

IBM Rational Developer for z Systems
バージョン 9.5.1

構成ガイド



IBM Rational Developer for z Systems
バージョン 9.5.1

構成ガイド



お願い

本書をご使用になる前に、163 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM Rational Developer for z Systems バージョン 9.5.1 (プログラム番号 5724-T07、またはプログラム番号 5697-CDT の一部) および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC27-8577-00
IBM Rational Developer for z Systems
Version 9.5.1
Configuration Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

© Copyright IBM Corporation 2015, 2015.

目次

図	vii
---	-----

表	ix
---	----

本書について	xi
--------	----

本書の対象読者	xii
---------	-----

変更の要約	xii
-------	-----

文書内容の説明	xiv
---------	-----

計画	xv
----	----

基本的なカスタマイズ	xv
------------	----

共通アクセス・リポジトリ・マネージャー	
---------------------	--

(CARMA)	xv
---------	----

SCLM Developer Toolkit	xv
------------------------	----

統合デバッガー	xv
---------	----

ホスト・ベースのコード分析	xv
---------------	----

その他のカスタマイズ・タスク	xvi
----------------	-----

インストール検査	xvi
----------	-----

セキュリティ定義	xvi
----------	-----

マイグレーション・ガイド	xvi
--------------	-----

ホスト構成リファレンス	xvi
-------------	-----

IBM Rational Developer for z Systems ホスト構成ガイド 1

第 1 章 計画 3

マイグレーションに関する考慮事項	3
------------------	---

計画に関する考慮事項	4
------------	---

製品の概要	4
-------	---

スキルの要件	4
--------	---

時間の要件	5
-------	---

インストール前の考慮事項	5
--------------	---

インストール・ユーザー ID	5
----------------	---

必要な製品	6
-------	---

必要なリソース	6
---------	---

構成前の考慮事項	9
----------	---

ワークロード管理	9
----------	---

リソース使用量とシステム限度	10
----------------	----

必要な製品の必須の構成	10
-------------	----

ユーザー ID に関する考慮事項	10
------------------	----

サーバーに関する考慮事項	11
--------------	----

デプロイメント前の考慮事項	12
---------------	----

クライアント・チェックリスト	13
----------------	----

第 2 章 基本的なカスタマイズ 17

要件およびチェックリスト	17
--------------	----

カスタマイズのセットアップ	17
---------------	----

PARMLIB の変更	19
-------------	----

BPXPRMxx での z/OS UNIX 限度の設定	19
-----------------------------	----

COMMNDxx への開始タスクの追加	20
---------------------	----

IEASVCxx での SVC 定義	20
--------------------	----

IFAPRDxx での製品イネーブルメント	21
-----------------------	----

LPALSTxx での LPA 定義	21
--------------------	----

PROGxx での APF 許可	22
------------------	----

PROGxx での LINKLIST 定義	23
-----------------------	----

必要な LINKLIST 定義と LPA 定義	24
-------------------------	----

他の製品用の LINKLIST 定義	25
--------------------	----

PROCLIB の変更	26
-------------	----

DBGMGR、デバッグ・マネージャー開始タスク	26
-------------------------	----

ELAXF* リモート・ビルド・プロシージャ	27
------------------------	----

セキュリティ定義	29
----------	----

rdz.env、環境構成ファイル	30
------------------	----

特定のコンポーネント	34
------------	----

インストール検査	34
----------	----

第 3 章 共通アクセス・リポジトリ・マ ネージャー (CARMA) 35

要件およびチェックリスト	35
--------------	----

サーバー始動方式とアクティブ RAM の選択	36
------------------------	----

CARMA サーバーの始動	36
---------------	----

CRASTART	36
----------	----

バッチ実行依頼	36
---------	----

実動 RAM	37
--------	----

CA Endeavor SCM RAM	37
---------------------	----

CA Endeavor SCM パッケージ RAM	37
---------------------------	----

サンプル RAM	37
----------	----

PDS RAM	37
---------	----

スケルトン RAM	37
-----------	----

SCLM RAM	37
----------	----

事前構成済みの RAM とサーバー始動の組み合わ	
--------------------------	--

せ	37
---	----

CA Endeavor SCM RAM による CRASTART	38
----------------------------------	----

CARMA VSAM データ・セットの作成	38
-----------------------	----

CRASRV.properties のカスタマイズ	38
---------------------------	----

crastart.endeavor.conf のカスタマイズ	39
--------------------------------	----

(オプション) 追加の CA Endeavor SCM RAM の	
-----------------------------------	--

カスタマイズ	40
--------	----

サンプル RAM による CRASTART	41
-----------------------	----

CARMA VSAM データ・セットの作成	41
-----------------------	----

CARMA	41
-------	----

サンプル RAM	41
----------	----

CRASRV.properties のカスタマイズ	42
---------------------------	----

crastart.conf のカスタマイズ	42
-----------------------	----

(オプション) 追加のカスタム RAM のカスタマイ	
----------------------------	--

ズ	43
---	----

CA Endeavor SCM RAM によるバッチ実行依頼	43
--------------------------------	----

CARMA VSAM データ・セットの作成	44
-----------------------	----

CRASRV.properties のカスタマイズ	44
---------------------------	----

CRASUBCA のカスタマイズ	44
------------------	----

(オプション) 追加の CA Endeavor SCM RAM の	
-----------------------------------	--

カスタマイズ	46
--------	----

サンプル RAM によるバッチ実行依頼	46
---------------------	----

VSAM データ・セットの作成	46
CARMA	46
サンプル RAM	47
CRASRV.properties のカスタマイズ	47
CRASUBMT のカスタマイズ	47
(オプション) 追加のカスタム RAM のカスタマイズ	48
CARMA 構成詳細	49
CRASRV.properties、CARMA への RSE インターフェース	49
crastart*.conf、CRASTART サーバーの始動	53
CRASTART ログ・ファイルの収集	56
CRASUB*、バッチ実行依頼サーバー始動	56
CARMA VSAM データ・セット	59
CRADEF、構成データ・セット	59
CRAMSG、メッセージ・データ・セット	59
CRASTRS、カスタム・ストリング・データ・セット	59
CARMA Repository Access Manager (RAM)	60
CA Endeavor SCM RAM	60
CA Endeavor SCM パッケージ RAM	60
PDS RAM	61
スケルトン RAM	61
SCLM RAM	61
CRACFG、CRASCL、CRASHOW、および CRATMAP、CA Endeavor SCM RAM 構成ファイル	62
CRACFG、CA Endeavor SCM RAM と SCM の相互作用	62
CRASCL、CA Endeavor SCM RAM テンプレート SCL	62
CRASHOW、CA Endeavor SCM RAM のデフォルト・フィルター	63
CRATMAP、CA Endeavor SCM RAM のファイル拡張子のマッピング	63
CRANDVRA および CRADYNDA、CA Endeavor SCM RAM 割り振り exec	63
CRANDVRA、CA Endeavor SCM RAM 初期割り振り exec	64
CRADYNDA、CA Endeavor SCM RAM 再割り振り exec	64
CA Endeavor SCM RAM バッチ・アクション	64
CRABCFG、CA Endeavor SCM RAM バッチ・アクション構成	66
CRABATCA、CA Endeavor SCM RAM バッチ・アクション JCL	67
CRABJOBC、CA Endeavor SCM RAM バッチ・アクション JOB カード	67
CRAALLOC、カスタム RAM 割り振り exec	68
CARMA 戻りコード	68
(オプション) 複数の RAM のサポート	69
例	69
(オプション) カスタム割り振り exec	70
(オプション) CARMA ユーザー出口	71
(オプション) IRXJCL と CRAXJCL	72
CRAXJCL の作成	73

第 4 章 SCLM Developer Toolkit . . . 75

要件およびチェックリスト	75
前提条件	76
SCLMDT 用の ISPF.conf の更新	76
SCLMDT 用の rdz.env の更新	77
(オプション) ロング/ショート・ネーム変換	78
LSTRANS.FILE (ロング/ショート・ネーム変換 VSAM) の作成	78
rdz.env ロング/ショート・ネーム変換用に更新	80
(オプション) Ant のインストールおよびカスタマイズ	80
SCLMDT 用の SCLM の更新	81
WORKAREA および /tmp からの古いファイルの除去	82

第 5 章 統合デバッガー . . . 83

要件およびチェックリスト	83
統合デバッガーと COBOL v4	85
統合デバッガーおよびその他の言語環境ベースのデバッガー	85
統合デバッガーと暗号化通信	85
統合デバッガーと SYSPLEX	86
統合デバッガー構成パラメーター	86
統合デバッガーの parmlib 更新	86
統合デバッガーの SVC 更新	87
統合デバッガーの TCP/IP 更新	88
統合デバッガーのセキュリティ更新	88
統合デバッガーの言語環境プログラム (Language Environment) の更新	89
統合デバッガーの CICS アップデート	90
統合デバッガーの IMS の更新	91
統合デバッガーの DB2 ストアード・プロシージャの更新	91
統合デバッガーの ELAXF* の更新	92

第 6 章 ホスト・ベースのコード分析 . . 93

要件およびチェックリスト	93
コード・レビュー	93
コード・レビュー・プロセスの変更	94
コード・カバレッジ	94
コード・カバレッジの単一の呼び出し	94
コード・カバレッジの複数の呼び出し	95
コード・カバレッジの出力	95

第 7 章 その他のカスタマイズ・タスク 97

include.conf、C/C++ コンテンツ・アシスト用の強制インクルード	97
z/OS UNIX サブプロジェクト	98
REXEC または SSH のセットアップ	99
インクルード・プリプロセッサのサポート	99
Enterprise COBOL および PL/I での xUnit サポート	100
エンタープライズ・サービス・ツール・サポート	101
CICS 双方向言語サポート	102
エンタープライズ・サービス・ツールの診断用 IRZ メッセージ	103

FEKRNPLI、PL/I リモート・プリプロセッサ . . . 104	構成可能なファイル 135
第 8 章 インストール検査 105	バージョン 9.0 からバージョン 9.1 へのマイグレーション 137
開始タスクの検査 105	IBM Rational Developer for z Systems、FMID HHOP910 137
DBGMGR、デバッグ・マネージャー 105	構成可能なファイル 140
サービスの検査 105	IBM Rational Developer for z Systems Host Utilities、FMID HAKG910 146
IVP の初期化 106	構成可能なファイル 146
CARMA 接続 106	バージョン 8.5 からバージョン 9.0 へのマイグレーション 147
SCLMDT 接続 107	IBM Rational Developer for z Systems、FMID HHOP900 147
第 9 章 セキュリティー定義 109	構成可能なファイル 149
要件およびチェックリスト 109	IBM Rational Developer for z Systems Host Utilities、FMID HAKG900 156
セキュリティーの設定およびクラスをアクティブにする 110	構成可能なファイル 156
Developer for z Systems 開始タスクの定義 . . . 110	第 11 章 ホスト構成リファレンス . . . 157
セキュアな z/OS UNIX サーバーとしてデバッグ・マネージャーを定義する 111	Developer for z Systems について 157
デバッグ・マネージャーの MVS プログラム制御ライブラリーを定義する 111	セキュリティーに関する考慮事項 157
統合デバッガーへのアクセスの定義 112	TCP/IP に関する考慮事項 157
データ・セット・プロファイルを定義する . . . 113	WLM に関する考慮事項 157
セキュリティー設定の検査 114	クライアントへのプッシュ機能に関する考慮事項 . . . 157
第 10 章 マイグレーション・ガイド 115	CICSTS に関する考慮事項 157
マイグレーションに関する考慮事項 115	AT-TLS のセットアップ 158
前に構成したファイルのバックアップ 115	参考文献 159
バージョン 9.5 からバージョン 9.5.1 へのマイグレーション 116	参考資料 159
IBM Rational Developer for z Systems、FMID HHOP951 117	情報資料 160
IBM Rational Developer for z Systems Host Utilities、FMID HAKG951 126	特記事項 163
構成可能なファイル 126	プログラミング・インターフェース情報 165
バージョン 9.1 からバージョン 9.5 へのマイグレーション 126	商標 165
IBM Rational Developer for z Systems、FMID HHOP950 127	製品資料に関するご使用条件 165
構成可能なファイル 129	著作権使用許諾 166
IBM Rational Developer for z Systems Host Utilities、FMID HAKG950 135	商標の帰属表示 166



1. DBGMGR: デバッグ・マネージャー開始タスク	27	13. CRASUB*: バッチ実行依頼を使用した CARMA の始動	58
2. rdz.env: 環境構成ファイル	30	14. CRACFG - CA Endeavor SCM RAM と SCM の相互作用	62
3. CRASRV.properties: CA Endeavor SCM RAM に よる CRASTART	38	15. CRASHOW - CA Endeavor SCM RAM のデフ ォルト・フィルター	63
4. crastart.endeavor.conf: CA Endeavor SCM RAM に よる CRASTART	40	16. CRATMAP: CA Endeavor SCM RAM のデフ ォルト・フィルター	63
5. CRASRV.properties: サンプル RAM による CRASTART	42	17. CRABCFG: CA Endeavor SCM RAM バッチ・ アクション構成	66
6. crastart.conf: サンプル RAM による CRASTART	43	18. CRABATCA: CA Endeavor SCM RAM バッチ・ アクション JCL	67
7. CRASRV.properties - CA Endeavor SCM RAM によるバッチ実行依頼	44	19. CRABJOB: CA Endeavor SCM RAM バッチ・ アクション JOB カード	68
8. CRASUBCA: CA Endeavor SCM RAM によるバ ッチ実行依頼	45	20. SCLMDT 用の ISPF.conf の更新	77
9. CRASRV.properties: サンプル RAM によるバ ッチ実行依頼	47	21. SCLMDT 用の rdz.env の更新	78
10. CRASUBMT: サンプル RAM によるバッチ実 行依頼	48	22. FLM02LST: ロング/ショート・ネーム変換セッ トアップ JCL	79
11. CRASRV.properties - CARMA 構成ファイル	49	23. include.conf - C/C++ コンテンツ・アシスト用 の強制インクルード	98
12. crastart*.conf: CRASTART を使用した CARMA サーバーの始動	55		

