-le dans `SYS1.PROCLIB`.

Si la procédure AKGCC ne peut pas être copiée dans une bibliothèque de procédure système, demandez aux utilisateurs de Developer for z Systems d'ajouter à leur travail d'appel une carte JCLLIB juste après la carte JOB.

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

Appels de couverture de code multiples

Selon la manière dont votre processus de développement de logiciel est configuré, les avantages apportés par le fait que la procédure AKGCC se charge de tout peuvent ne pas compenser la consommation de ressources et de temps correspondant au démarrage d'un collecteur de couverture de code pour chaque analyse de programme.

Developer for z Systems Host Utilities fournit le script ccstart pour démarrer un collecteur de couverture de code qui reste actif. Ce collecteur peut alors être utilisé dans plusieurs appels de couverture de code. Le script ccstop peut être utilisé pour arrêter le collecteur.

Ces scripts (ccstart, ccstop et codecov) se trouvent dans le répertoire /usr/lpp/rdzutil/bin/ si vous avez installé Developer for z Systems Host Utilities dans l'emplacement par défaut. L'utilisation de ces scripts est expliquée dans l'IBM Knowledge Center Developer for z Systems (http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html).

Voici un scénario d'utilisation générique :

1. Appelez ccstart avec l'option pour utiliser un numéro de port fixe et l'option pour rediriger la sortie vers un emplacement connu.
2. Appelez aussi souvent que nécessaire les programmes à analyser avec l'option de démarrage. TEST(,,,TCP/IP&<ip_hôte>:<port>).
3. Appelez ccstop.
4. Dirigez le client Developer for z Systems vers l'emplacement de la sortie pour afficher les rapports.

Rapports générés par la fonction de couverture de code

Les rapports générés par la fonction de couverture de code sont destinés à être importés dans un client Developer for z Systems. Ils sont donc enregistrés dans un fichier z/OS UNIX. La couverture de code est également capable d'exploiter les résultats d'une exécution précédente et de les associer à ceux de l'exécution en cours pour inclure dans le même rapport plusieurs chemins de code.

Pour ces différentes raisons, Developer for z Systems Host Utilities ne supprime pas les sorties générées par la couverture de code, et celles-ci s'accumulent au fur et à mesure.

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. En utilisant le démon z/OS UNIX cron, qui exécute des commandes à des dates et à des heures données, vous pouvez définir un outil automatisé qui nettoie périodiquement les répertoires cible. Pour plus d'informations sur le script skulker et sur le démon cron, voir le manuel *UNIX System Services Command Reference* (SA22-7802).

Chapitre 7. Autres tâches de personnalisation

La présente section regroupe diverses tâches de personnalisation facultatives. Pour configurer le service requis, suivez les instructions de la section appropriée.

Personnalisations dans les fichiers de configuration Developer for z Systems :

- «`include.conf`, inclusions forcées pour l'assistant de contenu C/C++»

Personnalisations associées à Developer for z Systems ou pour d'autres produits :

- «Sous-projets z/OS UNIX», à la page 92
- «Prise en charge d'inclusion de précompilateur», à la page 93
- «Prise en charge de xUnit pour Enterprise COBOL et PL/I», à la page 94
- «support Enterprise Service Tools», à la page 95
- «Support de langue bidirectionnelle CICS», à la page 95
- «Messages IRZ de diagnostic pour Enterprise Service Tools», à la page 96
- «FEKRNPIL, préprocesseur distant PL/I», à la page 97

`include.conf`, inclusions forcées pour l'assistant de contenu C/C++

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

L'assistant de contenu pour C/C++ peut utiliser les définitions contenues dans `include.conf` pour exécuter des inclusions forcées de fichiers ou de membres spécifiés. Une inclusion forcée signifie qu'un fichier, un répertoire ou un membre de fichier sera analysé lors de l'exécution d'une opération d'assistant de contenu, même si ce fichier ou membre a été inclus dans le code source à l'aide d'une directive de précompilateur.

Le fichier doit être référencé dans `rdz.env` par les variables `include.c` ou `include.cpp` avant d'être utilisé. Cette référence dans `rdz.env` implique que vous pouvez spécifier un autre fichier en vue d'une utilisation par C et C++. Les variables de `rdz.env` sont désactivées par défaut.

L'exemple `include.conf` se trouve dans `/etc/zexpl/`, sauf si vous avez spécifié un autre emplacement lorsque vous avez personnalisé et soumis le travail `FEL.SFELSAMP(FELSETUP)`. 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 définitions doivent commencer en colonne 1. Les lignes de commentaire débutent par un signe dièse (#) lorsqu'une page de codes US est utilisée. Les lignes de données peuvent comporter uniquement un nom de répertoire, de fichier ou de membre. Les commentaires ne sont pas autorisés sur la même ligne. La continuation de ligne n'est pas prise en charge.

```
# To include the stdio.h file from the /usr/include directory, input:
# /usr/include/stdio.h
#
# To include all files of the /usr/include directory and all of it's
# sub-directories, input:
# /usr/include
#
# Uncomment and customize variable FILETYPES to limit the z/OS UNIX
# wildcard include to selected (case sensitive) file types:
# The file types are specified in a comma-delimited list (no blanks)
# FILETYPES=H,h,hpp,C,c,cpp,cxx

# To include all members of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H
#
# To include the STDIOSTR member of the CBC.SCLBH.H data set, input:
# //CBC.SCLBH.H(STDIOSTR)
# The sample list contains some commonly used C standard library files
/usr/include/assert.h
/usr/include/ctype.h
/usr/include/errno.h
/usr/include/float.h
/usr/include/limits.h
/usr/include/locale.h
/usr/include/math.h
/usr/include/setjmp.h
/usr/include/signal.h
/usr/include/stdarg.h
/usr/include/stddef.h
/usr/include/stdio.h
/usr/include/stdlib.h
/usr/include/string.h
/usr/include/time.h
```

Figure 23. include.conf - Inclusions forcées pour l'assistant de contenu C/C++

Sous-projets z/OS UNIX

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 le système hôte. SSH (interpréteur de commandes sécurisé) est un service similaire, mais toutes les communications sont chiffrées. Developer for z Systems utilise l'un de ces services pour effectuer des actions à distance (basées sur l'hôte) dans des sous-projets z/OS UNIX.

Remarque :

- Developer for z Systems utilise la version z/OS UNIX de REXEC et non la version TSO.
- Si REXEC/SSH n'est pas configuré pour utiliser le port par défaut, le client de Developer for z Systems doit définir le port correct à utiliser avec les sous-projets z/OS UNIX. Cette configuration peut être effectuée en sélectionnant la page des préférences **Fenêtre > Préférences > Solutions z/O > Sous-projets USS > Options d'action distante**. Pour connaître le port utilisé, voir «Configuration REXEC ou SSH», à la page 93.

Configuration REXEC ou SSH

REXEC et SSH reposent sur les services fournis par INETD (Internet Daemon), qui est un autre service TCP/IP. Le document *Communications Server IP Configuration Guide* (SC31-8775) décrit la procédure de configuration de INETD, REXEC et SSH. Pour plus de détails et d'autres méthodes de configuration, consulter le livre blanc *Using INETD, REXEC and SSH with Developer for System z (SC14-7301)*, disponible dans la bibliothèque Developer for z Systems, <http://www-01.ibm.com/support/docview.wss?uid=swg27038517>.

Le port généralement utilisé par REXEC est le 512. Pour vérifier le port utilisé, consultez `/etc/inetd.conf` et `/etc/services`.

- 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`.

Prise en charge d'inclusion de précompilateur

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

Developer for z Systems prend en charge l'interprétation et le développement des instructions include COBOL et PL/I, y compris les instructions include tiers sélectionnées. Developer for z Systems fournit également l'exemple d'une instruction `exec REXX`, `FEKRNPLI`, qui peut être appelée par le client Developer for z Systems pour développer la source PL/I en appelant le compilateur PL/I.

`FEKRNPLI` se trouve dans `FEL.#CUST.CNTL` sauf si vous avez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail `FEL.SFELSAMP(FELSETUP)`. Pour plus d'informations, voir Chapitre 2, «Personnalisation de base», à la page 15.

Personnalisez l'exemple d'exécutable `FEL.#CUST.CNTL(FEKRNPLI)`, comme indiqué dans le membre. Vous devez fournir les informations suivantes :

- `compiler_hlq` : Qualificatif de haut niveau du compilateur PL/I

Le client Developer for z Systems utilise le service de commandes TSO pour exécuter l'exécutable. Cela implique que si l'exécutable `FEKRNPLI` se trouve dans la concaténation `SYSPROC` ou `SYSEXEC` pour le service de commandes TSO, l'utilisateur n'a pas besoin de connaître l'emplacement exact de l'exécutable, mais juste son nom. Lorsque la passerelle ISPF existante (Legacy ISPF Gateway) est utilisée, la concaténation de `SYSPROC` ou `SYSEXEC` est définie dans `ISPF.conf`. Vous trouverez de plus amples informations sur la personnalisation de ce fichier dans le manuel *IBM Explorer for z/OS - Guide de configuration hôte* (SC27-8437).

Prise en charge de xUnit pour Enterprise COBOL et PL/I

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
-

Les structures qui assistent les développeurs lors de l'écriture d'un code destiné à exécuter des tests d'unité automatisés reproductibles sont collectivement appelées xUnit. Developer for z Systems fournit une structure de ce type pour les tests d'unité du code Enterprise COBOL et PL/I, appelée zUnit.

Pour utiliser la structure zUnit, les développeurs doivent accéder aux modules de chargement AZU* et IAZU* dans la bibliothèque de chargement FEL.SFELLOAD, via STEPLIB ou LINKLIST. Le programme d'exécution de test zUnit, AZUTSTRN, quant à lui, doit accéder à diverses bibliothèques système, via STEPLIB ou LINKLIST :

- CEE.SCEERUN et CEE.SCEERUN2 (environnement d'exécution LE)
- SYS1.CSSLIB (services système appelables)
- SYS1.SIXML0D1 (kit d'outils XML)

Le programme d'exécution de test de zUnit doit également accéder à une bibliothèque de chargement qui contient les différents scénarios de test. Cette bibliothèque est vraisemblablement unique pour un développeur.

Le programme d'exécution de test de zUnit, AZUTSTRN, peut être appelé par le client Developer for z Systems en mode de traitement par lots, à partir de la ligne de commande TSO et à partir de la ligne de commande z/OS UNIX.

- Developer for z Systems fournit un exemple de procédure, AZUZUNIT, afin de simplifier l'appel du programme d'exécution de test de zUnit en mode de traitement par lots. AZUZUNIT se trouve dans FEL.#CUST.PROCLIB sauf si vous avez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Personnalisez l'exemple de procédure, FEL.#CUST.PROCLIB(AZUZUNIT), comme indiqué dans le membre, puis copiez-le dans SYS1.PROCLIB.

Les noms de la procédures et les noms des étapes de la procédure correspondent aux propriétés par défaut incluses dans le client Developer for z Systems. Si le nom d'une procédure ou le nom d'une étape de la procédure est changé, le fichier des propriétés correspondant sur tous les clients doit également être mis à jour. Il est recommandé de ne pas changer les noms de la procédure et de l'étape.

Si la procédure AZUZUNIT ne peut pas être copiée dans une bibliothèque de procédure système, demandez aux utilisateurs de Developer for z Systems d'ajouter à leur travail d'appel une carte JCLLIB juste après la carte JOB.

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

- Pour l'appel du programme d'exécution de test de zUnit à partir de z/OS UNIX (à l'aide du script /usr/lpp/ibm/rdz/bin/zunit), vous pouvez spécifier les fichiers non-LINKLIST requis dans la directive STEPLIB du fichier rdz.env, ce qui simplifiera la configuration du développeur.

rdz.env se trouve dans /etc/zexpl/, sauf si vous spécifiez un emplacement différent lors de la personnalisation et de la soumission du travail

FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Vous pouvez modifier le fichier à l'aide de la commande TSO **0EDIT**.

Le script *zunit* permet à l'utilisateur de spécifier des fichiers qui seront ajoutés à la directive STEPLIB utilisée par le script.

- Pour permettre l'appel du programme d'exécution de test zUnit à partir de la ligne de commande TSO à l'aide de la commande exec FEL.SFELPROC(FEKZUNIT), les bibliothèques système doivent figurer dans LINKLIST. Dans le cas contraire, les développeurs doivent spécifier les noms de fichiers système à chaque appel du programme d'exécution de test zUnit. Vous pouvez également écrire une commande exec d'encapsuleur qui effectue les affectations **TSOLIB** de ces fichiers. Vous pouvez utiliser FEKZUNIT comme exemple de codage de cette commande exec d'encapsuleur.

Le programme d'exécution de test zUnit permet le reformatage automatique des rapports de test. Developer for z Systems fournit des exemples de conversion (au format Ant ou jUnit, par exemple) qui se trouvent dans les répertoires /usr/lpp/ibm/rdz/samples/zunit/xsd et /usr/lpp/ibm/rdz/samples/zunit/xsl si vous avez installé Developer for z Systems à l'emplacement par défaut /usr/lpp/ibm/rdz.

support Enterprise Service Tools

Cette tâche de personnalisation ne requiert aucune aide, ressource ou tâche de personnalisation spécifique.

Le client Developer for z Systems contient un composant de génération de code appelé Enterprise Service Tools. Selon le type de code généré, ce code s'appuie sur les fonctions fournies par l'installation du système hôte Developer for z Systems. Les sections suivantes expliquent comment rendre ces fonctions du système hôte disponibles :

- «Support de langue bidirectionnelle CICS»
- «Messages IRZ de diagnostic pour Enterprise Service Tools», à la page 96

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

Support de langue bidirectionnelle CICS

Vous avez 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 Developer for z Systems 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 Enterprise Service Tools, des attributs bidirectionnels des messages d'interface.
- Le traitement des données bidirectionnelles dans les flux de services, est basé sur des attributs bidirectionnels tels que le type de texte, l'orientation du texte, la permutation numérique et la 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 Enterprise Service Tools comprend la conversion des données entre des zones de message dont les attributs bidirectionnels sont différents.

De plus, le code généré par Enterprise Service Tools 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 Enterprise Service Tools des appels de routines de conversion bidirectionnelle en spécifiant les options de transformation bidi appropriées dans les assistants de génération Enterprise Service Tools et en éditant des liens entre les programmes générés et la bibliothèque de conversion bidirectionnelle appropriée, FEL.SFELLOAD.

Pour activer la prise en charge bidirectionnelle de la langue CICS, exécutez la procédure suivante :

1. Placez les modules de chargement FEL.SFELLOAD, 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.

Important : 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 dans CICS en utilisant la commande CEDA appropriée.

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

Messages IRZ de diagnostic pour Enterprise Service Tools

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
 - Mise à jour du JCL de la région CICS
 - Mise à jour de la définition système CICS de la région CICS
-

Le client Developer for z Systems contient un composant de génération de code appelé Enterprise Service Tools. Pour que le code généré par les outils Enterprise Service Tools émette des messages d'erreur de diagnostic, tous les modules IRZM* et IIRZ* de la bibliothèque de chargement FEL.SFELMOD doivent être mis à sa disposition. Les outils Enterprise Service Tools peuvent générer un code pour les environnements suivants :

- CICS
- IMS
- Lot MVS

Utilisez les instructions suivantes lorsque le code généré est exécuté dans une transaction CICS.

- Ajoutez tous les modules IRZM* et IIRZ* de FEL.SFELMOD à 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.
- Personnalisez et soumettez le travail IRZCSD pour mettre à jour la définition système CICS (CSD) pour la région CICS. Pour des instructions de personnalisation, voir la documentation du membre. IRZCSD se trouve dans FEL.#CUST.JCL sauf si vous avez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15.

Dans tous les autres cas, rendez disponibles tous les modules IRZM* et IIRZ* dans FEL.SFELMOD via STEPLIB ou 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.

Si vous utilisez STEPLIB, définissez 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.

Remarque :

- Le module FEL.SFELMOD(IRZPWSIO) est statistiquement lié au cours de la génération du code IMS MPP descendant. Par conséquent, le module ne doit pas être disponible au cours de l'exécution du code généré. Il doit être disponible uniquement lors de la compilation.
- Dans la version 9.0.1, FEL.SFELMOD(IRZPWSIO) et l'exemple de PL/I connexe FEL.SFELSAMP(IRZPWSH) incluent un membre déplacé de Developer for z Systems vers IMS version 12. Les composants sont renommés IMS.SDFSRESL(DFSPWSIO) et IMS.SDFSMPPL(DFSPWSH), respectivement.

FEKRNPLI, préprocesseur distant PL/I

Vous n'avez besoin d'aucune aide pour effectuer cette tâche de personnalisation mais les ressources ou les tâches de personnalisation spéciales suivantes sont requises.

Developer for z Systems fournit un exemple de préprocesseur PL/I qui peut être appelé par le client Developer for z Systems.

L'exemple de membre FEKRNPLI se trouve dans FEL.#CUST.CNTL, sauf si vous avez spécifié un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus d'informations, voir «Configuration personnalisée», à la page 15. Personnalisez l'exemple de commande

| exec de préprocesseur PL/I (REXX) FEL.#CUST.CNTL(FEKRNPLI) (comme cela est
| décrit dans le membre), et indiquez aux utilisateurs Developer for z Systems à quel
| emplacement se trouve la commande exec.

Chapitre 8. 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

DBGMGR : Gestionnaire de débogage

Lancez la tâche démarrée ou le travail utilisateur facultatif DBGMGR. Le serveur émet le message de console suivant si le démarrage a abouti, où `clientport` correspond au numéro de port utilisé pour la communication externe (client-hôte) et `hostport` au numéro de port utilisé pour les communications internes (limitées au système hôte).

```
AQECM001I Debug Manager startup complete (clientport/hostport)
```

Si le travail s'arrête avec un code retour 66, `FEL.SFELAUTH` n'a pas de droits APF.

Remarque : Lancez DBGMGR avant de continuer les autres tests IVP relatifs au débogage.

Vérification des services

L'installation de Developer for z Systems 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, par défaut `/usr/lpp/ibm/rdz/bin/`.

Tableau 15. Programmes de vérification de l'installation pour les services

fekfivpc	«Connexion à CARMA», à la page 100
fekfivps	«Connexion SCLMDT», à la page 101

Les tâches décrites dans les sections suivantes nécessitent des actions de votre part sur le système z/OS UNIX. Vous pouvez les effectuer en lançant la commande TSO **OMVS**. Pour revenir à TSO, utilisez la commande **exit**.

Une taille de région importante est nécessaire pour l'ID utilisateur qui exécute les IVP, car des fonctions telles que Java, qui demandent beaucoup de mémoire, vont être exécutées. Il est recommandé d'attribuer 131072 octets (128 mégaoctets) 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 z/OS Explorer et Developer for z Systems doivent être actives avant le lancement du test IVP.

Initialisation IVP

Tous les exemples de commandes de la présente section nécessitent la définition de certaines variables d'environnement. 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/zexpl
$ ./ivpinit
-- RSE_CFG set to /etc/zexpl -- based on current location
-- RSE_HOME set to /usr/lpp/ibm/zexpl -- defined in $RSE_CFG/rse.env
-- added product $PATH to PATH
-- PATH=/etc/zexpl:/bin:/usr/lpp/java/J6.0/bin:/usr/lpp/ibm/zexpl/bin:/usr/lpp/i
spf/bin:/bin:/usr/lpp/ibm/rdz/bin:/bin
```

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/ibm/rdz/bin/fekfivpc

De même, si **./ivpinit** n'est pas exécuté en premier, tous les scripts **fekfivp*** demandent l'emplacement du répertoire contenant les fichiers de configuration personnalisés

Connexion à CARMA

Vérifiez la connexion à CARMA en exécutant la commande suivante :

```
fekfivpc
```

La commande doit retourner un message de réussite :

Remarque : En cas d'échec de la procédure de vérification d'installation (IVP), vérifiez le contenu de **/tmp/fekfivpc.log**. Ce journal décrit les communications entre RSE et CARMA. Il peut contenir des informations clé pour trouver la cause première de l'échec.

fekfivpc présente plusieurs paramètres facultatifs ne dépendant pas de la position :

-noram

Par défaut, **fekfivpc** lance le premier RAM défini dans le fichier VSAM CRADEF. Si vous ne souhaitez pas tester le RAM pour certaines instances, par exemple dans le cas d'un RAM tiers répertorié en premier et qui requiert une entrée inattendu, vous pouvez utiliser l'argument de démarrage **-noram** pour omettre les étapes spécifiques au RAM du test IVP.

Connexion SCLMDT

Vérifiez la connexion de SCLM Developer Toolkit en exécutant la commande suivante :

fekfivps

La commande doit renvoyer un message de réussite.

fekfivps 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, \$TMPDIR/fekfivps.log, où \$TMPDIR est la valeur de la directive TMPDIR dans rse.env, dont la valeur par défaut est /tmp.

-debug

Le paramètre **-debug** crée une sortie détaillée des tests. N'utilisez pas cette option sans instruction du point service IBM.

Chapitre 9. Définitions de sécurité

Personnalisez et soumettez l'exemple de travail FELRACF, qui contient des exemples de commandes RACF permettant de créer les définitions de sécurité de base de Developer for z Systems. Personnalisez et soumettez l'exemple de travail AQERACF, qui contient des exemples de commandes RACF permettant de créer les définitions de sécurité du débogueur intégré.

FELRACF et AQERACF se trouvent dans FEL.#CUST.JCL sauf si vous avez indiqué un emplacement différent lors de la personnalisation et de la soumission du travail FEL.SFELSAMP(FELSETUP). Pour plus de détails, voir «Configuration personnalisée», à la page 15.

Voir *RACF Command Language Reference (SA22-7687)* pour plus d'informations sur les commandes RACF.

Configuration requise et liste de contrôle

Pour effectuer la configuration de la sécurité, l'administrateur de sécurité doit connaître les valeurs indiquées dans le tableau 16. Ces valeurs ont été définies dans les étapes précédentes d'installation et de personnalisation de Rational Developer for z Systems.

Tableau 16. Variables de configuration de la sécurité

Description	<ul style="list-style-type: none">Valeur par défautEmplacement de la réponse	Valeur
Qualificatif de haut niveau du produit Developer for z Systems	<ul style="list-style-type: none">FELInstallation SMP/E	
Qualificatif de haut niveau de personnalisation Developer for z Systems	<ul style="list-style-type: none">FEL.#CUSTFEL.SFELSAMP(FELSETUP), comme décrit dans «Configuration personnalisée», à la page 15.	
Nom de tâche démarrée du débogueur intégré	<ul style="list-style-type: none">DBGMGRFEL.#CUST.PROCLIB(DBGMGR), comme décrit dans «Modifications de PROCLIB», à la page 23	

La liste ci-après présente les actions requises pour effectuer la configuration de sécurité de base de Developer for z Systems. Comme indiqué dans les sections ci-après, différentes méthodes peuvent répondre à vos exigences, en fonction du niveau de sécurité requis.

- «Activation des paramètres et des classes de sécurité», à la page 104
- «Définition des tâches démarrées de Developer for z Systems», à la page 104

- «Définition du gestionnaire de débogage en tant que serveur sécurisé z/OS UNIX», à la page 105
- «Définition des bibliothèques contrôlées par le programme MVS pour le gestionnaire de débogage», à la page 105
- «Définition de l'accès au débogueur intégré», à la page 106
- «Définition des profils de fichier», à la page 106
- «Vérification des paramètres de sécurité», à la page 107

Activation des paramètres et des classes de sécurité

Developer for z Systems utilise différents mécanismes de sécurité pour fournir au client un environnement de système hôte sécurisé et contrôlé. Pour ce faire, plusieurs classes et paramètres de sécurité doivent être actifs, comme indiqué par les exemples de commande RACF suivants :

- Affichage des paramètres en cours
 - SETROPTS LIST
- Activation de la classe de fonction pour le débogueur intégré
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- Activation des définitions de tâche démarrée pour le débogueur intégré
 - SETROPTS GENERIC(STARTED)
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- Activation du contrôle de programme pour le débogueur intégré
 - 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 le logiciel de sécurité. Dans ce cas de figure, vous devez fusionner la définition * existante et la nouvelle définition **. Utilisez le profil **, comme indiqué dans le manuel dans *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 essayer ce contrôle de programmes avant de l'activer sur un système de production.

Définition des tâches démarrées de Developer for z Systems

Les exemples de commande RACF ci-dessous créent la tâche démarrée DBGMR, avec l'ID utilisateur protégé (STCDBM), ainsi que le groupe STCGROUP qui lui est affecté.