表

1. 必要なリソース	6	15. サービス用の IVP	105
2. オプションのリソース	7	16. セキュリティー・セットアップの変動要素	109
3. 必須タスクに必要な管理者	8	17. バージョン 9.5.1 のカスタマイズ	120
4. オプションのタスクに必要な管理者	8	18. バージョン 9.5.1 z/OS Explorer のカスタマイズ	124
5. クライアント・チェックリスト: 必須部分	13	19. Host Utilities バージョン 9.0 のカスタマイズ	126
6. クライアント・チェックリスト: オプション部分	13	20. バージョン 9.5 のカスタマイズ	130
7. ロード・モジュールと機能のマッチング	23	21. Host Utilities バージョン 9.0 のカスタマイズ	135
8. サンプル ELAXF* プロシージャ	27	22. バージョン 9.1.0 のカスタマイズ	140
9. ELAXF* 高位修飾子チェックリスト	29	23. Host Utilities バージョン 9.0 のカスタマイズ	147
10. ELAXF*	29	24. バージョン 9.0 のカスタマイズ	150
11. デバッグ・マネージャーへの自動再接続	31	25. Host Utilities バージョン 9.0 のカスタマイズ	156
12. CARMA 戻りコード	68	26. 参考資料	159
13. SCLM 管理者チェックリスト	82	27. 参照される Web サイト	160
14. 動的統合デバッガー SVC 更新	88	28. 情報資料	160

本書について

本書では、IBM® Rational® Developer for z Systems™ の機能の構成について説明しています。ここには、ご使用の z/OS® ホスト・システム上に IBM Rational Developer for z Systems バージョン 9.5.1 を構成するための説明が記載されています。

これ以降、本書では以下の名前が使用されています。

- *IBM Explorer for z/OS* は *z/OS Explorer* と呼ばれます。
- *IBM Rational Developer for z Systems* は *Developer for z Systems* と呼ばれます。
- *IBM Rational Developer for z Systems* 統合デバッガーは統合デバッガーと呼ばれます。
- *IBM Rational Developer for z Systems Interface for CA Endevor® SCM* は *CA Endevor® SCM RAM* と呼ばれます。
- 共通アクセス・リポジトリ・マネージャー は、*CARMA* と省略されます。
- *Software Configuration and Library Manager Developer Toolkit* は *SCLM Developer Toolkit* と呼ばれ、*SCLMDT* と省略されます。
- *IBM z/OS Automated Unit Testing Framework* は *zUnit* と呼ばれます。
- *z/OS UNIX* システム・サービス は、*z/OS UNIX* と呼ばれます。
- 顧客情報管理システム (*CICS*) *Transaction Server* は、*CICSTS* と呼ばれ、*CICS®* と略されます。

本書は、Developer for z Systems のホスト・システム構成を説明した文書セットの一部です。これらの文書は、それぞれ特定の読者を対象としています。Developer for z Systems の構成を行うために、すべての資料に目を通す必要はありません。

- 「*IBM Rational Developer for z Systems* ホスト構成ガイド」(SC43-2913) は、すべての計画タスク、構成タスク、およびオプション (任意指定のものを含む) について詳しく説明し、代替シナリオを提供しています。
- 「*IBM Rational Developer for z Systems* ホスト構成リファレンス」(SC43-2912) は、Developer for z Systems の設計について説明し、Developer for z Systems、z/OS コンポーネント、および Developer for z Systems に関連するその他の製品 (WLM や CICS など) のさまざまな構成タスクに関する背景情報を提供しています。

本書の情報は、すべての IBM Rational Developer for z Systems バージョン 9.5 パッケージに適用されます。

本書の最新バージョンについては、「*IBM Rational Developer for z Systems* ホスト構成ガイド (SC88-5663)」(<http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC43-2913>) を参照してください。

インストールの説明、ホワイト・ペーパー、ポッドキャスト、およびチュートリアルを含む、資料一式の最新バージョンについては、IBM Rational Developer for z

本書の対象読者

本書は、IBM Rational Developer for z Systems バージョン 9.5.1 のインストールおよび構成を行うシステム・プログラマーを対象にしています。

本書では、製品を完全にセットアップするために必要な手順を、デフォルト以外のシナリオも含め、詳細に記載しています。構成の計画と実行に役立つ基本的な情報については、「*IBM Rational Developer for z Systems* ホスト構成リファレンス (SC43-2912)」に記載されています。本書を使用するには、z/OS UNIX システム・サービスおよび MVS™ ホスト・システムに精通している必要があります。

変更の要約

このセクションでは、「*IBM Rational Developer for z Systems* バージョン 9.5.1 ホスト構成ガイド」(SC43-2913-00) (2015 年 12 月に改訂) の変更点を要約します。

本文または図表に対して技術的な変更または追加が行われている場合には、その個所の左側に縦線を引いて示してあります。

新しい情報:

- バージョン 9.5.1 でのマイグレーション情報。
- `rdz.env` の新しいディレクティブおよび削除されたディレクティブ。

削除された情報:

バージョン 9.5.1 では、RSE および JES ジョブ・モニターの関連機能が IBM Rational Developer for z Systems から別の製品 IBM Explorer for z/OS に移動されました。この移動に伴い、関連する記述も一緒に移動されています。

- RSE 固有の情報がすべての章から削除されました。
- JES ジョブ・モニター固有の情報がすべての章から削除されました。
- TSO コマンド・サービス固有の情報がすべての章から削除されました。
- クライアントへのプッシュに関する情報がすべての章から削除されました。
- オペレーター・コマンドの説明が削除されました。

本書には、「*IBM Rational Developer for z Systems* バージョン 9.5 ホスト構成ガイド (SC88-5663-14)」に記載されていた情報が含まれています。

新しい情報:

- バージョン 9.5.1 でのマイグレーション情報。
- バージョン 9.5 でのマイグレーション情報。
- `rdz.env` の新しいディレクティブおよび削除されたディレクティブ。
- DEBUG STORAGE オペレーター・コマンドおよび SEND オペレーター・コマンド。
- メッセージの送信のサポート。

- 対話式 ISPF ゲートウェイのサポート。
- PL/I プリプロセッサ。

移動された情報:

- 統合デバッガーの情報は、別の章に移動されました。

削除された情報:

- Application Deployment Manager の提供はなくなりました。したがって、Application Deployment Manager に関する情報は、すべて削除されました。

本書には、「*IBM Rational Developer for z Systems* バージョン 9.1.1 ホスト構成ガイド」(SC88-5663-11) に記載されていた情報が含まれています。

新しい情報:

- 変更された DBGMGR 開始タスク定義。
- 統合デバッガーの新しいセキュリティー・セットアップ。
- Modify Display Process コマンドでの RSE スレッド・プールの新しい rejectLogon 状況。
- rdz.env 内の新しいディレクティブおよび除去されたディレクティブ。
- オプションの統合デバッガーに関する変更されたセットアップ手順

本書には、「*IBM Rational Developer for z Systems* バージョン 9.1 ホスト構成ガイド」(SC88-5663-14) に記載されていた情報が含まれています。

新しい情報:

- バージョン 9.1.0 でのマイグレーション情報。
- RSED 開始タスク定義の新しいオプション。
- rdz.env 内の新しいディレクティブおよび変更されたディレクティブ。
- 新しいオペレーター・コマンド。
- 統合デバッガーの新しいオプション。

本書には、「*IBM Rational Developer for z Systems* バージョン 9.0.1 ホスト構成ガイド」(SC88-5663-09) に記載されていた情報が含まれています。

新しい情報:

- オプションの統合デバッガーに関する追加のセットアップ手順

本書には、「*IBM Rational Developer for z Systems* バージョン 9.0.1 ホスト構成ガイド」(SC88-5663-08) に記載されていた情報が含まれています。

新しい情報:

- バージョン 9.0.1 でのマイグレーション情報。
- 新しいオプションの PARMLIB 更新。
- 新しいオプションの開始タスク。
- 新しいオペレーター・コマンド。
- rdz.env の新しいディレクティブおよび変更されたディレクティブ。
- CRASRV.properties 内の新しいディレクティブ。

- CARMA CA Endeavor® SCM RAM 用の新しい構成ファイル。
- CARMA の始動時に呼び出されるユーザー出口のサポート。

本書には、「*IBM Rational Developer for z Systems* バージョン 9.0 ホスト構成ガイド」(SC88-5663-07) に記載されていた情報が含まれています。

新しい情報:

- バージョン 9.0 でのマイグレーション情報。
- `rdz.env` の新しいディレクティブおよび変更されたディレクティブ。
- `FEJJCNFG` の新しいディレクティブおよび削除されたディレクティブ。
- 新しい `ELAXF*` `PROCLIB` メンバー。
- 新しい `JMON` オペレーター・コマンドおよび `RSED` オペレーター・コマンド。
- ホスト・ベースのコード分析に関する追加情報。

削除された情報:

- `LOCKD` 開始タスクが使用されなくなったため、ロック・デーモンに関する情報がすべて削除されました。
- サンプルの `DB2®` ストアード・プロシージャが新しい `ELAXF*` ビルド・プロシージャに置き換わりました。これに伴い、`DB2` ストアード・プロシージャに関するすべての情報が削除されました。
- サポートされていないリリースのマイグレーション情報が削除されました。

本書には、「*IBM Rational Developer for z Systems* バージョン 8.5.1 ホスト構成ガイド」(SC88-5663-06) に記載されていた情報が含まれています。

新しい情報:

- `rdz.env` 内の新しいディレクティブおよび変更されたディレクティブ。
- CA Endeavor® SCM RAM バッチ・アクションについて拡張されたサポート。
- インストール・ユーザー ID の要件。

本書には、「*IBM Rational Developer for z Systems* バージョン 8.5 ホスト構成ガイド」(SC88-5663-05) に記載されていた情報が含まれています。

新しい情報:

- `FEJJCNFG` 内の新しいオプションのディレクティブ。`FEJJCNFG`、`JES` ジョブ・モニター構成ファイルを参照してください。
- `rdz.env` 内の新しいオプションのディレクティブ。
- 新しいオプションの構成ファイル。
- 新しいオプションのコンポーネント。
- 新しいオプションのコンポーネント。
- 新しいオプションのコンポーネント。
- 新しいオペレーター・コマンドおよび拡張されたオペレーター・コマンド。

文書内容の説明

このセクションでは、本書に記載する情報を要約します。

計画

この章の情報をを使用して、Developer for z Systems のインストールとデプロイメントを計画してください。

基本的なカスタマイズ

以下のカスタマイズ・ステップは、基本的な Developer for z Systems セットアップ用です。

- 17 ページの『カスタマイズのセットアップ』
- 19 ページの『PARMLIB の変更』
- 26 ページの『PROCLIB の変更』
- 29 ページの『セキュリティー定義』
- 30 ページの『rdz.env、環境構成ファイル』

共通アクセス・リポジトリ・マネージャー (CARMA)

共通アクセス・リポジトリ・マネージャー (CARMA) は、Repository Access Manager (RAM) 向けのサーバー・プラットフォームです。RAM は、z/OS システム・ベースの Software Configuration Manager (SCM) 用のアプリケーション・プログラミング・インターフェース (API) です。SCM 機能を RAM に内包することにより、サポートされる SCM にクライアントがアクセスするときに、単一の API が使用できるようになります。

Developer for z Systems は事前に組み込まれた複数の RAM と、ユーザーが独自の RAM を作成するためのソース・コード・サンプルを提供しています。

IBM Rational Developer for z Systems Interface for CA Endevor® Software Configuration Manager は、Developer for z Systems クライアントが CA Endevor® SCM に直接アクセスできるようにします。

SCLM Developer Toolkit

SCLM Developer Toolkit は、SCLM の機能を拡張するために必要なツールをクライアントに提供します。SCLM 自体はホスト・ベースのソース・コード・マネージャーであり、ISPF に組み込まれています。

SCLM Developer Toolkit は、SCLM へのインターフェースとして機能する Eclipse ベースのプラグインを備えています。これにより、従来型コード開発におけるすべての SCLM プロセスへのアクセスを可能にし、メインフレーム上の SCLM と同期したワークステーション上で完全な Java™ および J2EE 開発を行えるようにサポートします。同期アクティビティーには、メインフレームからの J2EE コードのビルド、アセンブル、およびデプロイメントが含まれます。

統合デバッガー

Developer for z Systems 統合デバッガー・ホスト・コンポーネントを使用すると、クライアントは、言語環境プログラム (Language Environment® (LE)) ベースのさまざまなアプリケーションをデバッグできます。

ホスト・ベースのコード分析

Developer for z Systems クライアントと同様に、Developer for z Systems ホストは別の製品として提供されている実行中のコード分析ツール、IBM Rational Developer

for z Systems Host Utilities をサポートします。このコード分析をホスト上で実行する利点は、それを日常のバッチ処理として組み込むことができる点です。

以下のコード分析ツールを、ホスト上で使用することができます。

- コード・レビュー: コード・レビューは、異なる重大度レベルを持つ規則を使用して、ソース・コードをスキャンし、規則違反を報告します。
- コード・カバレッジ: 実行中のプログラムを分析し、実行可能な行の総数に対する実行済みの行についてのレポートを生成します。

その他のカスタマイズ・タスク

このセクションは、さまざまなオプションのカスタマイズ・タスクを結合したものです。必要なサービスを構成するには、該当するセクションの説明に従ってください。

Developer for z Systems 構成ファイルへのカスタマイズ

- include.conf、C/C++ コンテンツ・アシスト用の強制インクルード

他の製品への Developer for z Systems 関連のカスタマイズ :

- z/OS UNIX サブプロジェクト
- インクルード・プリプロセッサのサポート
- Enterprise COBOL および PL/I での xUnit サポート
- エンタープライズ・サービス・ツール・サポート
- CICS 双方向言語サポート
- エンタープライズ・サービス・ツールの診断用 IRZ メッセージ
- FEKRNPLI、PL/I リモート・プリプロセッサ

インストール検査

製品のカスタマイズの完了後、この章で説明するインストール検査プログラム (IVP) を使用して、主要な製品コンポーネントのセットアップが正常であることを検査できます。

セキュリティー定義

ここでは、サンプルの RACF® コマンドを使用して必須およびオプションのセキュリティー定義について説明します。

マイグレーション・ガイド

ここでは、本製品の以前のリリースと比較したインストールおよび構成上の変更点に重点を置いて説明します。また、このリリースへのマイグレーションに関する一般的なガイドラインも示します。

ホスト構成リファレンス

このセクションでは、「*Rational Developer for z Systems* ホスト構成リファレンス (SC43-2912)」に記載する情報を要約します。

IBM Rational Developer for z Systems ホスト構成ガイド

第 1 章 計画

Developer for z Systems のインストールとデプロイメントを計画する場合は、この章の情報と Developer for z Systems のソフトウェア要件 (前提条件および相互前提条件を含む) を使用してください。以下の事柄について説明しています。