- ```
ADDGROUP STCGROUP OMVS(AUTOGID)
DATA('GROUP WITH OMVS SEGMENT FOR STARTED TASKS')
```
- ```
ADDUSER STCDBM DFLTGRP(STCGROUP) NOPASSWORD NAME('DEBUG MANAGER')
OMVS(AUTOUID HOME(/tmp) PROGRAM(/bin/sh) )
DATA('Rational Developer for z Systems')
```

```

RDEFINE STARTED DBGMGR.* DATA('DEBUG MANAGER')
STDATA(USER(STCDBM) GROUP(STCGROUP) TRUSTED(NO))
• SETROPTS RACLIST(STARTED) REFRESH

```

Remarque :

- Assurez-vous que les ID utilisateur des tâches démarrées sont protégés en indiquant le mot clé NOPASSWORD.
- La tâche démarrée du gestionnaire de débogage (DBGMGR) est uniquement utilisée par la fonction de débogueur intégré.

Définition du gestionnaire de débogage en tant que serveur sécurisé z/OS UNIX

Le débogueur intégré requiert un accès UPDATE au profil BPX.SERVER pour créer ou supprimer l'environnement de sécurité de l'unité d'exécution de débogage. Notez que l'utilisation de UID(0) pour ignorer cette exigence n'est pas prise en charge. Cette autorisation est requise uniquement lorsque la fonction de débogueur intégré facultative est utilisée.

- RDEFINE FACILITY BPX.SERVER UACC(NONE)
- PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(STCDBM)
- 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. Ce basculement peut avoir une incidence sur d'autres applications et opérations z/OS UNIX. Vous devez tester la sécurité avant de l'activer sur un système de production. Pour plus d'informations sur les différents niveaux de sécurité, voir le manuel *UNIX System Services Planning* (GA22-7800).

Définition des bibliothèques contrôlées par le programme MVS pour le gestionnaire de débogage

Les serveurs disposant des droits BPX.SERVER doivent être exécutés dans un environnement propre, contrôlé par un programme. Cette exigence implique que tous les programmes appelés par le gestionnaire de débogage doivent également être contrôlés par programme. Pour les bibliothèques de chargement MVS, le contrôle par programme est géré par votre logiciel de sécurité.

Le gestionnaire de débogage utilise des bibliothèques système, l'environnement d'exécution Language Environment et la bibliothèque de chargement Developer for z Systems' (ISP.SISPLoad).

- RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.LINKLIB'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('SYS1.CSSLIB'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('CEE.SCEERUN2'//NOPADCHK)
- RALTER PROGRAM ** UACC(READ) ADDMEM('FEL.SFELAUTH'//NOPADCHK)
- SETROPTS WHEN(PROGRAM) REFRESH

Remarque : N'utilisez pas le profil ** si le profil * existe déjà dans la classe PROGRAM. Le profil 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 **. Utilisez le profil **, comme indiqué dans la documentation *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 propres à un produit avec lequel Developer for z Systems interagit (IBM Explorer for z/OS, par exemple).

- Autre bibliothèque d'exécution REXX, pour SCLM Developer Toolkit
 - REXX.*.SEAGALT

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 :

- Bibliothèque d'exécution REXX, pour SCLM Developer Toolkit
 - REXX.*.SEAGLPA
- Developer for z Systems, pour CARMA
 - FEL.SFELLPA

Définition de l'accès au débogueur intégré

Les utilisateurs doivent disposer du droit d'accès en lecture à l'un des profils AQE.AUTHDEBUG.* répertoriés pour pouvoir utiliser le débogueur intégré afin de déboguer les programmes à l'état problème. Les utilisateurs autorisés à accéder au profil AQE.AUTHDEBUG.AUTHPGM peuvent également déboguer des programmes autorisés par APF. Remplacez la marque de réservation #apf par des ID utilisateur ou des noms de groupes RACF pour les utilisateurs admis pour déboguer des programmes autorisés.

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

Remarque : Les versions de IBM Rational Developer for System z antérieures à la version 9.1.1 utilisaient un autre profil de classe FACILITY, AQE.AUTHDEBUG.WRITEBUFFER, qui n'est plus utilisé. Il peut être supprimé si votre système hôte utilise uniquement IBM Rational Developer for System z version 9.1.1 ou une version ultérieure.

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 z Systems. 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 FEL.#CUST est celui relatif aux fichiers créés pendant le processus de personnalisation.

- - ADDGROUP (FEL) OWNER(IBMUSER) SUPGROUP(SYS1)
 - DATA('IBM Rational Developer for z Systems - HLQ STUB')
-

```

ADDSD 'FEL.*.**' UACC(READ)
DATA('IBM Rational Developer for z Systems')

```

-

```

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

```

-

```

SETROPTS GENERIC(DATASET) REFRESH

```

Remarque :

- Protégez FEL.SFELAUTH contre les mises à jour car ce fichier dispose de droits APF.
- Les exemples de commande utilisés dans la présente publication et dans le travail FELRACF supposent que l'EGN (Enhanced Generic Naming) est activé. Dans ce cas, le qualificatif ** peut être utilisé pour représenter tout nombre de qualificatifs dans la classe DATASET. Remplacez ** par * si l'EGN n'est pas activé dans votre système. Pour plus d'informations sur EGN, voir le manuel *Security Server RACF Security Administrator's Guide* (SA22-7683).

Certains des composants Developer for z Systems requièrent des profils de fichier de sécurité supplémentaires. Remplacez les marques de réservation #sysprog et #ram-developer par des ID utilisateur ou des noms de groupes RACF valides :

- Si la traduction des noms longs/abrégiés de SCLM Developer Toolkit est utilisée, les utilisateurs doivent disposer d'un accès en mise à jour (UPDATE) au mappage VSAM, FEL.#CUST.LSTRANS.FILE.

```

-
ADDSD 'FEL.#CUST.LSTRANS.*.**' UACC(UPDATE)
DATA('IBM Rational Developer for z Systems - SCLMDT')
-
PERMIT 'FEL.#CUST.LSTRANS.*.**' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
SETROPTS GENERIC(DATASET) REFRESH

```

- Les développeurs du gestionnaire CARMA RAM (Repository Access Manager) requièrent un accès en mise à jour (UPDATE) aux gestionnaires VSAM CARMA, FEL.#CUST.CRA*.