- 『マイグレーションに関する考慮事項』
- 4 ページの『計画に関する考慮事項』
- 5 ページの『インストール前の考慮事項』
- 9 ページの『構成前の考慮事項』
- 12 ページの『デプロイメント前の考慮事項』
- 13 ページの『クライアント・チェックリスト』

Developer for z Systems のハードウェアとソフトウェアの要件 (前提条件および相互前提条件を含む) の完全なリストについては、Developer for z Systems 製品ページ (<http://www-03.ibm.com/software/products/en/developerforsystemz/>) の「**System Requirements**」タブを参照してください。

マイグレーションに関する考慮事項

115 ページの『第 10 章 マイグレーション・ガイド』では、本製品の以前のリリースと比較したインストールおよび構成の変更点を説明します。この情報を使用して、Developer for z Systems の現行リリースへのマイグレーションを計画してください。

注:

-

バージョン 9.5.1 で、Developer for z Systems は、IBM Explorer for z/OS の上にインストールされる製品になりました。サポート対象バージョンの IBM Developer for z/OS があることを確認してください。

- 以前に IBM Rational Developer for System z® を使用していた場合は、このバージョンの IBM Rational Developer for z Systems をインストールする前に、関連するカスタマイズ済みのファイルを保存してください。カスタマイズが必要だったファイルの概要については、115 ページの『第 10 章 マイグレーション・ガイド』を参照してください。
- Developer for z Systems の複数インスタンスを実行する予定の場合は、「*IBM Rational Developer for z Systems* 「ホスト構成リファレンス」 (SC43-2912)」の『複数のインスタンスの実行』を参照してください。

計画に関する考慮事項

製品の概要

Developer for z Systems は、ユーザーのパーソナル・コンピュータにインストールされたクライアント、および 1 台以上のホスト・システムにインストールされたサーバーで構成されます。クライアントとホストの両方とも IBM Explorer for z/OS の上にインストールされます。この資料では、z/OS ホスト・システムに関する情報を記載しています。

クライアントでは開発者は Eclipse ベースの開発環境を利用できます。この開発環境は、ホストに対するグラフィカル・インターフェースの統一を促進します。また、何よりも、ホストからクライアントへの作業のオフロードを可能にしてホスト上のリソースを節約します。

ホスト部分は、永続的にアクティブな各種のタスクと、臨時に開始されるタスクから構成されます。これらのタスクによって、クライアントは z/OS ホスト・システムのさまざまなコンポーネント (MVS データ・セット、TSO コマンド、z/OS UNIX のファイルとコマンド、ジョブ実行依頼、ジョブ出力など) を処理することができます。

Developer for z Systems では、z/OS Explorer が備える基本的なアクセス機能が強化されています。例えば、Developer for z Systems はホスト・システム上のサブシステムや他のアプリケーション・ソフトウェア (CICS、Software Configuration Manager (SCM) など) と対話できます (対話するように Developer for z Systems を構成し、それらの相互前提条件製品が使用可能な場合)。

Developer for z Systems 自体の情報、使用するシステム、前提条件製品、および相互前提条件製品と Developer for z Systems が対話する方法については、「*Rational Developer for z Systems* ホスト構成リファレンス (SC43-2912)」を参照してください。「*IBM Explorer for z/OS Host Configuration Reference* (SC27-8438)」には、Developer for z Systems の必要条件である z/OS Explorer についての同様の情報が記載されています。

Developer for z Systems に備わっている機能の詳細については、Developer for z Systems の Web サイト (<http://www-03.ibm.com/software/products/en/developerforsystemz/>) を参照するか、IBM 担当員にお問い合わせください。

スキルの要件

Developer for z Systems ホストをインストールするには、SMP/E のスキルが必要です。

Developer for z Systems を構成するために必要なシステム・プログラミングの権限と専門知識は標準的なレベルを超えているため、他のユーザーからの支援が必要になる場合があります。8 ページの表 3、および 8 ページの表 4 に、必須およびオプションのカスタマイズ・タスクに必要な管理者がリストされています。

時間の要件

Developer for z Systems ホスト・システム・コンポーネントのインストールと構成にかかる時間は、以下のようなさまざまな要因によって決まります。

- z/OS UNIX および TCP/IP の現行の構成
- 前提ソフトウェアと保守の可用性
- クライアントのインストールが完了しているユーザーの有無 (インストール済み環境をテストして発生の可能性のある問題を報告するため)

経験則から言うと、Developer for z Systems ホスト・システムをインストールして構成するプロセスが完了するには 1 日または 2 日必要です (必須製品である IBM Explorer for z/OS 製品のインストールと構成にかかる時間は含まれません)。これは、経験を積んだシステム・プログラマーが新規インストールを行う場合の時間要件です。問題が発生したり、必要とされるスキルが不足していたりする場合は、セットアップにさらに時間がかかります。

インストール前の考慮事項

SMP/E による製品のインストールの詳細な手順については、「*Program Directory for IBM Rational Developer for z Systems*」(GI88-4172) を参照してください。

Developer for z Systems サーバーは、単一システム志向であり、SYSPLEX 対応ではありません。SYSPLEX 内のサーバーを使用している場合、エンド・ユーザーの要求したデータ (データ・セット、ジョブ出力、z/OS UNIX ファイル) が、Developer for z Systems がインストールされているシステムにあることを確認する必要があります。他のシステムで Developer for z Systems のクローン作成を行う方法については、12 ページの『デプロイメント前の考慮事項』を参照してください。

複数インスタンスの Developer for z Systems を単一のホスト・システムで実行するには、「*Rational Developer for z Systems* 「ホスト構成リファレンス」(SC43-2912)」の『複数のインスタンスの実行』を参照してください。

Developer for z Systems がインストールされているファイル・システム (HFS または zFS) は、SETUID 許可ビットをオン (これはデフォルトです) にしてマウントする必要があります。NOSETUID パラメーターを指定してファイル・システムをマウントすると、ユーザーのセキュリティー環境が Developer for z Systems によって作成されず、クライアントの接続要求が拒否されます。z/OS Explorer、Java、および z/OS UNIX バイナリーをホストするファイル・システムにも同じことが当てはまります。

インストール・ユーザー ID

Developer for z Systems のインストール、またはメンテナンスのインストールに使用するユーザー ID には、最低でも以下の属性が必要です。

- TSO アクセス (通常の領域サイズ)。

注: インストール検査プログラム (IVP) を実行するユーザー ID には、大きな領域サイズが必要です。これは、大量のメモリーを必要とする機能 (Java など) が実行されるからです。領域サイズは、131072 キロバイト (128 メガバイト) 以上に設定してください。

- ユーザー ID とそのデフォルト・グループの両方に対して、セキュリティー・システム (RACF など) に定義された OMVS セグメント
 - HOME フィールドは、ユーザーに READ、WRITE、および EXECUTE アクセス権付きで割り振られたホーム・ディレクトリーを参照していなければなりません。
 - OMVS セグメント内の PROGRAM フィールドは、/bin/sh とするか、または、それ以外の有効な z/OS UNIX シェル (/bin/tcsh など) にしてください。
 - ユーザー ID のデフォルト・グループには GID が必要です。
- UID=0 または FACILITY クラスの BPX.SUPERUSER プロファイルの READ 権限。
- BPX.FILEATTR.APF または BPX.FILEATTR.PROGCTL プロファイルが FACILITY クラスで定義されている場合は、これらのプロファイルの READ アクセス権。
- /tmp ディレクトリー (または TMPDIR 環境変数で参照されるディレクトリー) への READ、WRITE、および EXECUTE アクセス権

必要な製品

Developer for z Systems には、製品が機能する前にインストールして操作可能にしておく必要がある、前提ソフトウェアがリストされています。また、Developer for z Systems の特定のフィーチャーをサポートするために相互に必要なソフトウェアのリストもあります。該当するフィーチャーを設計どおりに機能させるには、これらの必要なソフトウェアをインストールし、実行時に操作可能になるようにしておく必要があります。

Developer for z Systems のソフトウェア要件 (前提条件および相互前提条件を含む) の完全なリストについては、Developer for z Systems Web サイト (<http://www-03.ibm.com/software/products/en/developerforsystemz/>) の「System Requirements」タブを参照してください。

ご使用のサイトのポリシーによっては、そのために少し時間を要する場合もあります。これらの必要な製品が使用可能になるように、事前に計画を立ててください。以下に、基本セットアップの主要な必要条件を示します。

- z/OS 1.12 以上
- IBM Explorer for z/OS 3.0
- Java 6.0 以上 (31 または 64 ビット) の最新のサービス・リリース

必要なリソース

Developer for z Systems を使用するには、表 1 にリストしたシステム・リソースの割り振りが必要です。オプションのサービスには、7 ページの表 2 にリストしたリソースが必要です。これらのリソースを使用できるようにするための計画を立ててください。これは、ご使用のサイトのポリシーによっては、ソフトウェアの取得に一定程度の時間がかかる場合があるためです。

表 1. 必要なリソース

リソース	デフォルト値	情報
MVS ビルド・プロシージャ	ELAXF*	26 ページの『PROCLIB の変更』

表 2. オプションのリソース

リソース	デフォルト値	情報
CLPA を伴う IPL	適用外	83 ページの『第 5 章 統合デバッガー』
開始タスク	DBGMR	83 ページの『第 5 章 統合デバッガー』
LINKLIST データ・セット	FEL.SFELAUTH および FEL.SFELLOAD	<ul style="list-style-type: none"> 75 ページの『第 4 章 SCLM Developer Toolkit』 83 ページの『第 5 章 統合デバッガー』
LPA データ・セット	FEL.SFELLPA	<ul style="list-style-type: none"> 21 ページの『LPALSTxx での LPA 定義』 35 ページの『第 3 章 共通アクセス・リポジトリ・マネージャー (CARMA)』 83 ページの『第 5 章 統合デバッガー』
セキュリティー・プロファイル	AQE.**	83 ページの『第 5 章 統合デバッガー』
ホスト限定使用のポート範囲	使用可能な任意のポートを使用	<ul style="list-style-type: none"> 35 ページの『第 3 章 共通アクセス・リポジトリ・マネージャー (CARMA)』 93 ページの『第 6 章 ホスト・ベースのコード分析』
ホスト限定使用のポート範囲	5336	83 ページの『第 5 章 統合デバッガー』
クライアント/ホスト通信用のポート	5335 (統合デバッガーの場合)	83 ページの『第 5 章 統合デバッガー』
CICS CSD アップデート	複数値	83 ページの『第 5 章 統合デバッガー』
CICS JCL アップデート	<ul style="list-style-type: none"> FEL.SFELLOAD FEL.SFELAUTH 	<ul style="list-style-type: none"> 102 ページの『CICS 双方向言語サポート』 83 ページの『第 5 章 統合デバッガー』

Developer for z Systems を構成するために必要なシステム・プログラミングの権限と専門知識は標準的なレベルを超えているため、他のユーザーからの支援が必要になる場合があります。8 ページの表 3、および 8 ページの表 4に、必須およびオプションのカスタマイズ・タスクに必要な管理者がリストされています。

表 3. 必須タスクに必要な管理者

管理者	タスク	情報
システム	すべてのカスタマイズ・タスクには、一般的なシステム・プログラマー・アクションが必要である	適用外

表 4. オプションのタスクに必要な管理者

管理者	タスク	情報
システム	すべてのカスタマイズ・タスクには、一般的なシステム・プログラマー・アクションが必要である	適用外
セキュリティ	<ul style="list-style-type: none"> データ・セット・プロファイルを定義する プログラム制御データ・セットを定義する xxx* ジョブを実行依頼するための許可を定義する デバッグ用の CICS プロファイルを定義する 暗号化通信の証明書を追加する 開始タスクを定義する z/OS UNIX サーバー・プロファイルを定義する デバッグ用のプロファイルを定義する メッセージ送信用のプロファイルを定義する 	<ul style="list-style-type: none"> 「ホスト構成リファレンス」(SA88-4226) の『セキュリティに関する考慮事項』
TCP/IP	新しい TCP/IP ポートを定義する	「ホスト構成リファレンス」(SC43-2912) の『TCP/IP ポート』
SCLM	<ul style="list-style-type: none"> Java EE サポート用の SCLM 言語変換プログラムを定義する Java EE サポート用の SCLM タイプを定義する 	75 ページの『第 4 章 SCLM Developer Toolkit』

表 4. オプションのタスクに必要な管理者 (続き)

管理者	タスク	情報
CICS TS	<ul style="list-style-type: none"> • CICS 領域 JCL を更新する • CICS 領域 CSD を更新する • CICS グループを定義する • CICS トランザクション名を定義する • CICS に対してプログラムを定義する • CICS に対してデバッガーを定義する 	<ul style="list-style-type: none"> • 102 ページの『CICS 双方向言語サポート』 • 83 ページの『第 5 章 統合デバッガー』
WLM	<ul style="list-style-type: none"> • Developer for z Systems タスクに最終目標を割り当てる 	<ul style="list-style-type: none"> • 「ホスト構成リファレンス」(SC43-2912) の『WLM に関する考慮事項』
LDAP	クライアントへのプッシュ用のグループを定義する	「ホスト構成リファレンス」(SC43-2912) の『クライアントへのプッシュ機能に関する考慮事項』

構成前の考慮事項

Developer for z Systems 自体の情報、使用するシステム、前提条件製品、および相互前提条件製品と Developer for z Systems が対話する方法については、「*Rational Developer for z Systems* 「ホスト構成リファレンス」 (SC43-2912)」を参照してください。「*IBM Explorer for z/OS Host Configuration Reference* (SC27-8438)」には、Developer for z Systems の必要条件である z/OS Explorer についての同様の情報が記載されています。この情報は、現在のニーズや今後の成長をサポートするようなセットアップを作成する上で助けとなります。

ワークロード管理

従来の z/OS アプリケーションとは異なり、Developer for z Systems は、ワークロード・マネージャー (WLM) で容易に識別できる一体構造のアプリケーションではありません。Developer for z Systems は、クライアントがホスト・システムのサービスとデータにアクセスできるようにするために相互に作用する、複数のコンポーネントで構成されています。WLM 構成を計画するには、「ホスト構成リファレンス」(SC43-2912) の『WLM に関する考慮事項』を参照してください。

注: Developer for z Systems は、相互に、およびクライアントと通信する複数のタスクから構成されています。これらのタスクでは、さまざまなタイマーを使用してパートナーとの通信が失われたことを検出します。CPU の負荷が高いシステムや、Developer for z Systems に対するワークロード管理 (WLM) の設定が正しくないシステムでは、タイムアウトの問題が発生する可能性があります (タイムアウト・ウィンドウ内で CPU 時間が不足するため)。

リソース使用量とシステム限度

Developer for z Systems では、アドレス・スペースや z/OS UNIX のプロセスおよびスレッドなど、不定数のシステム・リソースが使用されます。これらのリソースの可用性は、さまざまなシステム定義によって制限されます。システム構成を計画するために主要なリソースの使用量を見積もるには、「ホスト構成リファレンス」(SC43-2912) の『チューニングに関する考慮事項』を参照してください。Developer for z Systems は、ストレージ・リソースの制限を大幅に変更すれば、31 ビットまたは 64 ビット・モードのいずれでも実行できます。

必要な製品の必須の構成

MVS システム・プログラマー、セキュリティー管理者、および TCP/IP 管理者に、必要な製品とソフトウェアがインストールされ、テストされ、機能しているかどうかを確認してください。必要であるのに見過ごされる可能性があるカスタマイズ・タスクのいくつかを、以下にリストします。

- すべての Developer for z Systems ユーザーが、Java ディレクトリーに対する READ アクセス権と EXECUTE アクセス権を持っている必要があります。
- z/OS UNIX サブプロジェクトのリモート (ホスト・ベースの) アクションを実行するには、ホスト・システム上で z/OS UNIX バージョンの REXEC または SSH がアクティブであることが必要です。

ユーザー ID に関する考慮事項

Developer for z Systems ユーザーのユーザー ID には、以下の属性が最低でも必要です。

- TSO アクセス (通常の領域サイズ)

注: インストール検査プログラム (IVP) を実行するユーザー ID には、大きな領域サイズが必要です。これは、大量のメモリーを必要とする機能 (Java など) が実行されるからです。領域サイズは、131072 キロバイト (128 メガバイト) 以上に設定してください。

- ユーザー ID とそのデフォルト・グループの両方に対して、セキュリティー・システム (RACF など) に定義された OMVS セグメント
 - HOME フィールドは、ユーザーに (READ、WRITE、および EXECUTE アクセス権付で) 割り振られたホーム・ディレクトリーを参照する必要があります。
 - OMVS セグメント内の PROGRAM フィールドは、/bin/sh とするか、または、それ以外の有効な z/OS UNIX シェル (/bin/tcsh など) にしてください。
 - ASSIZEMAX フィールドは設定せず、システム・デフォルトが使用されるようにしてください。
 - ユーザー ID は UID 0 を必要としません。

例 (コマンド **LISTUSER userid NORACF OMVS**):

USER=userid

OMVS INFORMATION

UID= 0000003200
HOME= /u/userid
PROGRAM= /bin/sh
CPUTIMEMAX= NONE

```

ASSIZEMAX= NONE
FILEPROCMAX= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE

```

- ユーザー ID のデフォルト・グループには GID が必要です。

例 (コマンド **LISTGRP group NORACF OMVS**):

```
GROUP group
```

```

OMVS INFORMATION
-----
GID= 0000003243

```

- z/OS Explorer および Developer for z Systems のインストールと構成に関するディレクトリーとファイル (デフォルトでは /usr/lpp/ibm/zexpl/*、/etc/zexpl/*、/var/zexpl/*、/usr/lpp/ibm/rdz/*、および /var/rdz/*) に対する READ および EXECUTE アクセス権
- z/OS Explorer WORKAREA ディレクトリー (デフォルトでは /var/zexpl/WORKAREA)、およびユーザー・ログ・ディレクトリー (デフォルトでは /var/zexpl/logs) に対する READ、WRITE、および EXECUTE アクセス権
- z/OS Explorer インストール・データ・セット (デフォルトでは FEK.SFEK* および FEL.SFEL*) に対する READ アクセス権
- /tmp ディレクトリーまたは TMPDIR 環境変数で参照されるディレクトリーへの READ、WRITE、および EXECUTE アクセス権

サーバーに関する考慮事項

z/OS Explorer および Developer for z Systems は、開始タスクまたはユーザー・ジョブである永続的にアクティブな複数のサーバーで構成されています。これらのサーバーは、要求されたサービスをそれら自体が提供するか、他のサーバー (z/OS UNIX スレッドまたはユーザー・ジョブなど) を始動してサービスを提供します。始動順序は特にありません。唯一の要件は、最初のユーザーが接続を試みるまでにサーバーが稼働していることです。z/OS Explorer および Developer for z Systems のサーバーとサービスで使用するセキュリティー・メカニズムは、それらが存在するデータ・セットとファイル・システムが保護されていることを前提とします。つまり、信頼されたシステム管理者のみがプログラム・ライブラリーと構成ファイルを更新できる状態でなければなりません。

- デバッグ・マネージャー (DBGMGR) は、デバッグ関連のサービスを提供します。
- JES ジョブ・モニター (JMON) は、JES に関連したすべてのサービスを提供します。JMON は z/OS Explorer に含まれています。
- リモート・システム・エクスプローラー (RSE) は、クライアントをホスト・システムに接続したり、特定のサービス用に他のサーバーを始動するなどの、コア・サービスを提供します。RSE は z/OS Explorer に含まれています。RSE は、次の 2 つの論理エンティティーから構成されます。
 - RSE デーモン (RSED)。これは接続セットアップを管理し、単一サーバー・モードでの実行を担当します。
 - RSE サーバー。これは個々のクライアント要求を処理します。

「ホスト構成リファレンス」(SC43-2912) の『TCP/IP ポート』で説明されているように、特定のホスト・システム・サービスとそれらのポートが、クライアントから接続可能でなければならない、ホスト・システムを保護するファイアウォールに定義されている必要があります。これ以外に Developer for z Systems が使用するポートは、すべて、ホスト専用トラフィックを持ちます。以下に、基本的な Developer for z Systems セットアップでの外部通信に必要なポートを示します。

- クライアント/ホスト通信セットアップ用 (TCP を使用) の RSE デーモン、デフォルト・ポート 4035。
- クライアント/ホスト通信用 (TCP を使用) の RSE サーバー。デフォルトでは、使用可能な任意のポートが使用されますが、使用可能なポートを指定の範囲に制限することができます。

デプロイメント前の考慮事項

Developer for z Systems は、インストール済み環境のクローンを別のシステムに作成することをサポートしているため、システムごとに SMP/E によるインストールを行う必要がありません。

以下のデータ・セット、ディレクトリー、およびファイルは、他のシステムへのデプロイメントに必須です。ファイルを別のロケーションへコピーしてある場合は、下記のリスト内の相当するファイルをそのファイルに置き換える必要があります。

注: 以下のリストでは、(z/OS Explorer を含む) 前提条件および相互必要条件となるソフトウェアのデプロイメントの必要性については考慮されていません。

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/*
- FEL.#CUST.JCL でのカスタマイズ・ジョブで生成された定義、データ・セット、ファイル、およびディレクトリー

Developer for z Systems Host Utilities

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

注:

- FEL および /usr/lpp/ibm/rdz は、Developer for z Systems のインストール時に使用された高位修飾子およびパスです。FEL.#CUST、/etc/zexpl および /var/rdz

は製品のカスタマイズ時に使用されるデフォルトの場所です (/etc/zexpl は z/OS Explorer 構成ディレクトリーです)。

- AKG および /usr/lpp/ibm/rdzutil は、Developer for z Systems Host Utilities のインストール時に使用された高位修飾子およびパスです。
- 製品の z/OS UNIX の部分をデプロイしやすくするために、Developer for z Systems を専用ファイル・システム (HFS または zFS) にインストールしてください (z/OS Explorer と一緒にインストールできます)。専用ファイル・システムを使用できない場合は、z/OS UNIX の tar コマンドなどのアーカイブ・ツールを使用して z/OS UNIX ディレクトリーをシステム間で転送してください。この方式は、Developer for z Systems のファイルとディレクトリーの属性 (プログラム制御など) を保存するためのものです。

Developer for z Systems インストール・ディレクトリーをアーカイブおよび復元するための以下のサンプル・コマンドについては、「UNIX System Services コマンド解説書」(SA88-8641) を参照してください。

- アーカイブ: `cd /SYS1/usr/lpp/ibm/rdz; tar -cSf /u/userid/rdz.tar`
- 復元: `cd /SYS2/usr/lpp/ibm/rdz; tar -xSf /u/userid/rdz.tar`

クライアント・チェックリスト

Developer for z Systems クライアントのユーザーは、クライアントを正しく機能させるために、特定のホスト・システム・カスタマイズの結果 (TCP/IP ポート番号など) を把握しておく必要があります。これらのチェックリストを使用して、必要な情報を収集してください。

表 5 のチェックリストは、必須のカスタマイズ・ステップに必要な結果を示しています。表 6 は、オプションのカスタマイズ・ステップに必要な結果を示しています。

表 5. クライアント・チェックリスト: 必須部分

カスタマイズ	値
(前提条件) RSE デーモン TCP/IP ポート番号。デフォルトは 4035 です。 このポートは z/OS Explorer の構成中に設定されます。	

表 6. クライアント・チェックリスト: オプション部分

カスタマイズ	値
ELAXF* プロシーチャーのロケーション (システム・プロシーチャー・ライブラリーに入っていない場合)。デフォルトは FEL.#CUST.PROCLIB です。 27 ページの『ELAXF* リモート・ビルド・プロシーチャー』で JCLLIB に関する注を参照してください。	
ELAXF* プロシーチャーのプロシーチャー名またはステップ名 (変更された場合)。 27 ページの『ELAXF* リモート・ビルド・プロシーチャー』で、それらの変更に関する注を参照してください。	

表 6. クライアント・チェックリスト: オプション部分 (続き)

カスタマイズ	値
<p>AKGCR プロシージャーのロケーション (システム・プロシージャー・ライブラリーに入っていない場合)。デフォルトは AKG.#CUST.PROCLIB です。</p> <p>93 ページの『コード・レビュー』で、JCLLIB に関する注を参照してください。</p>	
<p>AKGCC プロシージャーのロケーション (システム・プロシージャー・ライブラリーに入っていない場合)。デフォルトは AKG.#CUST.PROCLIB です。</p> <p>94 ページの『コード・カバレッジ』で、JCLLIB に関する注を参照してください。</p>	
<p>FEKRNPLI インクルード・プリプロセッサ exec ステートメントのロケーション。デフォルトは FEL.#CUST.CNTL です。</p> <p>99 ページの『インクルード・プリプロセッサのサポート』を参照してください。</p>	
<p>デバッガー・ロード・モジュールのロケーション (LINKLIST に入っていない場合)。デフォルトは FEL.SFELAUTH です。83 ページの『第 5 章 統合デバッガー』を参照してください</p>	
<p>単体テスト・ロード・モジュールのロケーション (rdz.env の LINKLIST または STEPLIB に入っていない場合)。デフォルトは FEL.SFELLOAD です。</p> <p>100 ページの『Enterprise COBOL および PL/I での xUnit サポート』を参照してください。</p>	
<p>AZUZUNIT プロシージャーのロケーション (システム・プロシージャー・ライブラリーに入っていない場合)。デフォルトは FEL.#CUST.PROCLIB です。</p> <p>100 ページの『Enterprise COBOL および PL/I での xUnit サポート』で、JCLLIB に関する注を参照してください。</p>	
<p>単体テスト出力のフォーマット設定に使用されるサンプル XML ファイル *.xsd および *.xsl のロケーション。デフォルトは /usr/lpp/ibm/rdz/samples/zunit/xsd および /usr/lpp/ibm/rdz/samples/zunit/xsl です。</p> <p>100 ページの『Enterprise COBOL および PL/I での xUnit サポート』を参照してください。</p>	
<p>(相互前提条件) Host Connect Emulator の TN3270 ポート番号。デフォルトは 23 です。</p> <p>「ホスト構成リファレンス」(SC43-2912) の『TCP/IP ポート』を参照してください。</p>	
<p>(相互前提条件) REXEC ポート番号または SSH ポート番号。デフォルトはそれぞれ 512 および 22 です。</p> <p>98 ページの『z/OS UNIX サブプロジェクト』を参照してください。</p>	
<p>(相互前提条件) Debug Tool サーバーのポート番号 (デフォルトなし)。</p> <p>『(オプション) DB2 および IMS™ のデバッグのサポート』を参照してください。</p>	

表 6. クライアント・チェックリスト: オプション部分 (続き)

カスタマイズ	値
CARMA RAM サンプルの SFELSAMP サンプル・ライブラリーのロケーション。デフォルトは FEL.SFELSAMP です。 「 <i>CARMA Developer's Guide</i> 」(SC23-7660) を参照してください。	
CARMA SCLM RAM データ・セット割り振りの CRA#ASLM JCL のロケーション。デフォルトは FEL.#CUST.JCL です。 61 ページの『SCLM RAM』で CRA#ASLM に関する注を参照してください。	

第 2 章 基本的なカスタマイズ

以下のカスタマイズ手順は、さまざまな Developer for z Systems サービスで共通のものです。特定のコンポーネントのカスタマイズ要件については、それらのコンポーネントに関する章を参照してください。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、セキュリティー管理者および TCP/IP 管理者の支援が必要になります。このタスクには、以下のリソースと特殊なカスタマイズ・タスクが必要です。

- LPA データ・セット
- APF 許可データ・セット
- 各種の PARMLIB 更新
- 各種のセキュリティー・ソフトウェア更新
- 内部およびクライアント/ホスト通信用の各種 TCP/IP ポート
- (オプション) SVC をアクティブ化する IPL

インストールを検証し、ご使用のサイトで Developer for z Systems の使用を開始するためには、以下のタスクを実行してください。特に断りがない限り、すべてのタスクは必須です。

1. カスタマイズ可能なサンプルのコピーを作成し、Developer for z Systems 用の作業環境を作成します。詳細については、『カスタマイズのセットアップ』を参照してください。
2. z/OS UNIX システム限度を更新し、製品登録を更新し、開始タスクを開始します。さらに、APF 許可データ・セットと LINKLIST データ・セットを定義し、オプションで SVC および LPA データ・セットを定義します。詳細については、19 ページの『PARMLIB の変更』を参照してください。
3. 開始タスク・プロシーチャーを作成し、プロシーチャーをコンパイルしてリンクします。詳細については、26 ページの『PROCLIB の変更』を参照してください。
4. セキュリティー定義を更新します。詳細については、29 ページの『セキュリティー定義』を参照してください。
5. Developer for z Systems 構成ファイルをカスタマイズします。詳細については、以下を参照してください。
 - 30 ページの『rdz.env、環境構成ファイル』