```

-
ADDSD 'FEL.#CUST.CRA*.*' UACC(READ)
DATA('IBM Rational Developer for z Systems - CARMA')
-
PERMIT 'FEL.#CUST.CRA*.*' CLASS(DATASET) ACCESS(ALTER) ID(#sysprog)
-
PERMIT 'FEL.#CUST.CRA*.*' CLASS(DATASET) ACCESS(UPDATE) ID(#ram-developer)
-
SETROPTS GENERIC(DATASET) REFRESH

```

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
- Tâches démarrées
 - LISTGRP STCGROUP OMVS

	– LISTUSER STCDBM OMVS
	– RLIST STARTED DBGMR.* ALL STDATA
	• Gestionnaire de débogage en tant que serveur sécurisé z/OS UNIX
	– RLIST FACILITY BPX.SERVER ALL
	• Bibliothèques contrôlées par le programme MVS pour le gestionnaire de débogage
	– RLIST PROGRAM ** ALL
	• Accès au débogueur intégré
	– RLIST FACILITY AQE.** ALL
	• Profils de fichier
	– LISTGRP FEL
	– LISTDSD PREFIX(FEL) ALL

Chapitre 10. 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 plus d'informations, voir les sections connexes du présent manuel.

- Si vous disposez d'une version antérieure de IBM Rational Developer for z Systems, sauvegardez les fichiers personnalisés associés avant de procéder à la mise à jour vers cette version d'IBM Rational Developer for z Systems.
- Si vous prévoyez d'exécuter plusieurs instances de Developer for z Systems, lisez "Exécution de plusieurs instances" dans le manuel *Guide de référence de la configuration hôte* (SC27-8578).
- Si votre scénario de migration couvre plus de deux éditions, vous devrez effectuer à nouveau les personnalisations, comme si aucune édition antérieure n'était présente.

Sauvegarde des fichiers précédemment configurés

Si vous disposez d'une version antérieure de Developer for z Systems, sauvegardez les fichiers personnalisés associés avant d'installer cette version d'IBM Developer for z Systems.

Les fichiers personnalisables de Developer for z Systems se trouvent dans les emplacements suivants :

- Toutes les anciennes versions
 - FEK.SFEKSAMP, certains membres sont copiés vers 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 vers /etc/rdz/ et /var/rdz/sc1mdt/* par l'exemple de travail FEKSETUP, où * correspond à CONFIG/, CONFIG/PROJECT/ et CONFIG/script/

Les configurations précédentes de Developer for z Systems documentent également les modifications apportées aux fichiers de configuration d'autres produits.

- Versions plus anciennes jusqu'à la version 9.0.1
 - SYS1.PARMLIB(IEASCVxx)
 - Toutes les anciennes versions
 - SYS1.PARMLIB(BPXPRMxx)
 - SYS1.PARMLIB(COMMNDxx)
 - SYS1.PARMLIB(LPALSTxx)
 - SYS1.PARMLIB(PROGxx)
 - (WLM)
- Affectez un environnement d'application pour une procédure mémorisée DB2.

Migration de la version 9.5 vers la version 9.5.1

Dans la version 9.5.1, les fonctions de moniteur de travaux RSE et JES ont été déplacées d'IBM Rational Developer for z Systems vers un autre produit, IBM Explorer for z/OS. IBM Explorer for z/OS est à présent un prérequis pour IBM Rational Developer for z Systems.

IBM recommande fortement de refaire toutes les personnalisations en utilisant les nouveaux exemples z/OS Explorer et Developer for z Systems afin d'être sûr d'utiliser les nouveaux noms de fichiers MVS et les nouveaux chemins z/OS UNIX.

Fichiers SMP/E :

- z/OS Explorer est un prérequis SMP/E pour Developer for z Systems. Cela signifie que z/OS Explorer et Developer for z Systems doivent être installés dans le même environnement CSI.
- Il n'existe aucune modification liée à SMP/E pour Developer for z Systems Host Utilities.

Fichiers MVS :

- z/OS Explorer utilise le qualificatif de haut niveau FEK et les noms de fichiers SFEK*. Aucune action de migration n'est requise ici.
- Developer for z Systems utilise un nouveau qualificatif de haut niveau, FEL, et de nouveaux noms de fichiers SFEL*. Recherchez les références en cours aux fichiers FEK et mettez-les à jour en utilisant FEL lorsque cela est nécessaire.
 - Définitions de sécurité dans les classes DATASET et PROGRAM
 - PROCLIB pour les tâches démarrées, les procédures de construction à distance et zUnit
 - Configuration pour démarrer CARMA
 - JCL pour traiter les gestionnaires VSAM CARMA
 - JCL pour traiter l'exit utilisateur LE
- Developer for z Systems Host Utilities utilise le qualificatif de haut niveau AKG et les noms de fichiers SAKG*. Aucune action de migration n'est requise.

Chemins z/OS UNIX :

- IBM Explorer for z/OS utilise de nouveaux chemins z/OS UNIX, /usr/lpp/ibm/zexpl, /etc/zexpl et /var/zexpl. Recherchez les références en cours aux anciens chemins (/usr/lpp/rdz, /etc/rdz et /var/rdz) et mettez-les à jour en utilisant les nouveaux chemins, le cas échéant.
 - PROCLIB pour les tâches démarrées
 - Références JCL à /etc/rdz
 - Notez que /etc/rdz/rsed.envvars est remplacé par /etc/zexpl/rse.env
 - Notez que les journaux, la fonction push-to-client (envoi au client par commande push) et ISPF WORKAREA ont été déplacés vers /var/zexpl
- Developer for z Systems utilise de nouveaux chemins z/OS UNIX, /usr/lpp/ibm/rdz et /etc/zexpl. Cependant, il utilise toujours le chemin /var/rdz. Recherchez les références en cours aux anciens chemins (/usr/lpp/rdz, /etc/rdz) et mettez-les à jour en utilisant les nouveaux chemins, le cas échéant.
 - JCL pour utiliser les exemples de convertisseurs de sortie zUnit
 - Notez que /etc/rdz/rsed.envvars est remplacé par /etc/zexpl/rdz.env
 - Notez que SCLMDT reste dans /var/rdz/sclmdt

- Developer for z Systems Host Utilities utilise le chemin `/usr/lpp/rdzutil`. Aucune action de migration n'est requise.

IBM Rational Developer for z Systems, FMID HHOP951

- L'emplacement d'installation SMP/E par défaut des composants MVS et z/OS UNIX est passé de FEK.SFEK* et `/usr/lpp/rdz/*` à FEL.SFEL* et `/usr/lpp/ibm/rdz/*`. Le qualificatif de haut niveau FEK est à présent utilisé par IBM Explorer for z/OS.
- L'emplacement de personnalisation par défaut des composants MVS et z/OS UNIX est passé de `/etc/rdz` à `/etc/zexpl`.
- Des fonctions ont été déplacées vers IBM Explorer for z/OS
 - L'utilitaire de configuration hôte est déplacé vers IBM Explorer for z/OS
 - Les activités de changement de RSE, du moniteur de travaux JES et de la passerelle ISPF existante restent
 - Les activités de changement de CARMA, du débogueur intégré et de SCLMDT sont supprimées
 - Le moniteur de travaux JES est déplacé vers IBM Explorer for z/OS
 - CNTL(FEJTSO)
 - PROCLIB(JMON)
 - SAMPLIB(FEJCNFG)
 - La fonction push-to-client (envoi au client par commande push) est déplacée vers IBM Explorer for z/OS
 - `/etc/rdz/pushtoclient.properties` est déplacé vers `/etc/zexpl/pushtoclient.properties`
 - `/etc/zexpl/pushtoclient.properties` est passé de `/var/rdz` vers `/var/zexpl`
 - RSE est déplacé vers IBM Explorer for z/OS
 - PROCLIB(RSED)
 - `/etc/rdz/rsed.envvars` est déplacé vers `/etc/zexpl/rse.env` (renommé)
 - `/etc/rdz/ISPF.conf` est déplacé vers `/etc/zexpl/ISPF.conf`
 - `/etc/rdz/rsecomm.properties` est déplacé vers `/etc/zexpl/rsecomm.properties`
 - `/etc/rdz/ssl.properties` est déplacé vers `/etc/zexpl/ssl.properties`
 - `/etc/zexpl/rse.env` contient uniquement les définitions de personnalisation des fonctions z/OS Explorer
 - Les outils système ont été déplacés vers IBM Explorer for z/OS
 - JCL(FEKRACF) contient uniquement les définitions de sécurité des fonctions z/OS Explorer
 - JCL(FEKSETUP) contient uniquement les définitions de personnalisation des fonctions z/OS Explorer
 - JCL(FEKSETUP) passé de `usr/lpp/rdz` à `/usr/lpp/ibm/zexpl`, de `/etc/rdz` à `/etc/zexpl` et de `/var/rdz` à `/var/zexpl`
- Certaines fonctions restent dans IBM Rational Developer for z Systems
 - Assistant de contenu C/C++
 - `/etc/rdz/include.conf` est déplacé vers `/etc/zexpl/include.conf`
 - CARMA
 - CNTL(CRASUBCA) passe de FEK.* à FEL.*, de SFEK* à SFEL*
 - CNTL(CRASUBMT) passe de FEK.* à FEL.*, de SFEK* à SFEL*

- JCL(CRA\$VCAD) passe de FEK.* à FEL.*, de SFEK* à SFEL*
- JCL(CRA\$VCAS) passe de FEK.* à FEL.*, de SFEK* à SFEL*
- JCL(CRA\$VDEF) passe de FEK.* à FEL.*, de SFEK* à SFEL*
- JCL(CRA\$VMSG) passe de FEK.* à FEL.*, de SFEK* à SFEL*
- JCL(CRA\$VSTR) passe de FEK.* à FEL.*, de SFEK* à SFEL*
- JCL(CRA#CIRX) passe de FEK.* à FEL.*
- JCL(CRA#UADD) passe de FEK.* à FEL.*
- JCL(CRA#UQRY) passe de FEK.* à FEL.*
- JCL(CRA#VPDS) passe de FEK.* à FEL.*
- JCL(CRA#VSLM) passe de FEK.* à FEL.*
- /etc/rdz/CRASRV.properties est déplacé vers /etc/zexpl/
CRASRV.properties
- /etc/rdz/crastart.conf est déplacé vers /etc/zexpl/crastart.conf
- /etc/rdz/crastart.endevor.conf est déplacé vers /etc/zexpl/
crastart.endevor.conf
- /etc/zexpl/crastart.conf passe de FEK.* à FEL.*, de SFEK* à SFEL*
- /etc/zexpl/crastart.endevor.conf passe de FEK.* à FEL.*, de SFEK* à
SFEL*
- Débogueur intégré
 - JCL(AQED3CEE) passe de FEK.* à FEL.*
 - Cela entraîne une mise à jour de l'emplacement par défaut de l'exit
utilisateur LE ; les utilisateurs devront donc sans doute mettre à jour la
syntaxe de l'exit utilisateur LE
 - JCL(AQED3CXT) passe de FEK.* à FEL.*
 - JCL(AQERACF) passe de FEK.SFEK* à FEL.SFEL*
 - PROCLIB(DBGMGR) passe de FEK.SFEK* à FEL.SFEL*
- Procédures de construction à distance
 - PROCLIB(ELAXFADT) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFASM) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFCOC) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFCOP) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFCOT) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFCP1) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFDCL) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFGO) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFLNK) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFPLP) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFPLT) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFPL1) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFPP1) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFTSO) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFUOP) passe de FEK à FEL (LODPRFX & STEPLIB)
 - PROCLIB(ELAXFSP) passe de FEK.* à FEL.*
 - PROCLIB(ELAXFSQL) passe de FEK.* à FEL.*
 - SQL(FEKTEP2) est déplacé vers SQL(FELTEP2)
 - SQL(FEKTIAD) est déplacé vers SQL(FELTIAD)

- Outils système
 - (nouveau) JCL(FELRACF) contient les définitions de sécurité des fonctions Developer for z Systems
 - (nouveau) JCL(FELSETUP) contient les définitions de personnalisation des fonctions Developer for z Systems
 - (nouveau) /etc/zexpl/rdz.env contient les définitions de personnalisation des fonctions Developer for z Systems qui étaient antérieurement situées dans /etc/rdz/rsed.envvars
- zUnit
 - PROCLIB(AZUZUNIT) passe de FEK à FEL (LODPRFX & STEPLIB)

Les utilisateurs auront sans doute besoin de mettre à jour l'appel de procédure pour pouvoir utiliser le nouveau chemin de Developer for z Systems, /usr/lpp/ibm/rdz/samples/zunit/*

Fichiers configurables

Le tableau 17 présente les fichiers Developer for z Systems qui ont été personnalisés dans la version 9.5.1. Les exemples de bibliothèque Developer for z Systems, FEL.SFELSAMP, FEL.SFELSAMPV et /usr/lpp/ibm/rdz/samples/, contiennent 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 FELSETUP copie tous les membres listés dans des fichiers et des répertoires différents, par défaut FEL.#CUST.* et /etc/zexpl/*.

Tableau 17. Personnalisations de la version 9.5.1

Membre/Fichier	Emplacement par défaut	Objectif	Remarques sur la migration
FELSETUP	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL permettant de créer des fichiers et des répertoires et de les remplir avec des fichiers personnalisables	Nouveau, la configuration est requise
DBGMGR	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour le gestionnaire de débogage	Utilisation de nouveaux noms de fichiers/chemins
AQEJCL	FEL.SFELSAMP [FEL.#CUST.PROCLIB(DBGMGR)]	Nom du membre DBGMGR	Voir Membre DBGMGR
ELAXF*	FEL.SFELSAMP [FEL.#CUST.PROCLIB]	JCL de génération de projets distants, etc.	Utilisation de nouveaux noms de fichiers/chemins
FELRACF	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour les définitions de sécurité	Nouveau, la configuration est requise
AQERACF	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour le gestionnaire de débogage	Utilisation de nouveaux noms de fichiers/chemins
CRA\$VMSG	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire CARMA	Utilisation de nouveaux noms de fichiers/chemins

Tableau 17. Personnalisations de la version 9.5.1 (suite)

Membre/Fichier	Emplacement par défaut	Objectif	Remarques sur la migration
CRA\$VDEF	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration du gestionnaire CARMA	Utilisation de nouveaux noms de fichiers/chemins
CRA\$VSTR	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées du gestionnaire CARMA	Utilisation de nouveaux noms de fichiers/chemins
CRA\$VCAD	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration CARMA du gestionnaire RAM CA Endeavor® SCM	Utilisation de nouveaux noms de fichiers/chemins
CRA\$VCAS	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées CARMA pour le gestionnaire RAM CA Endeavor® SCM	Utilisation de nouveaux noms de fichiers/chemins
CRASUBMT	FEL.SFELSAMP [FEL.#CUST.CNTL]	Liste de commandes du démarrage du traitement par lots CARMA	Utilisation de nouveaux noms de fichiers/chemins
CRASUBCA	FEL.SFELSAMP [FEL.#CUST.CNTL]	Liste de commandes de démarrage du traitement par lots CARMA pour CA Endeavor® SCM RAM	Utilisation de nouveaux noms de fichiers/chemins
CRACFG	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	Configuration d'interaction CARMA pour CA Endeavor® SCM RAM	Aucune
CRABCFG	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	Configuration des actions par lots CARMA pour CA Endeavor® SCM RAM	Aucune
CRABATCA	FEL.SFELSAMP [FEL.#CUST.CNTL]	JCL des actions de traitement par lots CARMA pour CA Endeavor® SCM RAM	Aucune
CRASCL	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	Modèle SCL pour CA Endeavor® SCM	Aucune

Tableau 17. Personnalisations de la version 9.5.1 (suite)

Membre/Fichier	Emplacement par défaut	Objectif	Remarques sur la migration
CRASHOW	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	Configuration CARMA pour CA Endevor® SCM RAM	Aucune
CRATMAP	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	Configuration CARMA pour CA Endevor® SCM RAM	Aucune
CRANDVRA	FEL.SFELPROC	REXX d'allocation CARMA pour CA Endevor® SCM RAM	Aucune
CRADYNDA	FEL.SFELPROC	REXX d'allocation CARMA pour CA Endevor® SCM RAM	Aucune
CRAALLOC	FEL.SFELPROC	REXX d'allocation CARMA	Aucune
CRA#VSLM	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire RAM SCLM	Utilisation de nouveaux noms de fichiers/chemins
CRA#ASLM	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer les fichiers du gestionnaire RAM SCLM	Utilisation de nouveaux noms de fichiers/chemins
CRA#VPDS	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire RAM PDS	Utilisation de nouveaux noms de fichiers/chemins
CRA#UADD	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL de fusion des définitions RAM	Utilisation de nouveaux noms de fichiers/chemins
CRA#UQRY	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL d'extraction des définitions RAM	Utilisation de nouveaux noms de fichiers/chemins
CRAXJCL	FEL.SFELSAMP [FEL.#CUST.ASM]	Exemple de code source pour le remplacement de IRXJCL	Aucune
CRA#CIRX	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL de compilation de CRAXJCL	Utilisation de nouveaux noms de fichiers/chemins
AQECSD	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour définir le débogueur intégré sur des régions CICS	Aucune
AQED3CEE	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer des modules d'exécution LE personnalisés	Utilisation de nouveaux noms de fichiers/chemins

Tableau 17. Personnalisations de la version 9.5.1 (suite)

Membre/Fichier	Emplacement par défaut	Objectif	Remarques sur la migration