カスタマイズのセットアップ

Developer for z Systems には、いくつかのサンプル構成ファイルとサンプル JCL が含まれています。カスタマイズした内容がメンテナンスの適用時に上書きされないように、それらのメンバーと z/OS UNIX ファイルをすべて別のロケーションにコピーし、そのコピーをカスタマイズしてください。

Developer for z Systems の一部の機能は、z/OS UNIX 内に特定のディレクトリーが存在することを必要とします。それらのディレクトリーを製品のカスタマイズ時に作成する必要があります。インストールの労力を軽減するために、コピーと必要なディレクトリーを作成するサンプル・ジョブ、FELSETUP が提供されています。

構成ファイルおよび構成 JCL のカスタマイズ可能なコピーを作成し、必要な z/OS UNIX ディレクトリーを作成するには、FEL.SFELSAMP データ・セット内のサンプル・メンバー FELSETUP をカスタマイズして実行依頼します。必要なカスタマイズ・ステップは、このメンバー内に記述されています。

このジョブは、以下のタスクを実行します。

- FEL.#CUST.PARMLIB を作成し、これにサンプル構成ファイルを取り込みます。
- FEL.#CUST.PROCLIB を作成し、これにサンプル SYS1.PROCLIB メンバーを取り込みます。
- FEL.#CUST.JCL を作成し、これにサンプル構成 JCL を取り込みます。
- FEL.#CUST.CNTL を作成し、これにサンプルのサーバー始動スクリプトを取り込みます。
- FEL.#CUST.ASM を作成し、これにサンプル・アセンブラー・ソース・コードを取り込みます。
- FEL.#CUST.SQL を作成し、これにサンプル SQL コマンド・ファイルを取り込みます。
- z/OS Explorer 構成ディレクトリー /etc/zexpl/* の中にサンプル構成ファイルを取り込みます。
- /var/rdz/* を Developer for z Systems の各種機能の作業ディレクトリーとして作成し、これにサンプル・ファイルを取り込みます。

注:

- この資料の構成ステップでは、特に断りがない限り、FELSETUP ジョブによって作成されたメンバーおよびファイル・ロケーションを使用します。更新不可のオリジナルのサンプルは、FEL.SFELSAMP および /usr/lpp/ibm/rdz/samples/ に入っています。
- どのサンプル・メンバーがどのデータ・セットにコピーされるか、どのディレクトリーが作成されるか、それらの許可ビット・マスク、および各種サンプル・ファイルがどこにコピーされるかについての詳細は、FEL.SFELSAMP(FELSETUP) 内のコメントを参照してください。
- FEL.SFELSAMP(FELSETUP) 内のコメントには、既存のセットアップをマイグレーションする際に役立つように、Rational Developer for z Systems の異なるバージョン間の変更点についても記載されています。
- すべての z/OS Explorer および Developer for z Systemsz/OS UNIX ファイルを同じファイル・システム (HFS または zFS) に保持するが、構成ファイルを /etc/zexpl に配置したい場合には、シンボリック・リンクを使用して、この問題を解決できます。以下のサンプル z/OS UNIX コマンドは、既存のファイル・システム (/usr/lpp/ibm/rdz/cust) 内に新規ディレクトリーを作成し、それへのシンボリック・リンク (/etc/zexpl) を定義します。

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

PARMLIB の変更

このセクションでは、以下の PARMLIB の変更点について説明します。

- 『BPXPRMxx での z/OS UNIX 限度の設定』
- 20 ページの『COMMNDxx への開始タスクの追加』
- 20 ページの『IEASVCxx での SVC 定義』
- 21 ページの『IFAPRDxx での製品イネーブルメント』
- 21 ページの『LPALSTxx での LPA 定義』
- 22 ページの『PROGxx での APF 許可』
- 23 ページの『PROGxx での LINKLIST 定義』
- 24 ページの『必要な LINKLIST 定義と LPA 定義』
- 25 ページの『他の製品用の LINKLIST 定義』

次のセクションにリストされている PARMLIB 定義について詳しくは、「MVS 初期設定およびチューニング解説書」(SA88-8564)を参照してください。サンプルのコンソール・コマンドについて詳しくは、「MVS システム・コマンド」(SA88-8593)を参照してください。

BPXPRMxx での z/OS UNIX 限度の設定

SCLM Developer Toolkit は、いくつかのユーザー・アクティビティー中にユーザー所有の複数のプロセスを同時に開始します。

IPL 時にどの BPXPRMxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで OMVS=xx を定義してください。

MAXPROCUSER は、単一の z/OS UNIX ユーザー ID が同時にアクティブにしておくことができるプロセスの最大数を指定します。SYS1.PARMLIB(BPXPRMxx) で MAXPROCUSER を 50 以上に設定してください。この設定は、Developer for z Systems を使用する各クライアントに対してアクティブにする必要があるため、システム全体の限度となります。

これらの値は、以下のコンソール・コマンドで検査し、動的に (次回の IPL まで) 設定できます。

- DISPLAY OMVS,0
- SETOMVS MAXPROCUSER=50

注:

- ここで提案した MAXPROCUSER 値は、固有の z/OS UNIX ユーザー ID (UID) を持つユーザーに基づいています。ユーザーが同じ UID を共用する場合は、この値を大きくしてください。
- Developer for z Systems を SMP/E でインストールするとき、別個のファイル・システム (HFS の zFS) にコードを配置し、そのファイル・システムがシステム IPL 時にマウントされるように BPXPRMxx を更新することが勧められていました。この更新を依然として必要としている場合のために、マウント・コマンドのサンプルを以下に再掲載します。

```

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 */

```

- Developer for z Systems Host Utilities を SMP/E でインストールするとき、個別のファイル・システム (HFS の zFS) にコードを配置し、そのファイル・システムがシステム IPL 時にマウントされるように BPXPRMxx を更新することが推奨されていました。この更新を依然として必要としている場合のために、マウント・コマンドのサンプルを以下に再掲載します。

```

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 */

```

COMMNDxx への開始タスクの追加

Developer for z Systems サーバーの始動コマンドを SYS1.PARMLIB(COMMANDxx) に追加し、これらのサーバーが次のシステム IPL で自動的に始動されるようにします。IPL 時にどの COMMNDxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで CMD=xx を定義してください。

オプションの統合デバッガーが機能するには、システム上で Developer for z Systems DBGMR サーバーがアクティブ状態である必要があります。

サーバーを定義および構成した後、これらのサーバーを以下のコンソール・コマンドで動的に (次の IPL まで) 始動できます。

- S DBGMR

注: 他のすべてのサーバーより前に、またはそれらと一緒に、サーバーには特定の始動順序はありません。唯一の要件は、最初のユーザーが接続を試みるまでにサーバーが稼働していることです。

IEASVCxx での SVC 定義

オプションの統合デバッガーは、システムに Developer for z Systems 監視プログラム呼び出し (SVC) が定義されていることを必要とします。

インストール先定義 SVC は SYS1.PARMLIB(IEASVCxx) に定義され、IPL がアクティブ化されている必要があります。関連するロード・モジュールが IPL 時に LPA にロードされる必要があります。ただし、統合デバッガーを使用すれば、デバッグ・マネージャー開始タスクによって動的インストールまたは更新を行うことができます。IPL 時にどの IEASVCxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで SVC=xx を定義してください。

IEASVCxx で以下を指定して Developer for z Systems SVC を定義します。

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

SVC 番号 251 がデフォルトですが、z/OS で決められている 200 から 255 までの範囲内であれば、任意の値を使用できます。統合デバッガーは、使用される SVC 番号を検出します。

注: バージョン 9.1.1 より前のバージョンの Developer for z Systems では、現在使用されていない別の SVC、AQESVC01 を使用していました。これは、ご使用のホスト・システムにバージョン 9.1.1 以上の Developer for z Systems しかない場合には、削除できます。

IFAPRDxx での製品イネーブルメント

Developer for z Systems を、製品コード 5697-CDT の IBM Enterprise COBOL Suite for z/OS V1.1 (以降) の一部として購入し、製品コード 5697-CDT の他のプログラムについてはまだ購入していない場合は、項目を IFAPRDxx parmlib メンバーに含めて関連プログラムを使用可能にしてください。IPL 時にどの IFAPRDxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで PROD=xx を定義してください。

IFAPRDxx に以下を指定して、Enterprise COBOL Suite for z/OS (製品コード 5697-CDT) を定義します:

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

あるいはオプションで、Developer for z Systems を (Enterprise COBOL Suite for z/OS の一部ではなく) 別箇に購入した場合、スタンドアロン製品コード 5724-T07 を使用して Developer for z Systems の IFAPRDxx parmlib メンバーに項目を含めることもできます。

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

IFAPRDxx parmlib メンバーが更新されたら、以下のコンソール・コマンドを使用して (次回の IPL まで) 動的にアクティブ化できます。

```
SET PROD=xx
```

注: Developer for z Systems によって、以下のフィーチャーが登録されます。

- RDZ-RSED (Developer for z Systems 用)
- RDZ-CC (Developer for z Systems Host Utilities のコード・カバレッジ用)
- RDZ-CR (Developer for z Systems Host Utilities のコード・レビュー)

LPALSTxx での LPA 定義

オプションの共通アクセス・リポジトリ・マネージャー (CARMA) サービスは、CARMA サーバーに対するさまざまなサーバー始動方式をサポートしています。CRASTART 開始メソッドでは、FEL.SFELLPA ロード・ライブラリー内のモジュールがリンク・パック域 (LPA) にあることが必要です。

LPA データ・セットは、SYS1.PARMLIB(LPALSTxx) で定義されます。IPL 時にどの LPALSTxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで LPA=xx を定義してください。

以下のコンソール・コマンドで、LPA 定義を動的に (次回の IPL まで) 設定できます。

```
• SETPROG LPA,ADD,DSN=FEL.SFELLPA,MASK=*
```

注:

- LPALSTxx にリストされているデータ・セットは、マスター・カタログか、LPALSTxx メンバーで識別されるユーザー・カタログにカタログされる必要があります。
- 新規データ・セットを LPALSTxx に追加するには、CLPA (LPA の作成) を指定した IPL がアクティブ化されている必要があります。
- LPA にロードされるすべてのライブラリーは、自動的に、APF 許可が与えられていて、プログラム制御されるものとみなされます。これらのライブラリー用に、適切なセキュリティー管理を適切な場所に配置します。
- LPA 配置用に設計されたライブラリーを LPA に配置しないことを選択し、代わりに LINKLIST または STEPLIB を使用する場合、APF 許可およびプログラム制御状況を定義してください。

PROGxx での APF 許可

オプションのデバッグ・マネージャーを機能させるためには、FEL.SFELAUTH および FEL.SFELLPA ロード・ライブラリー内のモジュールに APF 許可が必要です。FEL.SFELLPA はデバッグ・マネージャーの STEPLIB になければならないため、明示的な APF 許可が必要です。

オプションの SCLM Developer Toolkit サービスが機能するためには、REXX ランタイム・ライブラリー (REXX.*.SEAGLPA) に APF 許可が必要です。

APF 許可は、デフォルトでは SYS1.PARMLIB(PROGxx) で定義されます。IPL 時にどの PROGxx parmlib メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで PROG=xx を定義してください。

APF 許可は、以下のコンソール・コマンドで動的に (次回の IPL まで) 設定できます。ここで、volser はデータ・セットが存在するボリューム (SMS の管理対象でない場合) です。

```
• SETPROG APF,ADD,DSN=FEL.SFELAUTH,SMS
• SETPROG APF,ADD,DSN=FEL.SFELLPA,SMS
• SETPROG APF,ADD,DSN=REXX.V1R4M0.SEAGLPA,VOL=volser
```

注:

- REXX 製品パッケージに代替ライブラリーを使用している場合、デフォルトの REXX ランタイム・ライブラリー名は REXX.*.SEAGALT であり、前のサンプルで使用した REXX.*.SEAGLPA ではありません。
- REXX.*.SEAGLPA などの LPA ライブラリーは、LPA 内に置かれている場合は自動的に APF 許可があり、したがって明示的な定義は必要ありません。
- また、前提条件または相互前提条件となる一部の製品 (z/OS Explorer など) にも APF 許可が必要です。詳細については、関連製品のカスタマイズ・ガイドを参照してください。

PROGxx での LINKLIST 定義

Developer for z Systems の LINKLIST 定義は、以下の 3 つのカテゴリーに分けることができます。

- Developer for z Systems 機能に必要な Developer for z Systems ロード・ライブラリー。これらの定義については、このセクションで説明します。
- Developer for z Systems 機能に必要な必須ロード・ライブラリー。これらの定義については、24 ページの『必要な LINKLIST 定義と LPA 定義』で説明します。
- 他の製品が必要とする Developer for z Systems ロード・ライブラリー。これらの定義については、25 ページの『他の製品用の LINKLIST 定義』で説明します。

表 7. ロード・モジュールと機能のマッチング

ロード・ライブラリー	ロード・モジュール	使用法	STEPLIB
FEL.SFELAUTH	AQE* および CEE*	83 ページの『第 5 章 統合デバッガー』	ELAXFGO プロシージャー、または CICS
FEL.SFELLMOD	IRZ* および IIRZ*	103 ページの『エンタープライズ・サービス・ツールの診断用 IRZ メッセージ』	CICS、IMS、または MVS のパッチ
FEL.SFELLOAD	AZU* および IAZU*	100 ページの『Enterprise COBOL および PL/I での xUnit サポート』	rdz.env または MVS パッチ
	BWB*	75 ページの『第 4 章 SCLM Developer Toolkit』	rdz.env
	CRA*	35 ページの『第 3 章 共通アクセス・リポジトリ・マネージャー (CARMA)』	CRASUB* または crastart*.conf
	ELAX*	27 ページの『ELAXF* リモート・ビルド・プロシージャー』 (エラーのフィードバックおよびインクルード・プリプロセッサ)	ELAXF* プロシージャー
	FEJB*	102 ページの『CICS 双方向言語サポート』	CICS
FEL.SFELLPA	CRA*	35 ページの『第 3 章 共通アクセス・リポジトリ・マネージャー (CARMA)』	CRASRV.properties
	AQE*	83 ページの『第 5 章 統合デバッガー』	適用不可 (LPA は必須)

リストされている Developer for z Systems サービスが機能するためには、STEPLIB または LINKLST を通じて、表 7 に示されているサービスに関連するすべてのモジュールを使用可能にする必要があります。SFELLMOD ライブラリーは、Developer for z Systems 自体によって使用されるのではなく、Developer for z Systemsによって生

成されたコードによって使用されることに注意してください。STEPLIB を使用する
場合、どこで STEPLIB (または CICS の場合は DFHRPL) を定義する必要があるの
かについては、23 ページの表 7 の STEPLIB の欄を参照してください。ただし、以
下のことに注意する必要があります。

- STEPLIB を z/OS UNIX で使用すると、パフォーマンスに悪い影響が出ます。
- 1 つの STEPLIB ライブラリーに APF 許可がある場合、すべての STEPLIB ラ
イブラリーに APF 許可があることが必要です。ライブラリーは、STEPLIB 内で
許可のないライブラリーと混用した場合、APF 許可を失います。
- JCL で STEPLIB DD に追加されたライブラリーは、その JCL によって開始さ
れた z/OS UNIX プロセスに伝搬されません。

使用しているサイトが IBM の推奨に従っている場合、LINKLIST データ・セット
は、SYS1.PARMLIB(PROGxx) の中で定義されています。IPL 時にどの PROGxx parmlib
メンバーが使用されるかを指定するには、IEASYSxx parmlib メンバーで PROG=xx を
定義してください。

必要な定義は以下のようになります。ここで、listname はアクティブにされる
LINKLIST セットの名前で、volser はマスター・カタログにカタログされていない
データ・セットが存在しているボリュームです。

- LNKLIST ADD NAME(listname) DSNNAME(FEL.SFELAUTH) VOLUME(volser)
- LNKLIST ADD NAME(listname) DSNNAME(FEL.SFELLOAD)

以下のコンソール・コマンド・グループを使用して、動的に (次回の IPL までの)
LINKLIST 定義を作成できます。ここで volser は、マスター・カタログ内にカタ
ログされていないデータ・セットが存在するボリュームです。

1. LNKLIST DEFINE,NAME=LLTMP,COPYFROM=CURRENT
2. LNKLIST ADD NAME=LLTMP,DSN=FEL.SFELAUTH,VOL=volser
3. LNKLIST ADD NAME=LLTMP,DSN=FEL.SFELLOAD
4. LNKLIST ACTIVATE,NAME=LLTMP

必要な LINKLIST 定義と LPA 定義

オプションのサービスを使用できるようにするには、以下の追加ライブラリーを
STEPLIB または LINKLIST/LPALIB を通じて使用可能にする必要があります。この
リストには、Developer for z Systems と対話する製品 (z/OS Explorer など) に固有
のデータ・セットは含まれていません。

- システム・ロード・ライブラリー
 - SYS1.LINKLIB
- 言語環境プログラム・ランタイム
 - CEE.SCEERUN
 - CEE.SCEERUN2
- C++ の DLL クラス・ライブラリー
 - CBC.SCLBDLL
- ISPF の TSO/ISPF クライアント・ゲートウェイ (SCLMDT 用)
 - ISP.SISPLoad

- ISP.SISPLPA
- REXX ランタイム・ライブラリー (SCLMDT 用)
 - REXX.*.SEAGLPA
- システム・ロード・ライブラリー (統合デバッガー用)
 - SYS1.MIGLIB
- システム・ロード・ライブラリー (z/OS 1.13 以上の統合デバッガー用)
 - SYS1.SIEAMIGE
- システム・ロード・ライブラリー (Enterprise COBOL および PL/I の単体テスト用)
 - SYS1.CSSLIB
 - SYS1.SIXML0D1

注:

- REXX 製品パッケージに代替ライブラリーを使用している場合、デフォルトの REXX ランタイム・ライブラリー名は REXX.*.SEAGALT であり、前のサンプルで使用した REXX.*.SEAGLPA ではありません。
- LPA にロードされるすべてのライブラリーは、自動的に、APF 許可が与えられていて、プログラム制御されるものとみなされます。これらのライブラリー用に、適切なセキュリティー管理を適切な場所に配置します。
- LPA 配置用に設計されたライブラリー (REXX.*.SEAGLPA など) は、LINKLIST または STEPLIB によってアクセスされる場合、追加のプログラム制御か APF 許可を必要とすることがあります。
- また、前提条件または相互前提条件となる一部の製品 (z/OS Explorer など) にも STEPLIB または LINKLIST/LPALIB 定義が必要です。詳細については、関連製品のカスタマイズ・ガイドを参照してください。

デフォルトでは、LINKLIST データ・セットは SYS1.PARMLIB(PROGxx) に定義します。LPA データ・セットは、SYS1.PARMLIB(LPALSTxx) で定義されます。

STEPLIB を使用する場合は、LINKLIST/LPALIB によって使用できないライブラリーを、rdz.env (RSE 構成ファイル) の STEPLIB ディレクティブで定義する必要があります。ただし、以下の点に注意してください。

- STEPLIB を z/OS UNIX で使用すると、パフォーマンスに悪い影響が出ます。
- 1 つの STEPLIB ライブラリーに APF 許可を与える場合、他のすべての STEPLIB ライブラリーにも APF 許可を与える必要があります。ライブラリーは、STEPLIB 内で許可のないライブラリーと混用した場合、APF 許可を失います。
- JCL で STEPLIB DD に追加されたライブラリーは、その JCL によって開始された z/OS UNIX プロセスに伝搬されません。

他の製品用の LINKLIST 定義

Developer for z Systems クライアントには、エンタープライズ・サービス・ツールと呼ばれるコード生成コンポーネントがあります。生成コードが診断用エラー・メ

ッセージを発行するためには、FEL.SFELMOD ロード・ライブラリー内のすべての IRZM* モジュールと IIRZ* モジュールを、STEPLIB または LINKLIST を通じて使用できるようにする必要があります。

デフォルトでは、LINKLIST データ・セットは SYS1.PARMLIB(PROGxx) に定義します。

STEPLIB を使用する場合は、LINKLIST によって使用できないライブラリーを、コード (IMS またはバッチ・ジョブ) を実行するタスクの STEPLIB ディレクティブで定義する必要があります。ただし、1 つの STEPLIB ライブラリーに APF 許可を与える場合、他のすべての STEPLIB ライブラリーにも許可を与える必要があります。ライブラリーは、STEPLIB 内で許可のないライブラリーと混用した場合、APF 許可を失います。

PROCLIB の変更

このセクションでは、以下の PROCLIB の変更点について説明します。

- 『DBGMGR、デバッグ・マネージャー開始タスク』
- 27 ページの『ELAXF* リモート・ビルド・プロシージャー』

次のセクションで示す開始タスク・プロシージャーおよびリモート・ビルド・プロシージャーは、使用する JES サブシステムに対して定義されたシステム・プロシージャー・ライブラリー内に存在する必要があります。次のセクションの説明では、IBM のデフォルトのプロシージャー・ライブラリー SYS1.PROCLIB を使用しています。

DBGMGR、デバッグ・マネージャー開始タスク

サンプル開始タスク・メンバー FEL.#CUST.PROCLIB(DBGMGR) 内の説明に従ってこのメンバーをカスタマイズし、それを SYS1.PROCLIB にコピーします。下記のコード・サンプルに示すように、以下の情報を提供します。

- タイムゾーン・オフセット (デフォルトは EST5DST)
- 外部 (クライアント/ホスト) 通信に使用されるポート番号 (デフォルトは 5335)
- 内部 (ホスト限定) 通信に使用されるポート (デフォルトは 5336)
- 統合デバッガーに使用される SVC 番号 (デフォルトは 251)
- ロード・ライブラリーの高位修飾子 (デフォルトは FEL)

```

/*
/* RDz Debug Manager
/*
/*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
/*

```

図 1. DBGMGR: デバッグ・マネージャー開始タスク

注:

- これはオプションの開始タスクです。これは Developer for z Systems の統合デバッガー機能によって使用されます。詳しくは、83 ページの『第 5 章 統合デバッガー』を参照してください。
- サンプル JCL の最初の名前は FEL.SFELSAMP(AQJCL) ですが、17 ページの『カスタマイズのセットアップ』で FEL.#CUST.PROCLIB(DBGMGR) に名前変更されます。
- このタスクにおける推奨ワークロード・マネージャー (WLM) 目標については、「ホスト構成リファレンス (SC43-2912)」の『WLM に関する考慮事項』を参照してください。
- 統合デバッガー SVC がすでにロードされている場合、ここで指定される SVC 番号は無視され、アクティブな SVC 番号が使用されます。

ELAXF* リモート・ビルド・プロシージャー

Developer for z Systems は、CICS BMS マップ、IMS MFS 画面、および COBOL、PL/I、アセンブラー、C/C++ の各プログラムの、JCL 生成、リモート・プロジェクト・ビルド、およびリモート構文検査の各フィーチャーに使用できるサンプル JCL プロシージャーを提供します。これらのプロシージャーを使用すると、インストールごとに独自の標準を適用でき、開発者は、同じプロシージャーを同じコンパイラー・オプションおよびコンパイラー・レベルで使用できます。

サンプル・プロシージャーとその機能を、表 8 に示します。

表 8. サンプル ELAXF* プロシージャー

メンバー	目的
ELAXFADT	高水準アセンブラー・プログラムのアセンブルとデバッグのためのサンプル・プロシージャー。
ELAXFASM	高水準アセンブラー・プログラムのアセンブルのためのサンプル・プロシージャー。
ELAXFBMS	CICS BMS オブジェクトおよびそれに対応する copy、dsect、または include メンバーを作成するためのサンプル・プロシージャー。

表 8. サンプル ELAXF* プロシージャ (続き)

メンバー	目的
ELAXFCOC	COBOL コンパイル、統合 CICS 変換、および統合 DB2 変換を行うためのサンプル・プロシージャ。
ELAXFCOP	COBOL プログラムに埋め込まれた EXEC SQL ステートメントの DB2 プリプロセスのためのサンプル・プロシージャ。
ELAXFCOT	COBOL プログラムに組み込まれている EXEC CICS ステートメントの CICS 変換するためのサンプル・プロシージャ。
ELAXFCPC	C コンパイルのためのサンプル・プロシージャ。
ELAXFCPP	C++ コンパイルのためのサンプル・プロシージャ。
ELAXFCPI	SCM プリプロセッサ・ステートメント (-INC および ++INCLUDE) を使用した COBOL コンパイルのためのサンプル・プロシージャ。
ELAXFDCL	プログラムを TSO モードで実行するためのサンプル・プロシージャ。
ELAXFGO	GO ステップのためのサンプル・プロシージャ。
ELAXFLNK	C/C++、COBOL、PLI、および高水準アセンブラの各プログラムをリンクするためのサンプル・プロシージャ。
ELAXFMFS	IMS MFS 画面を作成するためのサンプル・プロシージャ。
ELAXFPLP	PLI プログラムに埋め込まれた EXEC SQL ステートメントの DB2 プリプロセスのためのサンプル・プロシージャ。
ELAXFPLT	PLI プログラムに埋め込まれた EXEC CICS ステートメントの CICS 変換を行うためのサンプル・プロシージャ。
ELAXFPLI	PL/I コンパイル、統合 CICS 変換、および統合 DB2 変換を行うためのサンプル・プロシージャ。
ELAXFPP1	SCM プリプロセッサ・ステートメント (-INC および ++INCLUDE) を含む PL/I コンパイルのためのサンプル・プロシージャ。
ELAXFSP	ストアード・プロシージャを DB2 に登録するためのサンプル・プロシージャ。
ELAXFSQL	SQL を呼び出すためのサンプル・プロシージャ。
ELAXFTSO	生成された DB2 コードを TSO モードで実行およびデバッグするためのサンプル・プロシージャ。
ELAXFUOP	CICS または IMS サブシステムで実行するプログラムをビルドするときに、UOPT ステップを生成するためのサンプル・プロシージャ。

プロシージャの名前とプロシージャ内のステップの名前は、Developer for z Systems クライアントに付属しているデフォルト・プロパティに一致しています。プロシージャの名前またはプロシージャ内のステップの名前を変更する場合は、すべてのクライアント上の対応するプロパティ・ファイルを更新する必要があります。プロシージャ名とステップ名は変更しないでください。

サンプルのビルド・プロシージャ・メンバー FEL.#CUST.PROCLIB(ELAXF*) を、各メンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーします。各種製品ライブラリーに、29 ページの表 9 に示した正しい高位修飾子を指定します。

表 9. ELAXF* 高位修飾子チェックリスト

製品	デフォルト HLQ	値
Developer for z Systems	FEL	
CICS	CICSTS52.CICS	
DB2	DSNA11	
IMS	IMS	
COBOL	IGY.V5R2M0	
PL/I	PLI.V4R2M0	
C/C++	CBC	
LE	CEE	
システム LINKLIB	SYS1	
システム MACLIB	SYS1	

一部の ELAXF* プロシージャは、固定された低位修飾子を持たないデータ・セット名を参照します。その例の 1 つは DB2 実行ライブラリーです。これは、DB2 管理者によってコンパイルされた DB2 ユーティリティを保持します。表 10 を使用して、デフォルトのデータ・セット名をユーザーのサイトで使用されている名前にマップします。

表 10. ELAXF*：完全修飾データ・セット・チェックリスト

製品	デフォルト DSN	値
Developer for z Systems - SQL サンプル	FEL.#CUST.SQL	
DB2 実行ライブラリー	DSNA11.RUNLIB.LOAD	

ELAXF* プロシージャをシステム・プロシージャ・ライブラリーにコピーできない場合、JCLLIB カード (JOB カードの直後) をクライアントのジョブ・プロパティに追加するように Developer for z Systems ユーザーに依頼します。

```
//MYJOB    JOB <job parameters>
//PROCS    JCLLIB ORDER=(FEL.#CUST.PROCLIB)
```

セキュリティ定義

Developer for z Systems のセキュリティ定義を作成するには、サンプルの FELRACF および AQERACF メンバーをカスタマイズして実行依頼します。このジョブを実行依頼するユーザーは、RACF SPECIAL などのセキュリティ管理者特権を持っている必要があります。

FELRACF および AQERACF は FEL.#CUST.JCL に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

以下にリストする Developer for z Systems のセキュリティ関連定義については、109 ページの『第 9 章 セキュリティ定義』で詳しく説明します。