AQED3CXT	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour créer un exit utilisateur LE	Utilisation de nouveaux noms de fichiers/chemins
FEKTEP2	FEL.SFELSAMP [FEL.#CUST.SQL]	Entrée de commande SQL utilisée par ELAXF*	Aucune
FEKTIAD	FEL.SFELSAMP [FEL.#CUST.JCL]	Entrée de commande SQL utilisée par ELAXF*	Aucune
AZUZUNIT	FEL.SFELSAMP [FEL.#CUST.PROCLIB]	Procédure JCL pour zUnit	Utilisation de nouveaux noms de fichiers/chemins
FEKRNPLI	FEL.SFELSAMP [FEL.#CUST.CNTL]	REXX pour appeler le compilateur PL/I à partir de la structure de précompilateur	Aucune
IRZCSD	FEL.SFELSAMP [FEL.#CUST.JCL]	JCL pour définir les outils de service d'entreprise (Enterprise Service Tools) sur CICS	Aucune
rdz.env	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	Variables d'environnement RSE	Les copies plus anciennes doivent être remplacées par celle-ci et les personnalisations doivent être effectuées une nouvelle fois.
CRASRV.properties	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	Fichier de configuration du gestionnaire CARMA	Aucune
crastart.conf	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	Fichier de configuration CARMA pour l'utilisation de CRASTART	Utilisation de nouveaux noms de fichiers/chemins
crastart.endevor.conf	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	Fichier de configuration CARMA pour l'utilisation de CRASTART pour CA Endevor® SCM RAM	Utilisation de nouveaux noms de fichiers/chemins
include.conf	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	Inclusions forcées pour l'assistant de contenu C/C++	Aucune

Le tableau 18, à la page 117 présente les fichiers personnalisables qui ont été déplacés vers z/OS Explorer dans la version 9.5.1.

Remarque : L'exemple de travail FEKSETUP copie tous les membres listés dans des fichiers et des répertoires différents, par défaut FEK.#CUST.* et /etc/zexpl/*.

Tableau 18. Personnalisations de z/OS Explorer version 9.5.1

Membre/Fichier	Emplacement par défaut	Objectif	Remarques sur la migration
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de créer des fichiers et des répertoires et de les remplir avec des fichiers personnalisables	Mis à jour pour supprimer les actions des fichiers qui ne sont plus utilisés et en ajouter pour les nouveaux fichiers
JMON	FEK.SFEKSAMP (FEJJJCL) [FEK.#CUST.PROCLIB]	JCL pour le moniteur de travaux JES	Aucune
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (JMON)]	Nom du membre JMON	Voir membre JMON
RSED	FEK.SFEKSAMP (FEKRSED) [FEK.#CUST.PROCLIB]	JCL pour le démon RSE	Aucune
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (RSED)]	Nom du membre RSED	Voir membre RSED
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour les définitions de sécurité	Données propres à Developer for z Systems supprimées
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Fichier de configuration du moniteur de travaux JES	Aucune
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL pour les soumissions TSO	Aucune
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de collecter les journaux	Utilisation de nouveaux noms de fichiers/chemins Données propres à Developer for z Systems supprimées
rse.env	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	Variables d'environnement RSE	Les copies plus anciennes de rse.envvars doivent être remplacées par celle-ci et les personnalisations doivent être effectuées une nouvelle fois.
ISPF.conf	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	Fichier de configuration de la passerelle ISPF	Utilisation de nouveaux noms de fichiers/chemins
ssl.properties	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	Fichier de configuration de la communication chiffrée RSE	Aucune

Tableau 18. Personnalisations de z/OS Explorer version 9.5.1 (suite)

Membre/Fichier	Emplacement par défaut	Objectif	Remarques sur la migration
rsecomm.properties	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	Fichier de configuration de trace RSE	Aucune
pushtoclient.properties	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	Envoi des informations au fichier de configuration du client	Utilisation de nouveaux noms de fichiers/chemins

IBM Rational Developer for z Systems Host Utilities, FMID HAKG951

- 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 AKG.* et à /usr/lpp/rdzutil/*.

Fichiers configurables

Le tableau 19 présente les fichiers qui ont été personnalisés dans la version 9.5.1. Les exemples de bibliothèque Developer for z Systems Host Utilities, AKG.SAKGSAMP et /usr/lpp/rdzutil/samples, contiennent davantage de membres personnalisables que ceux répertoriés ici, tels qu'un exemple de script de post-traitement pour la révision du code.

Remarque : L'exemple de travail AKGSETUP copie tous les membres répertoriés dans des fichiers différents, par défaut AKG.#CUST.*.

Tableau 19. Personnalisations d'Host Utilities version 9.0

Membre ou fichier	Emplacement par défaut	Fonction	Notes sur la migration
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL permettant de créer des fichiers et de les remplir avec des fichiers personnalisables	Aucune
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL permettant d'effectuer une couverture de code	Aucune
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL permettant d'effectuer une révision du code	Aucune
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL permettant d'ajouter du code tiers à la révision du code	Aucune

Migration de la version 9.1 vers la version 9.5

Ces remarques s'appliquent à la migration de la version de base 9.1 vers la version 9.5. Elles incluent les modifications déjà mentionnées dans le cadre de la maintenance de la version 9.1. Les modifications qui font partie du flux de maintenance, et qui sont donc peut-être déjà implémentées, sont marquées avec l'édition dans laquelle elles sont apparues.

Il est conseillé de remplacer un fichier `rsed.envvars` existant (par défaut dans `/etc/rdz`) par l'exemple fourni (par défaut dans `/usr/lpp/rdz/samples`) et de refaire les personnalisations.

IBM Rational Developer for z Systems, FMID HHOP950

- 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/*`.
- Depuis la version 9.5
 - Le gestionnaire de déploiement d'application a été retiré du produit. Les définitions suivantes ont été retirées :
 - SFEKLOAD(ADN*)
 - FEK.#CUST.JCL(ADN*)
 - FEK.#CUST.COBOLE(ADN*)
 - FEK.#CUST.ADNREPF0 [méthode d'accès VSAM créée par ADNVCRD]
 - FEK.#CUST.ADNMANF0 [méthode d'accès VSAM créée par ADNVMFST]
 - Les profils de sécurité de jeu de données FEK.#CUST.ADN* [établis par FEKRACF]
 - Les définitions CICS CSD [établies par ADNCSDAR, ADNCSDRS, ADNCSDTX, ADNCSDWS, ADNVMFST, AQECSD]
 - `usr/lpp/rdz/cicsts`
 - CARMA : Les membres personnalisables suivants ont changé :
 - CRACFG définit une nouvelle option, DYNAMIC-VB-DATASET-ALLOC
 - CRACFG définit une nouvelle option, DATASET-ALLOC-OVERRIDE
 - Enterprise Service Tools : nouveaux membres personnalisables
 - IRZCSD : mise à jour de la définition système CICS de la région CICS
 - Débogueur intégré : désormais, le débogage des transactions CICS requiert IP CICS Sockets.
 - PL/I : nouveaux membres personnalisables
 - FEKRNPLI : préprocesseur PL/I REXX
 - Procédures de construction à distance : les membres personnalisables suivants ont changé :
 - ELAXFDCL : ajout de SFEKAUTH à STEPLIB pour le débogueur intégré
 - ELAXFTSO : ajout de SFEKAUTH à STEPLIB pour le débogueur intégré
 - RSE : de nouvelles commandes de l'opérateur ont été ajoutées :
 - `F rsed,APPL=SEND 'message'[USER={ idutilisateur | (idutilisateur,idutilisateur,...)}]`
 - `F rsed,APPL=SEND 'message'[PID={ idprocessus | (idprocessus,idprocessus,...)}]`
 - `F rsed,APPL=SEND CLEARALL`

- RSE : de nouvelles directives facultatives ont été ajoutées à rsed.envvars:
 - CGI_CEATSO
 - CGI_CEATSO_KEEPLIVE
 - GSK_PROTOCOL_SSLV3
 - GSK_V3_CIPHER_SPECS
- RSE : de nouvelles directives facultatives ont été supprimées de rsed.envvars :
 - DSTORE_SSL_ALGORITHM
- RSE : des directives requises ont été marquées comme non modifiables dans rsed.envvars :
 - _RSE_HOST_CODEPAGE
- RSE : ajout de la prise en charge de l'envoi de messages à des clients avec une commande de l'opérateur, une commande z/OS UNIX ou TSO SEND
- RSE : ajout de la prise en charge de la passerelle ISPF interactive
- zUnit : les membres personnalisables suivants ont changé :
 - AZUZUNIT : ajout de la prise en charge de la couverture de code
- Depuis la version 9.1.1.1
 - Retour d'informations sur les erreurs : FEK.SFEKPROC(FEKFERRF) déplacé dans /usr/lpp/rdz/bin/fekferrf.rex
 - CARMA : Nouveaux membres personnalisables ajoutés :
 - CRASCL : Exemple de modèle CARMA SCL pour CA Endevor® SCM.
 - CARMA : Les membres personnalisables suivants ont changé :
 - CRACFG définit une nouvelle option, PACKAGE-EDITING-OPTION
 - CRACFG définit une nouvelle option, SCL-REQUIRED
 - CRACFG définit une nouvelle option, SCL-DATASET-TEMPLATE
 - Débogueur intégré : Les membres personnalisables suivants ont changé :
 - AQED3CXT nouveau niveau de débogage pour l'exit utilisateur LE
 - AQED3CEE garantit qu'un exit utilisateur correct est ajouté à l'exécution LE
 - RSE : le module LPA facultatif devient obligatoire.
 - FEKDSI doit désormais être disponible via LPA/LINKLIST/STEPLIB
 - RSE : De nouvelles commandes de l'opérateur ont été ajoutées :
 - F rsed,APPL=DEBUG {PROCESS | P},{STORAGE | STOR}[,PID=idprocessus]
 - F rsed,APPL=DEBUG {DAEMON | D},{STORAGE | STOR}
 - RSE : l'enregistrement du produit est obligatoire lorsque ce dernier a été acheté avec le code produit 5697-CDT, IBM Enterprise COBOL Suite for z/OS
 - SYS1.PARMLIB : mise à jour d'un nouveau membre PARMLIB :
 - IFAPRDxx : enregistrement du produit
- Depuis la version 9.1.1.0
 - CARMA : Les fichiers CRADEF et CRASTRS VSAM de CA Endevor® SCM RAM ont été mis à jour.
 - CARMA : Les membres personnalisables suivants ont changé :
 - CRACFG définit une nouvelle option, ALTERNATIVE-ALLOC
 - CRABCFG définit une nouvelle action, TRANSFER-ELEMENT
 - CRASUBCA a ajouté APIHJC
 - crastart.endevor.conf a ajouté APIHJC

- Débogueur intégré : prise en charge de la coexistence d'IBM Debug Tool for z/OS
- Débogueur intégré : Les membres personnalisables suivants ont changé :
 - AQECSD (mise à jour de la définition système CICS) définit de nouvelles ressources
 - AQERACF définit de nouveaux profils AQE.**
 - FEKRACF définit de nouveaux profils AQE.**
 - Débogueur intégré : des définitions CICS CSD requises ont été ajoutées
 - Files d'attente de données transitoires AQEM et AQED
 - Programme AQEW3Z3, AQEW3Z6, AQEL3Z3, AQEL3Z6, AQEL6Z3, AQEL6Z6
 - Débogueur intégré : certaines définitions CICS CSD ont été supprimées
 - Files d'attente de données transitoires CIGZ et CIBM
 - Programme AQEEV006
- Débogueur intégré : Les membres personnalisables suivants ont changé :
 - Pour la tâche démarrée AQESTC [DBGMGR], SFEKLPA se trouve désormais dans STEPLIB
- Débogueur intégré : l'appel du superviseur requis a été ajouté
 - Modifications de parmlib IEASVCxx pour AQESVC03 (chargement dynamique possible)
- Débogueur intégré : l'appel du superviseur facultatif a été supprimé
 - Modifications de parmlib IEASVCxx pour AQESVC01
- Débogueur intégré : des profils RACF requis ont été ajoutés
 - AQE.AUTHDEBUG.STDPGM
 - AQE.AUTHDEBUG.AUTHPGM
- Débogueur intégré : des profils RACF requis ont été supprimés
 - AQE.AUTHDEBUG.WRITEBUFFER
- Collecte de journaux : les clients peuvent collecter des journaux d'hôte
- RSE : ajout de la prise en charge d'une phrase de passe
- RSE : de nouvelles directives facultatives ont été ajoutées à rsed.envvars:
 - search.server.limit.scanned_objects
 - search,server.limit.errcount
- RSE : de nouvelles directives facultatives ont été supprimées de rsed.envvars :
 - enable.saf.check
 - RSE_DSICALL
 - search.server.limit.datasets

Fichiers configurables

Le tableau 20, à la page 122 présente les fichiers qui ont été personnalisés dans la version 9.5. Les exemples de bibliothèque Developer for z Systems, FEK.SFEKSAMP, FEK.SFEKSAMV et /usr/lpp/rdz/samples/, contiennent 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 20. Personnalisations de la version 9.5

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer des fichiers et des répertoires et de les remplir avec des fichiers personnalisables	Mis à jour pour supprimer les actions des fichiers qui ne sont plus utilisés et en ajouter pour les nouveaux fichiers
JMON	FEK.SFEKSAMP (FEJJJCL) [FEK.#CUST.PROCLIB]	JCL pour le moniteur de travaux JES	Aucune
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Nom du membre JMON	Voir membre JMON
RSED	FEK.SFEKSAMP (FEKRSED) [FEK.#CUST.PROCLIB]	JCL pour le démon RSE	Aucune
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Nom du membre RSED	Voir membre RSED
DBGMGR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour le gestionnaire de débogage	SFEKLPA ajouté à STEPLIB
AQEJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(DBGMGR)]	Nom du membre DBGMGR	Voir Membre DBGMBR
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL de génération de projets distants, etc.	SFEKAUTH ajouté à STEPLIB
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour les définitions de sécurité	Nouveaux profils AQE.**
AQERACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour le gestionnaire de débogage	Nouveaux profils AQE.**
FEKPBITS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de modifier les autorisations d'accès pour les fichiers journaux	Aucune
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Fichier de configuration du moniteur de travaux JES	Aucune
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL pour les soumissions TSO	Aucune
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire CARMA	Aucune
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration du gestionnaire CARMA	Aucune
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées du gestionnaire CARMA	Aucune

Tableau 20. Personnalisations de la version 9.5 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration CARMA du gestionnaire RAM CA Endeavor® SCM	Entrée VSAM modifiée
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées CARMA pour le gestionnaire RAM CA Endeavor® SCM	Entrée VSAM modifiée
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Démarrage par lots CARMA CLIST	Aucune
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Liste de commandes de démarrage du traitement par lots CARMA pour CA Endeavor® SCM RAM	DD APIHJC ajouté et nouvelles DD
CRACFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration d'interaction CARMA pour CA Endeavor® SCM RAM	Directives facultatives ajoutées
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration des actions par lots CARMA pour CA Endeavor® SCM RAM	Directives facultatives ajoutées
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL des actions par lots CARMA pour CA Endeavor® SCM RAM	Aucune
CRASCL	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Modèle SCL pour CA Endeavor® SCM	Nouveau, la configuration est facultative
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endeavor® SCM RAM	Aucune
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endeavor® SCM RAM	Aucune
CRANDVRA	FEK.SFEKPROC	REXX d'allocation CARMA pour CA Endeavor® SCM RAM	Allocation d'une nouvelle définition de données
CRADYNDA	FEK.SFEKPROC	REXX d'allocation CARMA pour CA Endeavor® SCM RAM	Nouveau, la configuration est facultative
CRAALLOC	FEK.SFEKPROC	REXX d'allocation CARMA	Aucune

Tableau 20. Personnalisations de la version 9.5 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire RAM SCLM	Aucune
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer les fichiers de RAM SCLM	Aucune
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du RAM PDS	Aucune