- セキュリティの設定およびクラスをアクティブにする

- セキュアな z/OS UNIX サーバーとしてデバッグ・マネージャーを定義する
- 統合デバッガーへのアクセスの定義
- データ・セット・プロファイルを定義する

rdz.env、環境構成ファイル

z/OS Explorer RSE サーバー・プロセス (RSE デーモン、RSE スレッド・プール、および RSE サーバー) は、Developer for z Systems 環境変数を認識するために `rdz.env` 内の定義を使用します。

`rdz.env` は `/etc/zexpl/` に置かれます。ただし、`FEL.SFELSAMP(FELSETUP)` ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

以下のサンプル `rdz.env` ファイルを参照してください (実際のシステム環境に合わせてカスタマイズできます)。明示的に指定されないすべての変数のために、デフォルト値が用意されています。ファイルの構文は標準的な z/OS UNIX シェル構文規則に従っています。例えば、US コード・ページを使用する場合、コメントは番号記号 (#) で始まり、等号 (=) の前後のスペースはサポートされません。

注: 変更を有効にするには、z/OS Explorer RSED 開始タスクを再始動する必要があります。

```
#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
```

図 2. `rdz.env`: 環境構成ファイル

以下の定義はオプションです。省略した場合は、デフォルト値が使用されます。

RDZ_HOME

Developer for z Systems ホーム・ディレクトリー。デフォルトは /usr/lpp/ibm/rdz です。コメントを外し、実際の Developer for z Systems インストール済み環境に合わせて変更してください。

RDZ_HLQ

Developer for z Systems をインストールするために使用する高位修飾子。デフォルトは FEK です。コメントを外し、Developer for z Systems データ・セットの場所に合わせて変更してください。

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

デバッグ・マネージャーに自動再接続します。デフォルトは 1 です。これは、デバッグ・マネージャー・サーバーとの接続が確立されなかったか、切断されたときに、デバッグ・マイナーがデバッグ・マネージャーへの再接続を 1 回試みることを意味します。コメントを外して、異なる値を指定して、デバッグ・マイナーがデバッグ・マネージャーに接続しようとする頻度を制限します。

表 11. デバッグ・マネージャーへの自動再接続

debug.miner.autoreconnect	再接続動作
-1	再接続しない
0 (デフォルト)	成功するまで毎分再接続しようとする
1-86400	指定された回数まで再接続しようとする。最大値は 86400 であり、24 時間です。

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

localhost の TCP/IP 定義の代替。デバッグ・マイナーは、localhost の指定を使用してデバッグ・マネージャーに接続しようとします。localhost がローカル・ループバック・アドレス (IPv4 の場合 127.0.0.1、IPv6 の場合 ::1) に解決しない場合、これは失敗します。必要な場合は、コメントを外して、ローカル・ループバック・アドレスを指定します。

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

この変数は、C コードでのコンテンツ・アシスト用の強制インクルードのリストが入っている完全修飾された z/OS UNIX ファイルを指します。強制インクルードは、ファイルまたはメンバーがプリプロセッサ・ディレクティブを使用してソース・コードに組み込まれたかどうかに関係なく、コンテンツ・アシスト操作の実行時に構文解析されるファイルまたはディレクトリー、データ・セット、またはデータ・セット・メンバーで構成されます。構成ファイルの名前を指定するには、コメントを外してカスタマイズします。

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

この変数は、C++ コードでのコンテンツ・アシスト用の強制インクルードのリストが入っている完全修飾された z/OS UNIX ファイルを指します。強制インクルードは、ファイルまたはメンバーがプリプロセッサ・ディレクティブを使用してソース・コードに組み込まれたかどうかに関係なく、コンテンツ・アシスト操作の実行時に構文解析されるファイルまたはディレクト

リー、データ・セット、またはデータ・セット・メンバーで構成されます。構成ファイルの名前を指定するには、コメントを外してカスタマイズします。

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DCPP_CLEANUP_INTERVAL=60000"

未使用の C/C++ ヘッダー・ファイルのクリーンアップ間隔 (ミリ秒単位)。デフォルトは 60000、つまり 1 分です。クリーンアップ間隔を変更するには、コメントを外してカスタマイズします。値 0 を指定すると、C/C++ ヘッダー・ファイルのキャッシュが行われないため、エディターでのリモート・コンテンツ・アシストのパフォーマンスが低下します。

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS -DRIS_BUFFER=8"

リモート索引の作成時に使用されるバッファー・サイズ (M バイト単位)。デフォルトは 8 MB です。バッファー・サイズを変更するには、コメントを外してカスタマイズします。有効な値は 1 から 2000 まで (両端の値を含む) の整数です。バッファーを大きくすると索引作成時間が短縮されますが、スレッド・プールの Java ヒープがより多く使用されます。索引の作成が終了する前にバッファーがいっぱいになると、バッファーは索引に自動的にフラッシュされます。

_RSE_JAVAOPTS="\$_RSE_JAVAOPTS

-DDISABLE_REMOTE_INDEX_SEARCH=true"

クライアントのリモート索引検索メニュー項目を使用不可にします。デフォルトは false です。ユーザーがホスト・システム・データ・セットのリモート索引を作成できないようにするには、コメントを外して true と指定します。

#_RSE_JAVAOPTS="\$_RSE_JAVAOPTS

-DDISABLE_DELETE_IN_SUBPROJECT=true"

z/OS サブプロジェクトのコンテキスト・メニューで「削除」メニュー項目を使用不可にします。デフォルトは false です。ユーザーが z/OS サブプロジェクトのコンテキスト・メニューで「削除」メニュー項目を使用できないようにするには、コメントを外して true と指定します。

STEPLIB

LINKLIST/LPALIB でなく、MVS データ・セットにアクセスします。デフォルトは NONE です。

以下の 1 つ以上の STEPLIB ディレクティブのコメントを外してカスタマイズすることにより、前提条件のライブラリーを LINKLIST/LPALIB 内に保持せずに済むことができます。以下にリストしたライブラリーの詳しい使用方法については、19 ページの『PARMLIB の変更』を参照してください。

```
# SCLM Developer Toolkit
STEPLIB=$STEPLIB:$RSE_HLQ.SFELAUTH:$RSE_HLQ.SFELLOAD
# zUnit, xUnit support for Enterprise COBOL and PL/I
STEPLIB=$STEPLIB:$RSE_HLQ.SFELLOAD:SYS1.CSSLIB:SYS1.SIXML01
```

注:

- STEPLIB を z/OS UNIX で使用すると、パフォーマンスに悪い影響が出ます。

- 1 つの STEPLIB ライブラリーに APF 許可を与える場合、他のすべての STEPLIB ライブラリーにも APF 許可を与える必要があります。ライブラリーは、STEPLIB 内で許可のないライブラリーと混用した場合、APF 許可を失います。
- LPA 配置用に設計されたライブラリーは、LINKLIST または STEPLIB によってアクセスされる場合に、追加のプログラム制御および APF 許可を必要とすることがあります。
- サーバー JCL 内の STEPLIB DD ステートメントのコーディングでは、要求された STEPLIB 連結は設定されません。

RSE_UBLD_DD

TSO または ISPF コマンドを呼び出す Developer for z Systems クライアントから IBM Rational Team Concert™ ユーザー・ビルド用 JCL を生成する際に使われる DD ステートメントを指定します。デフォルトで、Developer for z Systems は ISPF.conf (rdz.env 内の CGI_ISPCONF によって参照される) の定義を使用します。指定されているファイルでコメントを外し、DD 定義を使用するように変更します。『ISPF.conf、ISPF の TSO/ISPF クライアント・ゲートウェイ構成ファイル』で指定されている構文規則に従う必要があります。これは、オプションのディレクティブです。

RSE_UBLD_STEPLIB

TSO または ISPF コマンドを呼び出す Developer for z Systems クライアントから IBM Rational Team Concert ユーザー・ビルド用 JCL を生成する際に使われる STEPLIB ステートメントを指定します。デフォルトでは、Developer for z Systems は rdz.env 内の STEPLIB 定義を使用します。コメントを外して、指定されている STEPLIB 定義を使用するように変更します。これは、オプションのディレクティブです。

_SCLMDT_CONF_HOME

SCLM Developer Toolkit 基本構成ディレクトリー。デフォルトは /var/rdz/sclmdt です。SCLMDT が SCLM プロジェクト情報を保管するために使用する CONFIG ディレクトリーのロケーションに合わせて変更してください。このディレクティブは、SCLMDT を使用する場合にのみ使用されます。

注: SCLMDT_CNFG_HOME に指定したパスには、SCLMDT によって /CONFIG と /CONFIG/PROJECT が追加されます。手動では追加しないでください。

STEPLIB

STEPLIB については、以前に、必要な定義のセクションで説明しました。

_SCLMDT_TRANTABLE

ロング/ショート・ネーム変換 VSAM の名前。デフォルトは FEL.#CUST.LSTRANS.FILE です。コメントを外し、ISP.SISPSAMP (FLM02LST) SCLM サンプル・ジョブで使用される名前に合わせて変更してください。このディレクティブは、SCLM Developer Toolkit でロング/ショート・ネーム変換を使用する場合にのみ使用されます。

ANT_HOME

Ant インストールのホーム・ディレクトリー。デフォルトは /usr/lpp/Apache/Ant/apache-ant-1.7.1 です。使用する Ant インストール

に合わせて変更してください。このディレクティブは、SCLM Developer Toolkit で Java EE ビルド・サポートを使用する場合にのみ使用されます。

特定のコンポーネント

Developer for z Systems は、関係のないさまざまな機能で構成されており、機能ごとに独自のカスタマイズ・タスクがあります。求めるサービスを構成するには、該当するセクションの説明に従ってください。

Developer for z Systems スタンドアロン・コンポーネントへのカスタマイズ:

- 35 ページの『第 3 章 共通アクセス・リポジトリ・マネージャー (CARMA)』
- 75 ページの『第 4 章 SCLM Developer Toolkit』
- 83 ページの『第 5 章 統合デバッガー』
- 93 ページの『第 6 章 ホスト・ベースのコード分析』

Developer for z Systems 構成ファイルへのカスタマイズ

- 97 ページの『include.conf、C/C++ コンテンツ・アシスト用の強制インクルード』

他の製品への Developer for z Systems 関連のカスタマイズ:

- 98 ページの『z/OS UNIX サブプロジェクト』
- 99 ページの『インクルード・プリプロセッサのサポート』
- 100 ページの『Enterprise COBOL および PL/I での xUnit サポート』
- 101 ページの『エンタープライズ・サービス・ツール・サポート』
- 102 ページの『CICS 双方向言語サポート』
- 103 ページの『エンタープライズ・サービス・ツールの診断用 IRZ メッセージ』
- 104 ページの『FEKRNPLI、PL/I リモート・プリプロセッサ』

インストール検査

さまざまなインストール検査プログラム (IVP) についての詳細は、105 ページの『第 8 章 インストール検査』に記載されています。

第 3 章 共通アクセス・リポジトリ・マネージャー (CARMA)

共通アクセス・リポジトリ・マネージャー (CARMA) は、Repository Access Manager (RAM) 向けのサーバー・プラットフォームです。RAM は、z/OS ベースの Software Configuration Manager (SCM) 用のアプリケーション・プログラミング・インターフェース (API) です。SCM 機能を RAM に内包することにより、サポートされる SCM にクライアントがアクセスするときに、単一の API が使用できるようになります。

Developer for z Systems は事前に組み込まれた複数の RAM と、ユーザーが独自の RAM を作成するためのソース・コード・サンプルを提供しています。

ホスト・システム・ベースの SCM は、サービスにアクセスするために単一ユーザーのアドレス・スペースを必要とします。つまり、CARMA は、ユーザーごとに CARMA サーバーを 1 つ始動する必要があります。複数のユーザーをサポートする単一のサーバーを作成することはできません。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、セキュリティー管理者および TCP/IP 管理者の支援が必要になります。このタスクには、以下のリソースまたは特殊なカスタマイズ・タスクが必要です。

- (オプション) 内部通信用の TCP/IP ポート範囲
- (オプション) 開発者に CARMA VSAM ファイルの更新機能の使用を許可するセキュリティー規則
- (オプション) ユーザーに CRA* ジョブの実行依頼を許可するセキュリティー規則
- (オプション) LPA 更新

ご使用のサイトで CARMA の使用を開始するには、以下のタスクを実行してください。特に断りがない限り、すべてのタスクは必須です。

1. CARMA を始動する方式を選択し、どの RAM をアクティブにするかを選択します。事前構成のセットアップとして、RAM とサーバー始動方式のいくつかの組み合わせが可能です。詳細については、36 ページの『サーバー始動方式とアクティブ RAM の選択』を参照してください。
2. CARMA VSAM データ・セットを作成します。詳細については、59 ページの『CARMA VSAM データ・セット』と 60 ページの『CARMA Repository Access Manager (RAM)』を参照してください。
3. CARMA とインターフェースするための RSE 構成ファイルの初期カスタマイズ。完全なカスタマイズは、CARMA を始動するために選択した方式によって異なります。詳細については、49 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。
4. 選択した CARMA 始動方式と選択した RAM に応じて、関連する構成ファイルに必要なカスタマイズを行います。詳細については、以下を参照してください。

- 53 ページの『crastart*.conf、CRASTART サーバーの始動』
 - 56 ページの『CRASUB*、バッチ実行依頼サーバー始動』
5. オプションとして、CA Endeavor® SCM 固有の構成メンバーをカスタマイズします。詳細については、62 ページの『CRACFG、CRASCL、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』および 64 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。
 6. オプションで、データ・セット割り振り exec を更新します。詳細については、64 ページの『CRANDVRA、CA Endeavor® SCM RAM 初期割り振り exec』、68 ページの『CRAALLOC、カスタム RAM 割り振り exec』、および 70 ページの『(オプション) カスタム割り振り exec』を参照してください。
 7. オプションで、始動ユーザー出口を作成します。詳細については、(オプション) CARMA ユーザー出口を参照してください。
 8. オプションとして、IRXJCL に代わる CRAXJCL を作成します。詳細については、72 ページの『(オプション) IRXJCL と CRAXJCL』を参照してください。

注: この章で参照するサンプルのメンバーは FEL.#CUST.* および /etc/zexpl の中に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

サーバー始動方式とアクティブ RAM の選択

Developer for z Systems は CARMA サーバーを始動する複数の方式をサポートしています。Developer for z Systems も複数の Repository Access Manager (RAM) を備えています。これらは、実動 RAM とサンプル RAM という 2 つのグループに分けられます。ここでは、RAM とサーバー始動方式に考えられるいくつかの組み合わせを説明します。説明される構成シナリオは、それぞれ事前構成済みのセットアップとして使用できます。