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de fusion des définitions RAM	Aucune
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL d'extraction des définitions RAM	Aucune
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Exemple de code source pour le remplacement de IRXJCL	Aucune
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de compilation de CRAXJCL	Aucune
AQECSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le débogueur intégré sur des régions CICS	Définition de nouvelles ressources
AQED3CEE	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer des modules d'exécution LE	Mis à jour pour s'assurer qu'un exit LE correct est utilisé
AQED3CXT	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer un exit utilisateur LE	Nouveau niveau de débogage pour l'exit LE
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	Entrée de commande SQL utilisée par ELAXP*	Aucune
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	Entrée de commande SQL utilisée par ELAXP*	Aucune
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	Procédure JCL pour zUnit	SFEKAUTH ajouté à STEPLIB
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	REXX pour appeler le compilateur PL/I à partir de la structure de précompilateur	Aucune
IRZCSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir les outils de service d'entreprise sur CICS	Nouveau, la configuration est facultative
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de collecter les journaux	Aucune

Tableau 20. Personnalisations de la version 9.5 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	Variables d'environnement RSE	Les copies plus anciennes doivent être remplacées par celle-ci et les personnalisations doivent être effectuées une nouvelle fois.
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la passerelle client TSO/ISPF	Aucune
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration du gestionnaire CARMA	Aucune
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRASTART	Aucune
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRASTART pour CA Endevor [®] SCM RAM	DD APIHJC ajouté et nouvelles DD
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Inclusions forcées pour l'assistant de contenu C/C++	Aucune
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la communication chiffrée RSE	Aucune
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de trace RSE	Aucune
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Envoi des informations au fichier de configuration du client	Aucune

IBM Rational Developer for z Systems Host Utilities, FMID HAKG950

- 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 AKG.* et à /usr/lpp/rdzutil/*.
- Depuis la version 9.1.1.1
 - Couverture de code : l'enregistrement du produit est obligatoire lorsque ce dernier a été acheté avec le code produit 5697-CDT, IBM Enterprise COBOL Suite for z/OS
 - Révision du code : l'enregistrement du produit est obligatoire lorsque ce dernier a été acheté avec le code produit 5697-CDT, IBM Enterprise COBOL Suite for z/OS
- Depuis la version 9.1.1.0
 - Révision du code : nouvelles options pour la procédure AKGCR

- SYSLIB : remplace PROPERTY qui nécessitait des données importées depuis le client
- Liste : liste des fichiers/membres et leur type de fichier (remplace PDS, MEMBERS et EXTMAP)
- JUNIT : nouveau format de sortie

Fichiers configurables

Le tableau 21 présente les fichiers qui ont été personnalisés dans la version 9.5. Les exemples de bibliothèque Developer for z Systems Host Utilities, AKG.SAKGSAMP et /usr/lpp/rdzutil/samples, contiennent davantage de membres personnalisables que ceux répertoriés ici, tels qu'un exemple de script de post-traitement pour la révision du code.

Remarque : L'exemple de travail AKGSETUP copie tous les membres répertoriés dans des fichiers différents, par défaut AKG.#CUST.*.

Tableau 21. Personnalisations de Host Utilities version 9.0

Membre ou fichier	Emplacement par défaut	Fonction	Notes sur la migration
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL pour créer des fichiers et de les remplir avec des fichiers personnalisables	Aucune
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL pour Code Coverage (couverture de code)	Aucune
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL pour Code Review (révision du code)	Nouvelles DD SYSLIB, LIST et JUNIT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL pour ajouter du code tiers à la révision du code	Aucune

Version 9.1 - Notes sur la migration

Les notes sur la migration suivantes sont spécifiques à IBM Rational Developer for z Systems version 9.1. Elles sont valables pour la migration de IBM Rational Developer for z Systems version depuis la version 9.1.0 vers la version 9.1.1, et s'ajoutent aux notes existantes sur la version 9.1.0.

Tous les changements répertoriés sont valides depuis la version 9.1.1.

IBM Rational Developer for z Systems, FMID HHOP910

- CARMA : Les fichiers CRADEF et CRASTRS VSAM de CA Endevor® SCM RAM ont été mis à jour.
- CARMA : Les membres personnalisables suivants ont changé :
 - CRACFG définit une nouvelle option, ALTERNATIVE-ALLOC
 - CRABCFG définit une nouvelle action, TRANSFER-ELEMENT
 - CRASUBCA a ajouté APIHJC
 - crastart.endevor.conf a ajouté APIHJC

- Débogueur intégré : prise en charge de la coexistence d'IBM Debug Tool for z/OS
- Débogueur intégré : Les membres personnalisables suivants ont changé :
 - La tâche démarrée de DBGMGR n'utilise plus le numéro d'appel du superviseur
 - AQECSD (mise à jour de la définition système CICS) définit de nouvelles ressources
 - AQERACF définit de nouveaux profils AQE.**
 - FEKRACF définit de nouveaux profils AQE.**
 - Débogueur intégré : des définitions CICS CSD requises ont été ajoutées
 - Files d'attente de données transitoires AQEM et AQED
 - Programme AQEW3Z3, AQEW3Z6, AQEL3Z3, AQEL3Z6, AQEL6Z3, AQEL6Z6
 - Débogueur intégré : certaines définitions CICS CSD ont été supprimées
 - Files d'attente de données transitoires CIGZ et CIBM
 - Programme AQEEV006
- Débogueur intégré : l'appel du superviseur requis a été ajouté
 - Modifications de parmlib IEASVCxx pour AQESVC03
- Débogueur intégré : l'appel du superviseur facultatif a été supprimé
 - Modifications de parmlib IEASVCxx pour AQESVC01
- Débogueur intégré : des profils RACF requis ont été ajoutés
 - AQE.AUTHDEBUG.STDPGM
 - AQE.AUTHDEBUG.AUTHPGM
- Débogueur intégré : des profils RACF requis ont été supprimés
 - AQE.AUTHDEBUG.WRITEBUFFER
- RSE : ajout de la prise en charge d'une phrase de passe
- RSE : de nouvelles directives facultatives ont été ajoutées à rsed.envvars:
 - search.server.limit.scanned_objects
 - search,server.limit.errcount
- RSE : de nouvelles directives facultatives ont été supprimées de rsed.envvars :
 - enable.saf.check
 - RSE_DSICALL
 - search.server.limit.datasets

IBM Rational Developer for z Systems Host Utilities, FMID HAKG910

- Révision du code : Nouvelles options pour la procédure AKGCR
 - SYSLIB : remplace PROPERTY qui nécessitait des données importées depuis le client
 - Liste : liste des fichiers/membres et leur type de fichier (remplace PDS, MEMBERS et EXTMAP)
 - JUNIT : nouveau format de sortie

Migration de la version 9.0 vers la version 9.1

Ces remarques concernent la migration d'une version de base 9.0 vers la version 9.1. Elles incluent les modifications déjà mentionnées dans le cadre de la maintenance de la version 9.0. Les modifications qui font partie du flux de maintenance, et qui sont donc peut-être déjà implémentées, sont marquées avec l'édition dans laquelle elles sont apparues.

Il est conseillé de remplacer un fichier `rsed.envvars` existant (par défaut dans `/etc/rdz`) par l'exemple fourni (par défaut dans `/usr/lpp/rdz/samples`) et de refaire les personnalisations.

IBM Rational Developer for z Systems, FMID HHOP910

- 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/*`.
- CARMA : Les membres personnalisables suivants ont changé :
 - CRASUBMT
 - CRASUBCA
- Personnalisation : Le JCL FEKSETUP traite désormais les nouveaux membres :
 - AQED3CEE : copié dans FEK.#CUST.JCL(AQED3CEE)
 - AQED3CXT : copié dans FEK.#CUST.JCL(AQED3CXT)
 - FEKPBITS : copié dans FEK.#CUST.JCL(FEKPBITS)
- RSE : De nouvelles directives facultatives ont été ajoutées à `rsed.envvars` :
 - `(_RSE_JAVAOPTS) -Denable.dDVIPA`
 - `(_RSE_JAVAOPTS) -Dlog.file.mode`
 - `(_RSE_JAVAOPTS) -Dlog.secure.mode`
 - `(_RSE_JAVAOPTS) -Denable.saf.check`
 - `(_RSE_JAVAOPTS) -Dkeep.stats.copy.local`
 - `(_RSE_JAVAOPTS) -Ddebug.miner.localhost`
 - `(_RSE_JAVAOPTS) -DDSTORE_USE_THREADED_MINERS`
- RSE : De nouvelles commandes de l'opérateur ont été ajoutées :
 - `F rsed,APPL=LOG {USER | AUDIT | NOSERVER | OWNER}`
- RSE : De nouveaux messages de console ont été ajoutés :
 - FEK220I = Les journaux hôte sont écrits sur {0}
 - FEK221E = {0} a été interrompu en raison de {1}
 - FEK301E = {0} (uid:{1}) ne détient pas le répertoire de {2} (file_owner uid:{3})
 - FEK302E = Le demandeur {0} de la commande LOGS n'est pas autorisé à accéder au profil de {1}
 - FEK303E = Le lien symbolique {0} ne peut pas être utilisé en tant que répertoire de journaux
 - FEK304W = Le paramètre non valide {0}, {1}, a été spécifié. C'est le mode par défaut, {3}, qui est utilisé.
 - FEK305E = L'ID, {0}, ne dispose pas des droits appropriés pour accéder à {1}.
- Sécurité : La prise en charge de fichiers journaux sécurisés a été ajoutée :
 - FEKPBITS : script permettant de modifier les droits d'accès pour l'infrastructure de journal existante

- Nouveau comportement : les journaux serveur sont désormais placés dans `$daemon.log/server`
- Nouveau comportement : les journaux sont créés avec des droits d'accès restreints
- Sécurité : La prise en charge de nouveaux profils de sécurité a été ajoutée :
 - FEK.CMD.LOGS.**
 - FEK.REJECT.*.UPDATES.system.group
- depuis la version 9.0.1.1
 - RSE : De nouvelles directives facultatives ont été ajoutées à `rsed.envvars` :
 - (`_RSE_JAVAOPTS`) -Ddebug.miner.autoreconnect
- depuis la version 9.0.1
 - CARMA : Le fichier CRADEF VSAM de CA Endevor SCM RAM a été mis à jour.
 - CARMA : Le module de chargement CRASTART, qui se trouve dans la zone permanente de programme, a été mis à jour, ce qui rend nécessaire une mise à jour de la zone permanente de programme.
 - CARMA : Support ajouté pour exécuter un exit utilisateur au démarrage de CARMA.
 - CARMA : Support ajouté pour les arguments de démarrage du traitement des gestionnaires RAM.
 - CARMA : Nouveaux membres personnalisables ajoutés :
 - CRAEXIT : Exemple d'exit utilisateur CARMA.
 - CRAALLOC : Exec d'allocation pour les appels CARMA personnalisés.
 - CRACFG : Fichier de configuration de l'utilisation du gestionnaire RAM CA Endevor SCM.
 - CARMA : Les membres personnalisables suivants ont changé :
 - CRASRV.properties
 - crastart.conf
 - crastart.endevor.conf
 - CRASUBMT
 - CRASUBCA
 - CRANDVRA
 - CARMA : Des instructions de définition de données supplémentaires ont été ajoutées pour le gestionnaire RAM CA Endevor® SCM dans `crastart.endevor.conf` et `CRASUBCA` :
 - CRAPARM, alloué par CRANDVRA
 - CRACFG
 - CARMA : Des instructions de définition de données supplémentaires ont été ajoutées pour le gestionnaire RAM non CA Endevor SCM dans `crastart.conf` et `CRASUBMT` :
 - CRAPARM, alloué par CRAALLOC
 - Personnalisation : Le JCL FEKSETUP traite désormais les nouveaux membres :
 - CRACFG : copié dans FEK.#CUST.PARMLIB(CRACFG)
 - AQEJCL : copié dans FEK.#CUST.PROCLIB(DBGMGR)
 - AQECSD : copié dans FEK.#CUST.JCL(AQECSD)
 - Débogueur intégré : Nouveau service facultatif

- Mises à jour parmlib IEASVCxx, LPALSTxx et PROGxx (APF et LINKLIST)
- DBGMR : JCL de la tâche démarrée
- AQECSD : Exemple de JCL pour mettre à jour le CSD CICS
- AQERACF : Exemple de JCL pour effectuer la configuration de la sécurité uniquement pour le débogueur intégré
- RSE : Membres PROCLIB mis à jour
 - ELAXFGO
- RSE : De nouvelles directives facultatives ont été ajoutées à rsecomm.properties :
 - USER
- RSE : Nouvelles commandes de l'opérateur
 - F rsed,APPL=TRACE {USER | SERVER | CLEAR}
- RSE : De nouvelles directives facultatives ont été ajoutées à rsed.envvars :
 - (_RSE_JAVAOPTS) -Dsearch.server.limit.timeout
 - (_RSE_JAVAOPTS) -Dkeep.all.logs
 - (_RSE_JAVAOPTS) -Daudit.users
 - RSE_UBLD_DD
 - RSE_UBLD_STEPLIB
- RSE : De nouveaux messages de console ont été ajoutés :
 - FEK910I = FEK107E = Disque presque plein dans {0}, {1} existant supprimé
- zUnit : Nouveaux arguments de démarrage facultatifs ajoutés :
 - CLOCALE / -l

Remarque : Pour simplifier la migration à partir d'une configuration Developer for z Systems existante sans le débogueur intégré, l'exemple de JCL FEK.SFEKSAMP(AQERACF) avec des commandes RACF est fourni uniquement pour les définitions de sécurité liées au débogueur intégré.

Fichiers configurables

Le tableau 22 présente les fichiers qui ont été personnalisés dans la version 9.1.0. Les exemples de bibliothèque Developer for z Systems, FEK.SFEKSAMP, FEK.SFEKSAMV et /usr/lpp/rdz/samples/, contiennent 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 22. Personnalisations de la version 9.1.0

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer des fichiers et des répertoires et de les remplir avec des fichiers personnalisables	Mis à jour pour supprimer les actions des fichiers qui ne sont plus utilisés et en ajouter pour les nouveaux fichiers
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JCL pour le moniteur de travaux JES	Aucun
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Nom du membre JMON	Voir membre JMON

Tableau 22. Personnalisations de la version 9.1.0 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	JCL pour le démon RSE	Aucun
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Nom du membre RSED	Voir membre RSED
DBGMGR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour le gestionnaire de débogage	Nouveau, la personnalisation est facultative
AQJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(DBGMGR)]	Nom du membre DBGMGR	Voir Membre DBGMBR
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL de génération de projets distants, etc.	ELAXFGO inclut SFEKAUTH dans STEPLIB
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour les définitions de sécurité	Nouvelle tâche démarrée DBGMGR
AQERACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour le gestionnaire de débogage	Nouveau, la personnalisation est facultative
FEKPBITS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de modifier les autorisations d'accès pour les fichiers journaux	Nouveau, la personnalisation est facultative
FEJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Fichier de configuration du moniteur de travaux JES	Aucun
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL pour les soumissions TSO	Aucun
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire CARMA	Aucun
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration du gestionnaire CARMA	Aucun
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées du gestionnaire CARMA	Aucun
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration CARMA du gestionnaire RAM CA Endevor® SCM	Entrée VSAM modifiée
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées CARMA pour le gestionnaire RAM CA Endevor® SCM	Entrée VSAM modifiée

Tableau 22. Personnalisations de la version 9.1.0 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Démarrage par lots CARMA CLIST	Ajout d'une prise en charge pour 8 arguments de démarrage, la DD CRAPARM et un exec d'allocation
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Liste de commandes de démarrage du traitement par lots CARMA pour CA Endevor® SCM RAM	Ajout d'une prise en charge pour 8 arguments de démarrage, la DD CRAPARM et de nouvelles DD
CRACFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration d'interaction CARMA pour CA Endevor® SCM RAM	Nouveau, la configuration est facultative
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration des actions par lots CARMA pour CA Endevor® SCM RAM	Aucun
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL des actions par lots CARMA pour CA Endevor® SCM RAM	Aucun
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endevor® SCM RAM	Aucun
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endevor® SCM RAM	Aucun
CRANDVRA	FEK.SFEKPROC	REXX d'allocation CARMA pour CA Endevor® SCM RAM	Ajout d'une prise en charge pour un exit utilisateur et allocation de nouvelle DD
CRAALLOC	FEK.SFEKPROC	REXX d'allocation CARMA	Nouveau, la configuration est facultative
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire RAM SCLM	Aucun
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer les fichiers de RAM SCLM	Aucun
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du RAM PDS	Aucun
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de fusion des définitions RAM	Aucun
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL d'extraction des définitions RAM	Aucun
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Exemple de code source pour le remplacement de IRXJCL	Aucun

Tableau 22. Personnalisations de la version 9.1.0 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de compilation de CRA]JCL	Aucun
AQECSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le débogueur intégré sur des régions CICS	Nouveau, la configuration est facultative
AQED3CEE	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer des modules d'exécution LE	Nouveau, la configuration est facultative
AQED3CXT	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer un exit utilisateur LE	Nouveau, la configuration est facultative
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le serveur CRD RESTful dans la région CICS principale	Aucun
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir les ID de transaction secondaire dans la région CICS	Aucun
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer d'autres ID transaction	Aucun
ADMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de compilation d'ADNMSGHS	Aucun
ADMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Exemple de code source pour le gestionnaire de message de pipeline	Aucun
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer le référentiel CRD	Aucun
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le serveur CRD du service Web dans la région CICS principale	Aucun
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le serveur CRD dans les régions CICS non principales	Aucun
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de mise à jour des valeurs par défaut de CRD	Aucun
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer et définir le référentiel de manifestes	Aucun
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	Entrée de commande SQL utilisée par ELAXF*	Aucun
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	Entrée de commande SQL utilisée par ELAXF*	Aucun
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	Procédure JCL pour zUnit	Aucun

Tableau 22. Personnalisations de la version 9.1.0 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	REXX pour appeler le compilateur PL/I à partir de la structure de précompilateur	Aucun
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de collecter les journaux	Aucun
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	Variables d'environnement RSE	Les copies plus anciennes doivent être remplacées par celle-ci et les personnalisations doivent être effectuées une nouvelle fois.
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la passerelle client TSO/ISPF	Aucun
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration du gestionnaire CARMA	Ajout d'une prise en charge pour un exit utilisateur
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRAFT	Ajout d'une prise en charge pour 8 arguments de démarrage, la DD CRAPARM et un exec d'allocation
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRAFT pour CA Endevor® SCM RAM	Ajout d'une prise en charge pour 8 arguments de démarrage, la DD CRAPARM et de nouvelles DD
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Inclusions forcées pour l'assistant de contenu C/C++	Aucun
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la communication chiffrée RSE	Aucun
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de trace RSE	Ajout d'une prise en charge pour le mot-clé USER
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Envoi des informations au fichier de configuration du client	Aucun

IBM Rational Developer for z Systems Host Utilities, FMID HAKG910

- 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 AKG.* et à /usr/lpp/rdzutil/*.
- Révision du code : Des instructions de définition de données (DD) supplémentaires sont ajoutées dans AKGCR
 - BEXPORT

– BIMPORT

Fichiers configurables

Le tableau 23 présente les fichiers qui ont été personnalisés dans la version 9.1. Les exemples de bibliothèque Developer for z Systems Host Utilities, AKG.SAKGSAMP et /usr/lpp/rdzutil/samples, contiennent davantage de membres personnalisables que ceux répertoriés ici, tels qu'un exemple de script de post-traitement pour la révision du code.

Remarque : L'exemple de travail AKGSETUP copie tous les membres répertoriés dans des fichiers différents, par défaut AKG.#CUST.*.

Tableau 23. Personnalisations de Host Utilities version 9.0

Membre ou fichier	Emplacement par défaut	Fonction	Notes sur la migration
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL pour créer des fichiers et de les remplir avec des fichiers personnalisables	Aucune
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL pour Code Coverage (couverture de code)	Aucune
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL pour Code Review (révision du code)	Nouvelles DD BIMPORT et BEXPORT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL pour ajouter du code tiers à la révision du code	Aucune

Migration de la version 8.5 vers la version 9.0

Ces remarques s'appliquent à la migration de la version de base 8.5 vers la version 9.0. Elles incluent les modifications déjà mentionnées dans le cadre de la maintenance de la version 8.5. Les modifications qui font partie du flux de maintenance, et qui sont donc peut-être déjà implémentées, sont marquées avec l'édition dans laquelle elles sont apparues.

IBM Rational Developer for z Systems, FMID HHOP900

- 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/*.
- CARMA : Les fichiers CRADEF et CRASTRS VSAM pour le gestionnaire RAM CA Endevor® SCM doivent être mis à jour afin d'utiliser la nouvelle prise en charge des actions par lots CA Endevor® SCM (à compter de la version 8.5.1).
- CARMA : Prise en charge ajoutée pour désactiver un RAM durant la création de CRADEF VSAM (depuis la version 8.5.1).
- CARMA : Prise en charge ajoutée pour les références de fichier non absolues dans CRASRV.properties (depuis la version 8.5.1).
- CARMA : Nouveaux exemples de membres ajoutés :
 - CRABJOBC : Carte de travail par défaut pour les actions par lots CA Endevor® SCM (depuis la version 8.5.1).
- CARMA : Les membres personnalisables suivants ont changé :

- CRASRV.properties (depuis la version 8.5.1)
- carma.startup.rex (depuis la version 8.5.1)
- CRA\$VCAD (depuis la version 8.5.1)
- CRA\$VDEF (depuis la version 8.5.1)
- CRABATCA (depuis la version 8.5.1)
- CRABCFG (depuis la version 8.5.1)
- CRANDVRA (depuis la version 8.5.1)
- CARMA : Des instructions de définition de données supplémentaires ont été ajoutées pour le gestionnaire RAM CA Endevor® SCM dans crastart.endevor.conf et CRASUBCA :
 - CRABJCLO, alloué par CRANDVRA (depuis la version 8.5.1)
 - ENHCEDIT, alloué par CRANDVRA (depuis la version 8.5.1)
- Personnalisation : Le JCL FEKSETUP traite désormais les nouveaux membres :
 - CRABJOBC : Copié dans FEK.#CUST.CNTL(CRABJOBC) (depuis la version 8.5.1)
 - ELAXFSP : Copié dans FEK.#CUST.PROCLIB(ELAXFSP) (depuis la version 9.0)
 - ELAXFSQL : Copié dans FEK.#CUST.PROCLIB(ELAXFSQL) (depuis la version 9.0)
 - FEKTEP2 : Copié dans FEK.#CUST.SQL(FEKTEP2) (depuis la version 9.0)
 - FEKTIAD : Copié dans FEK.#CUST.SQL(FEKTEP2) (depuis la version 9.0)
- Fault Analyzer Integration : La prise en charge de FAI a cessé. Cette modification est incompatible avec les clients plus anciens qui utilisent toujours FAI.
- Moniteur de travaux JES - De nouvelles commandes d'opérateur ont été ajoutées à la tâche démarrée JMON :
 - MODIFY USERS (depuis la version 8.5.1)
 - MODIFY -T{N | E | I | V} (depuis la version 8.5.1)
 - MODIFY -M{N | E | W | I | V} (depuis la version 8.5.1)
 - MODIFY TRACE {N | E | I | V} (depuis la version 9.0)
 - MODIFY MESSAGE {N | E | W | I | V} (depuis la version 9.0)
- Moniteur de travaux JES - De nouvelles directives facultatives ont été ajoutées à FEJCNFG :
 - LOOPBACK_ONLY (depuis la version 9.0)
- Moniteur de travaux JES - Des directives facultatives ont été retirées de FEJCNFG :
 - _BPXK_SETIBMOPT_TRANSPORT (depuis la version 9.0)
- Identification de problème : Le JCL FEKLOGS prend désormais en charge la spécification de plusieurs ID utilisateur pour rassembler les journaux utilisateur (à compter de la version 8.5.1).
- Identification de problème : Le JCL FEKLOGS utilise désormais DD REFORMAT pour recueillir les journaux reformatés pour une identification plus rapide des problèmes (à compter de la version 8.5.1).
- Identification de problème : Les membres personnalisables suivants ont changé :
 - FEKLOGS (depuis la version 8.5.1)
- RSE - De nouvelles commandes d'opérateur ont été ajoutées à la tâche démarrée RSED :
 - MODIFY DISPLAY OWNER,DATASET=dataset (depuis la version 9.0)

- MODIFY DEBUG GC,PID=pid (depuis la version 9.0)
- RSE : De nouvelles directives non personnalisables ont été ajoutées à `rsed.envvars` :
 - `_CMDSERV_BASE_HOME` (depuis la version 8.5.1)
 - `_CMDSERV_CONF_HOME` (depuis la version 8.5.1)
 - `_CMDSERV_WORK_HOME` (depuis la version 8.5.1)
 - `RSE_DSN_SFEKLOAD` (depuis la version 9.0)
 - `(_RSE_JAVAOPTS) -Dlock.info.timeout` (depuis la version 9.0)
 - `(_RSE_JAVAOPTS) -DDSTORE_INITIAL_SIZE` (depuis la version 9.0)
 - `(_RSE_JAVAOPTS) -DDSTORE_MAX_FREE` (depuis la version 9.0)
- RSE : De nouvelles directives requises ont été ajoutées à `rsed.envvars` :
 - `RSE_HLQ` (depuis la version 9.0)
- RSE : De nouvelles directives facultatives ont été ajoutées à `rsed.envvars` :
 - `(_RSE_JAVAOPTS) -DRSE_DSICALL` (depuis la version 8.5.1)
 - `(_RSE_JAVAOPTS) -DDISABLE_REMOTE_INDEX_SEARCH` (depuis la version 8.5.1)
 - `(_RSE_JAVAOPTS) -DDISABLE_TEXT_SEARCH` (depuis la version 9.0)
 - `(_RSE_JAVAOPTS) -Dsearch.server.limit.hits` (depuis la version 9.0)
 - `(_RSE_JAVAOPTS) -Dsearch.server.limit.datasets` (depuis la version 9.0)
 - `(_RSE_JAVAOPTS) -Dsearch.server.limit.lines` (depuis la version 9.0)
 - `(_RSE_JAVAOPTS) -DDSTORE_SSL_ALGORITHM` (depuis la version 9.0)
- RSE : La valeur par défaut des directives non personnalisables dans `rsed.envvars` a changé :
 - `(_RSE_JAVAOPTS) -DSPIRIT_EXPIRY_TIME` (depuis la version 9.0)
- RSE : La valeur par défaut des directives facultatives dans `rsed.envvars` a changé :
 - `(_RSE_JAVAOPTS) -Xms` (depuis la version 8.5.1)
 - `(_RSE_JAVAOPTS) -Xmx` (depuis la version 8.5.1)
 - `(_RSE_JAVAOPTS) -Dmaximum.clients` (depuis la version 8.5.1)
 - `(_RSE_JAVAOPTS) -Dmaximum.threads` (depuis la version 8.5.1)
 - `CGI_ISPPREF` (depuis la version 9.0)
- Sécurité : La prise en charge de nouveaux profils de sécurité a été ajoutée :
 - `FEK.USR.**` (depuis la version 8.5.1)

Fichiers configurables

Le tableau 24, à la page 138 présente les fichiers personnalisés dans la version 9.0. Les exemples de bibliothèque Developer for z Systems, `FEK.SFEKSAMP`, `FEK.SFEKSAMV` et `/usr/lpp/rdz/samples/`, contiennent davantage de membres personnalisables que ceux répertoriés ici, tels que du code source CARMA et des travaux pour leur compilation.

Les membres et fichiers suivants ne sont plus personnalisables ou ne sont plus utilisés :

- Tâche démarrée LOCKD
- Exemple de procédure mémorisée DB2 ELAXMSAM
- Exemple de JCL pour la procédure mémorisée DB2 ELAXMJCL

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 24. Personnalisations de la version 9.0

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer des fichiers et des répertoires et de les remplir avec des fichiers personnalisables	Mis à jour pour supprimer les actions des fichiers qui ne sont plus utilisés et en ajouter pour les nouveaux fichiers
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JCL pour le moniteur de travaux JES	Aucune
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	Nom du membre JMON	Voir membre JMON
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	JCL pour le démon RSE	Aucune
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	Nom du membre RSED	Voir membre RSED
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	JCL de génération de projets distants, etc.	ELAXFSP et ELAXFSQL sont nouveaux, ELAXFCOC et ELAXFCP1 ont été mis à jour pour prendre en charge Cobol version 5
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour les définitions de sécurité	Aucune
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Fichier de configuration du moniteur de travaux JES	De nouvelles directives facultatives ont été ajoutées. Des directives facultatives existantes ont été retirées.
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL pour les soumissions TSO	Aucune
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire CARMA	Aucune
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration du gestionnaire CARMA	Ajout d'une prise en charge pour exclure les gestionnaires RAM
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées du gestionnaire CARMA	Aucune
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM à la configuration CARMA du gestionnaire RAM CA Endeavor® SCM	Ajout d'une prise en charge pour exclure les gestionnaires RAM et entrée VSAM modifiée.

Tableau 24. Personnalisations de la version 9.0 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM aux informations personnalisées CARMA pour le gestionnaire RAM CA Endeavor® SCM	Entrée VSAM modifiée
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Démarrage par lots CARMA CLIST	Aucune
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	Liste de commandes de démarrage du traitement par lots CARMA pour CA Endeavor® SCM RAM	Aucune
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration des actions par lots CARMA pour CA Endeavor® SCM RAM	Nouvelles directives.
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	JCL des actions par lots CARMA pour CA Endeavor® SCM RAM	Ajout d'une prise en charge pour la carte de travail variable
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endeavor® SCM RAM	Aucune
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	Configuration CARMA pour CA Endeavor® SCM RAM	Aucune
CRANDVRA	FEK.SFEKPROC	REXX d'allocation CARMA pour CA Endeavor® SCM RAM	Nouvelles allocations DD
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du gestionnaire RAM SCLM	Aucune
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer les fichiers de RAM SCLM	Aucune
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer la méthode d'accès VSAM au message du RAM PDS	Aucune
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de fusion des définitions RAM	Aucune
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL d'extraction des définitions RAM	Aucune

Tableau 24. Personnalisations de la version 9.0 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	Exemple de code source pour le remplacement de IRXJCL	Aucune
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de compilation de CRAXJCL	Aucune
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le serveur CRD RESTful dans la région CICS principale	Aucune
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir les ID de transaction secondaire dans la région CICS	Aucune
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer d'autres ID transaction	Aucune
ADMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de compilation d'ADMSGHS	Aucune
ADMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	Exemple de code source pour le gestionnaire de message de pipeline	Aucune
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer le référentiel CRD	Aucune
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le serveur CRD du service Web dans la région CICS principale	Aucune
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour définir le serveur CRD dans les régions CICS non principales	Aucune
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL de mise à jour des valeurs par défaut de CRD	Aucune
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL pour créer et définir le référentiel de manifestes	Aucune
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	Entrée de commande SQL utilisée par ELAXF*	Nouveau, la personnalisation est facultative
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	Entrée de commande SQL utilisée par ELAXF*	Nouveau, la personnalisation est facultative
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	Procédure JCL pour zUnit	Aucune
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	REXX pour appeler le compilateur PL/I à partir de la structure de précompilateur	Aucune

Tableau 24. Personnalisations de la version 9.0 (suite)

Membre/Fichier	Emplacement par défaut	Fonction	Notes sur la migration
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	JCL permettant de collecter les journaux	Vérifications supplémentaires ajoutées. Toute personnalisation de fichiers plus anciens doit être à nouveau effectuée.
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	Variables d'environnement RSE	Les copies plus anciennes doivent être remplacées par celle-ci et les personnalisations doivent être effectuées une nouvelle fois.
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la passerelle client TSO/ISPF	Aucune
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration du gestionnaire CARMA	Ajout d'une prise en charge pour les values par défaut
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRASTART	Aucune
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration CARMA pour l'utilisation de CRASTART pour CA Endevor® SCM RAM	Aucune
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	Inclusions forcées pour l'assistant de contenu C/C++	Aucune
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de la communication chiffrée RSE	Aucune
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Fichier de configuration de trace RSE	Aucune
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	Envoi des informations au fichier de configuration du client	Aucune

IBM Rational Developer for z Systems Host Utilities, FMID HAKG900

Il n'y a pas de notes sur la migration, car la version 8.5 ne propose pas de fonction équivalente.

Fichiers configurables

Le tableau 25, à la page 142 présente les fichiers personnalisés dans la version 9.0. Les exemples de bibliothèque Developer for z Systems Host Utilities, AKG.SAKGSAMP

et /usr/lpp/rdzutil/samples, contiennent davantage de membres personnalisables que ceux répertoriés ici, tels qu'un exemple de script de post-traitement pour la révision du code.

Remarque : L'exemple de travail AKGSETUP copie tous les membres répertoriés dans des fichiers différents, par défaut AKG.#CUST.*.

Tableau 25. Personnalisations de Host Utilities version 9.0

Membre ou fichier	Emplacement par défaut	Fonction	Notes sur la migration
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL pour créer des fichiers et de les remplir avec des fichiers personnalisables	None
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL pour Code Coverage (couverture de code)	None
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	JCL pour Code Review (révision du code)	Nouvelles DD BIMPORT et BEXPORT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	JCL pour ajouter du code tiers à la révision du code	None

Chapitre 11. Informations de référence sur la configuration hôte

La présente section résume les informations contenues dans le document *IBM Rational Developer for z Systems - Guide de référence de la configuration hôte* (SC27-8578). Pour plus d'informations, consultez cette publication.

Description de Developer for z Systems

Les différents composants du système hôte Developer for z Systems interagissent pour offrir au client un accès à des services et des données d'hôte. La compréhension de la conception de ces composants peut vous aider à prendre les décisions les plus appropriées en matière de configuration.

Remarques relatives à la sécurité

| Developer for z Systems interagit avec d'autres composants hôte, cela ayant des
| implications au niveau de la sécurité.

Remarques relatives à TCP/IP

Developer for z Systems utilise TCP/IP pour fournir un accès grand système aux utilisateurs d'un poste de travail standard. Il l'utilise également pour les communications entre différents composants et d'autres produits.

Remarques à propos de WLM

Contrairement aux applications z/OS classiques, Developer for z Systems n'est pas une application monolithique qui peut être identifiée facilement au niveau du Workload Manager (WLM). Les différents composants qui constituent Developer for z Systems interagissent pour offrir au client un accès aux services et données du système hôte. Certains de ces services sont actifs dans plusieurs espaces adresse, ce qui aboutit à différentes classifications WLM.

Remarques relatives à l'envoi au client

| Developer for z Systems étend la fonction push-to-client z/OS Explorer ou le
| contrôle client basé sur l'hôte avec une prise en charge des définitions de projet.

Remarques relatives à CICSTS

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

Configuration d'AT-TLS

Cette section vous aide à résoudre certains problèmes susceptibles de se produire lors de la configuration d'AT-TLS (Application Transparent Transport Layer Security) ou pendant la vérification ou la modification d'une configuration existante.

Bibliographie

Publications référencées

Les publications suivantes sont référencées dans ce document :

Tableau 26. Publications référencées

Titre de la publication	Référence de la commande	Référence	Site Web de référence
Répertoire du programme d'IBM Rational Developer for z Systems	GI11-7314	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Répertoire du programme pour l'utilitaire hôte IBM Rational Developer for z Systems	GI11-7463	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systems - Guide de la configuration hôte	SC43-2904	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systems - Guide de référence de la configuration hôte	SC43-2902	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systems Common Access Repository Manager Developer's Guide	SC23-7660	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
SCLM Developer Toolkit - Guide d'administration	SC11-6464	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Explorer for z/OS Host Configuration Guide	SC27-8437	z/OS Explorer	
IBM Explorer for z/OS Host Configuration Reference	SC27-8438	z/OS Explorer	
Communications Server IP CICS Sockets Guide	SC31-8807	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Configuration Guide	SC31-8775	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP Configuration Reference	SC31-8776	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Guide	SA22-7591	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Initialization and Tuning Reference	SA22-7592	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL Reference	SA22-7597	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS Planning Workload Management	SA22-7602	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

Tableau 26. Publications référencées (suite)

Titre de la publication	Référence de la commande	Référence	Site Web de référence
MVS System Commands	SA22-7627	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Command Language Reference	SA22-7687	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF Security Administrator's Guide	SA22-7683	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Command Reference	SA22-7802	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services Planning	GA22-7800	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services User's Guide	SA22-7801	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Using REXX and z/OS UNIX System Services	SA22-7806	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

Les sites Web suivants sont référencés dans le présent document :

Tableau 27. Sites Web référencés

Description	Site Web de référence
Developer for z Systems IBM Knowledge Center	http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html
Bibliothèque Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Page d'accueil Developer for z Systems	http://www-03.ibm.com/software/products/en/developerforsystemz/
Developer for z Systems Recommended service	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335
Developer for z Systems - Demande d'amélioration	https://www.ibm.com/developerworks/support/rational/rfe/
Télécharger Apache Ant	http://ant.apache.org/

Publications d'information

Les publications suivantes peuvent s'avérer utiles pour vous aider à comprendre les incidents de configuration pour les composants de système hôte requis :

Tableau 28. 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	http://www.redbooks.ibm.com/
Guide du programmeur système pour : Workload Manager	SG24-6472	Redbook	http://www.redbooks.ibm.com/

Tableau 28. Publications d'information (suite)

Titre de la publication	Référence de la commande	Référence	Site Web de référence
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	http://www.redbooks.ibm.com/
TCP/IP Implementation Volume 4: Security and Policy-Based Networking	SG24-7535	Redbook	http://www.redbooks.ibm.com/
Tivoli Directory Server for z/OS	SG24-7849	Redbook	http://www.redbooks.ibm.com/

Remarques

La présente documentation a été rédigée pour des produits et services proposés aux Etats-Unis. Ce matériel peut être disponible auprès d'IBM dans d'autres langues. Vous pouvez toutefois devoir détenir une copie du produit ou une version du produit dans cette langue pour pouvoir y accéder.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans certains pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Toutefois, il appartient à l'utilisateur d'évaluer et de vérifier le fonctionnement de produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans la présente documentation. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
U.S.A.*

Pour le Canada, veuillez adresser votre courrier à :

*IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada*

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse IBM suivante :

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT". IBM DECLINE TOUTE RESPONSABILITE, EXPLICITE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE NON-CONTREFAÇON ET 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.

La présente documentation peut contenir des inexactitudes ou des coquilles. Elle est mise 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 peut utiliser ou diffuser, de toute manière qu'elle juge appropriée et sans aucune obligation de sa part, les informations qui lui sont 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 :

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

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 performances et les exemples de clients ne sont présentés qu'à des fins d'illustration. Les performances réelles peuvent varier en fonction des configurations et des conditions d'exploitation spécifiques.

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 programmes exemple sont fournis "EN L'ETAT" sans aucune 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.

Documentation sur l'interface de programmation

Marques

IBM, le logo IBM et ibm.com sont des marques d'International Business Machines Corp. dans de nombreux pays. Les autres noms de produits et de services 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 l'adresse www.ibm.com/legal/copytrade.shtml.

Conditions d'utilisation de la documentation du produit

Les droits d'utilisation relatifs à ces publications sont soumis aux dispositions suivantes :

Conditions d'utilisation

Ces dispositions s'ajoutent aux conditions d'utilisation du site Web IBM.

Usage personnel

Vous pouvez reproduire ces informations pour votre usage personnel, non commercial, sous réserve que toutes les mentions de propriété soient conservées. Vous ne pouvez distribuer ou publier tout ou partie de ces informations ou en faire des oeuvres dérivées, sans le consentement exprès d'IBM.

Usage commercial

Vous pouvez reproduire, distribuer et publier ces informations uniquement au sein de votre entreprise, sous réserve que toutes les mentions de propriété soient conservées. Vous ne pouvez reproduire, distribuer, afficher ou publier tout ou partie de ces publications en dehors de votre entreprise, ou en faire des oeuvres dérivées, sans le consentement exprès d'IBM.

Droits

Excepté les droits d'utilisation expressément accordés dans ce document, aucun autre droit, licence ou autorisation, implicite ou explicite, n'est accordé pour ces publications ou autres informations, données, logiciels ou droits de propriété intellectuelle contenus dans ces publications.

IBM se réserve le droit de retirer les autorisations accordées ici si, à sa discrétion, l'utilisation des publications s'avère préjudiciable à ses intérêts ou que, selon son appréciation, les instructions susmentionnées n'ont pas été respectées.

Vous ne pouvez télécharger, exporter ou réexporter ces informations qu'en total accord avec toutes les lois et règlements applicables dans votre pays, y compris les lois et règlements américains relatifs à l'exportation.

IBM N'OCTROIE AUCUNE GARANTIE SUR LE CONTENU DE CES PUBLICATIONS. LES PUBLICATIONS SONT LIVREES "EN L'ETAT" ET 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.

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 de nombreux pays. Les autres noms de produits et services peuvent être 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>.

Adobe et PostScript sont des marques d'Adobe Systems Incorporated.

Cell Broadband Engine - Sony Computer Entertainment Inc.

Rational est une marque d'International Business Machines Corporation et de Rational Software Corporation aux Etats-Unis et/ou dans certains autres pays.

Intel, Intel Centrino, Intel SpeedStep, Intel Xeon, Celeron, Itanium et Pentium sont des marques d'Intel Corporation aux Etats-Unis et/ou dans certains autres pays.

IT Infrastructure Library est une marque de Central Computer and Telecommunications Agency.

ITIL est une marque de Minister for the Cabinet Office.

Linear Tape-Open, LTO et Ultrium sont des marques de HP, IBM Corp. et Quantum.

Linux est une marque de Linus Torvalds.

Microsoft, Windows et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

Java ainsi que tous les logos et toutes les marques 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.



SC43-2904-00