CARMA サーバーの始動

Developer for z Systems は CARMA サーバーを始動する複数の方式をサポートしています。それぞれの方式には利点と欠点があります。

CRASTART

「CRASTART」方式は、CARMA サーバーを RSE 内のサブタスクとして始動します。この方式では、CARMA サーバーを始動するために必要なデータ・セット割り振りとプログラム呼び出しを別個の構成ファイルで定義し、その構成ファイルを使用するので、非常に柔軟なセットアップが可能です。この方式では最良のパフォーマンスが得られ、使用するリソースも最少で済みますが、CRASTART モジュールを LPA 内に配置する必要があります。

バッチ実行依頼

「バッチ実行依頼」方式は、ジョブを実行依頼することによって CARMA サーバーを始動します。これが、提供されたサンプル構成ファイルで使用されるデフォルトの方式です。この方式の利点は、ジョブ出力内で CARMA ログに簡単にアクセスできることです。また、開発者自身が保守する開発者ごとのカスタム・サーバー JCL

を使用できます。ただし、この方式では、CARMA サーバーを始動する開発者ごとに JES イニシエーターが 1 つ使用されます。

実動 RAM

実動タイプの RAM はすべての機能が使用できる事前組み込み型の RAM で、実稼働環境で SCM にアクセスするために使用できます。

CA Endeavor® SCM RAM

IBM Rational Developer for z Systems Interface for CA Endeavor® Software Configuration Manager は、Developer for z Systems クライアントが CA Endeavor® SCM に直接アクセスできるようにします。

CA Endeavor® SCM パッケージ RAM

CA Endeavor® SCM パッケージ RAM は、Developer for z Systems クライアントが CA Endeavor® SCM パッケージに直接アクセスできるようにします。

サンプル RAM

サンプルの RAM は CARMA 環境の構成をテストする目的で、また、独自の RAM を開発するための例として提供されています。ソース・コードが組み込まれています。

重要: 提供されているサンプルの RAM を実稼働環境で使用しないでください。

PDS RAM

PDS RAM は、「リモート・システム」ビューの「MVS ファイル」->「ユーザー・データ・セット」と同様のデータ・セット・リストを提供します。

スケルトン RAM

スケルトン RAM は、ユーザーが独自の RAM を開発する際の開始点として使用できる機能フレームワークを提供します。

SCLM RAM

SCLM RAM は、ISPF の Software Configuration Manager である SCLM への基本的な入り口となります。SCLM RAM は、デフォルトでは使用可能に設定されていません。

事前構成済みの RAM とサーバー始動の組み合わせ

事前構成のセットアップとして、RAM とサーバー始動方式のいくつかの組み合わせが可能です。ご使用の環境に適合させるため、リストされたシナリオで必要なカスタマイズは、マイナーなもののみです。

- 38 ページの『CA Endeavor® SCM RAM による CRAFTSTART』
- 41 ページの『サンプル RAM による CRAFTSTART』
- 43 ページの『CA Endeavor® SCM RAM によるパッチ実行依頼』
- 46 ページの『サンプル RAM によるパッチ実行依頼』

シナリオごとのさまざまなステップについて詳しくは、49 ページの『CARMA 構成詳細』を参照してください。

RAM を任意の CARMA セットアップに追加することは、現時点でも今後の時点でも可能です。既存のセットアップへの RAM の追加の詳細については、69 ページの『(オプション) 複数の RAM のサポート』を参照してください。

CA Endeavor® SCM RAM による CRASTART

このシナリオの記載内容は、以下の指定で CARMA をセットアップする方法を説明したものです。

- サーバー始動: CRASTART 方式。この方式では、CRASTART が LPA 内にある必要があります。
- RAM: CA Endeavor® SCM RAM。

指定が異なる他のシナリオのいずれかを使用する場合、このカスタマイズ・ステップは省略できます。

CARMA VSAM データ・セットの作成

CARMA 関連の VSAM データ・セットを定義し、それらのデータ・セットにデータを設定するには、以下の JCL ジョブをカスタマイズして実行依頼します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、59 ページの『CARMA VSAM データ・セット』を参照してください。

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

CRASRV.properties のカスタマイズ

RSE サーバーは、/etc/zexpl/CRASRV.properties 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、TSO **OEDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動してください。

デフォルトのファイル・ロケーションを使用する場合、必要な変更は、clist.dsname ディレクティブの値を *CRASTART に変更すること、および crastart.configuration.file の値を crastart.endavor.conf に変更することのみです。各種ディレクティブの詳細については、49 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。

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

図 3. CRASRV.properties: CA Endeavor® SCM RAM による CRASTART

crastart.endevor.conf のカスタマイズ

CRASTART は、`/etc/zexpl/crastart.endevor.conf` の定義を使用して CARMA サーバーを始動するのに有効な TSO/ISPF 環境を作成します。このファイルは、**TSO OEDIT** コマンドで編集できます。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、ファイル内のドキュメンテーションを参照してください。CRASTART 始動方式の詳細については、53 ページの『`crastart*.conf`、CRASTART サーバーの始動』を参照してください。

注: ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。

```

* 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.SFELLOAD,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. )

```

図 4. crastart.endevor.conf: CA Endeavor® SCM RAM による CRASTART

(オプション) 追加の CA Endeavor® SCM RAM のカスタマイズ

CA Endeavor® SCM RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CARMA 始動処理にはオプションのユーザー出口があります。詳しくは、(オプション) CARMA ユーザー出口を参照してください。

- CA Endeavor® SCM RAM には、カスタマイズが可能な複数の構成ファイル FEL.#CUST.PARMLIB(CRA*) があります。詳しくは、62 ページの『CRACFG、CRASCL、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』を参照してください。
- CA Endeavor® SCM RAM には割り振り exec FEL.SFELPROC(CRANDVRA) があり、これもカスタマイズできます。詳しくは、64 ページの『CRANDVRA、CA Endeavor® SCM RAM 初期割り振り exec』を参照してください。
- CA Endeavor® SCM RAM は、バッチ・モードでの CA Endeavor® SCM アクションの実行をサポートしています。バッチ・アクションには構成ファイル FEL.#CUST.PARMLIB(CRABCFG) およびスケルトン JCL の FEL.#CUST.CNTL (CRABATCA) が必要で、これらはカスタマイズする必要があります。詳しくは、64 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。

サンプル RAM による CRAFTSTART

このシナリオの記載内容は、以下の指定で CRAFTMA をセットアップする方法を説明したものです。

- サーバー始動: CRAFTSTART 方式。この方式では、CRAFTSTART が LPA 内にある必要があります。
- RAM: サンプル RAM (実動環境で使用するものではありません)

異なる指定で他のシナリオのいずれかを使用したい場合は、このカスタマイズ・ステップを迂回できます。

CRAFTMA VSAM データ・セットの作成

以下の JCL ジョブをカスタマイズして実行依頼し、CRAFTMA 関連の VSAM データ・セットを定義して取り込みます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、59 ページの『CRAFTMA VSAM データ・セット』および 60 ページの『CRAFTMA Repository Access Manager (RAM)』を参照してください。

CRAFTMA

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

サンプル RAM

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

CRASRV.properties のカスタマイズ

RSE サーバーは、`/etc/zexpl/CRASRV.properties` 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、**TSO OEDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動する必要があります。

デフォルトのファイル・ロケーションを使用する場合、必要な変更は、`clist.dsname` ディレクティブの値を `*CRASTART` に変更することのみです。各種ディレクティブの詳細については、49 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。

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

図 5. *CRASRV.properties*: サンプル RAM による *CRASTART*

crastart.conf のカスタマイズ

CRASTART は、`/etc/zexpl/crastart.conf` の定義を使用して、CARMA サーバーを始動するのに有効な TSO/ISPF 環境を作成します。このファイルは、**TSO OEDIT** コマンドで編集できます。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、ファイル内のドキュメンテーションを参照してください。CRASTART 始動方式の詳細については、53 ページの『`crastart*.conf`、CRASTART サーバーの始動』を参照してください。

```

* 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
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                                    * CRAALLOC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

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

```

図6. crastart.conf: サンプル RAM による CRASTART

注: ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。

(オプション) 追加のカスタム RAM のカスタマイズ

カスタム RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CARMA 始動処理にはオプションのユーザー出口があります。詳しくは、(オプション) CARMA ユーザー出口を参照してください。
- カスタム RAM 始動には、割り振り exec FEL.SFELPROC(CRAALLOC) があり、これもカスタマイズできます。詳しくは、CRAALLOC、カスタム RAM 割り振り exec を参照してください。

CA Endeavor® SCM RAM によるバッチ実行依頼

このシナリオの記載内容は、以下の指定で CARMA をセットアップする方法を説明したものです。

- サーバー始動: バッチ実行依頼方式。この方式には JES イニシエーターが必要です。
- RAM: CA Endeavor® SCM RAM。

指定が異なる他のシナリオのいずれかを使用する場合、このカスタマイズ・ステップは省略できます。

CARMA VSAM データ・セットの作成

以下の JCL をカスタマイズして実行依頼し、CARMA 関連の VSAM データ・セットを定義して取り込みます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、59 ページの『CARMA VSAM データ・セット』を参照してください。

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

CRASRV.properties のカスタマイズ

RSE サーバーは、/etc/zexpl/CRASRV.properties 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、TSO **EDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動する必要があります。

デフォルトのファイル・ロケーションを使用する場合、必要な変更は、clist.dsname ディレクティブの値を FEL.#CUST.CNTL(CRASUBCA) に変更することのみです。各種ディレクティブの詳細については、49 ページの『CRASRV.properties、CARMA への RSE インターフェース』を参照してください。

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

図 7. CRASRV.properties - CA Endeavor[®] SCM RAM によるバッチ実行依頼

CRASUBCA のカスタマイズ

FEL.#CUST.CNTL(CRASUBCA) CLIST と組み込み JCL が CARMA サーバーに実行依頼します。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。バッチ実行依頼始動方式の詳細については、56 ページの『CRASUB*、バッチ実行依頼サーバー始動』を参照してください。

```

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)
//C1EXMSG1 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)

```

図 8. CRASUBCA: CA Endeavor® SCM RAM によるバッチ実行依頼

(オプション) 追加の CA Endeavor® SCM RAM のカスタマイズ

CA Endeavor® SCM RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CARMA 始動処理にはオプションのユーザー出口があります。詳しくは、71 ページの『(オプション) CARMA ユーザー出口』を参照してください。
- CA Endeavor® SCM RAM には、FEL.#CUST.PARMLIB(CRACFG)、FEL.#CUST.PARMLIB(CRASHOW) および FEL.#CUST.PARMLIB(CRATMAP) という、カスタマイズ可能な複数の構成ファイルがあります。詳しくは、62 ページの『CRACFG、CRASCL、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』を参照してください。
- CA Endeavor® SCM RAM には割り振り exec FEL.SFELPROC(CRANDVRA) があり、これもカスタマイズできます。詳しくは、64 ページの『CRANDVRA、CA Endeavor® SCM RAM 初期割り振り exec』を参照してください。
- CA Endeavor® SCM RAM は、バッチ・モードでの CA Endeavor® SCM アクションの実行をサポートしています。バッチ・アクションには構成ファイル FEL.#CUST.PARMLIB(CRABCFG)、スケルトン JCL の FEL.#CUST.CNTL(CRABATCA)、およびオプションのデフォルト・ジョブ・カード FEL.#CUST.CNTL(CRABJOBC) が必要で、これらはカスタマイズする必要があります。詳しくは、64 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。

サンプル RAM によるバッチ実行依頼

このシナリオの記載内容は、以下の指定で CARMA をセットアップする方法を説明したものです。

- サーバー始動: バッチ実行依頼方式 (JES イニシエーターが必要)
- RAM: サンプル RAM (実動環境で使用するものではありません)

指定が異なる他のシナリオのいずれかを使用する場合、このカスタマイズ・ステップは省略できます。

VSAM データ・セットの作成

以下の JCL ジョブをカスタマイズして実行依頼し、CARMA 関連の VSAM データ・セットを定義して取り込みます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。既存の VSAM データ・セットは置き換えられます。

このステップの詳細については、59 ページの『CARMA VSAM データ・セット』および 60 ページの『CARMA Repository Access Manager (RAM)』を参照してください。

CARMA

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

サンプル RAM

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

CRASRV.properties のカスタマイズ

RSE サーバーは、/etc/zexpl/CRASRV.properties 内の設定を使用して CARMA サーバーの始動および接続を行います。このファイルは、TSO **0EDIT** コマンドで編集できます。変更を有効にするには、RSED 開始タスクを再始動する必要があります。

| デフォルトのファイル・ロケーションを使用する場合、必要な変更は、clist.dsname
| ディレクティブの値を FEL.#CUST.CNTL(CRASUBMT) に変更することのみです。各種
| ディレクティブの詳細については、49 ページの『CRASRV.properties、CARMA へ
| の RSE インターフェース』を参照してください。

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

図 9. CRASRV.properties: サンプル RAM によるバッチ実行依頼

CRASUBMT のカスタマイズ

FEL.#CUST.CNTL(CRASUBMT) CLIST と組み込み JCL が CARMA サーバーに実行依頼します。変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。バッチ実行依頼始動方式の詳細については、56 ページの『CRASUB*、バッチ実行依頼サーバー始動』を参照してください。

```

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

```

図 10. CRASUBMT: サンプル RAM によるバッチ実行依頼

(オプション) 追加のカスタム RAM のカスタマイズ

カスタム RAM には、必要に応じてカスタマイズが可能な追加のコンポーネントがあります。

- CARMA 始動処理にはオプションのユーザー出口があります。詳しくは、(オプション) CARMA ユーザー出口を参照してください。
- カスタム RAM 始動には、割り振り `exec FEL.SFELPROC(CRAALLOC)` があり、これもカスタマイズできます。詳しくは、CRAALLOC、カスタム RAM 割り振り `exec` を参照してください。

CARMA 構成詳細

本書に記載のさまざまな構成シナリオでは、多数の CARMA 構成ファイルを共有しています。これら構成ファイルの詳細をここに記載しますが、これらは、さまざまなシナリオ内部から参照されています。

CRASRV.properties、CARMA への RSE インターフェース

CARMA サーバーは、ホスト・システムを使用して 1 つ以上の Software Configuration Manager (SCM) にアクセスする他の製品のために標準 API を提供します。しかし、クライアント・コンピューターと直接通信するための方式は提供しません。このような通信については、RSE サーバーなどの他の製品に依存しています。RSE サーバーは、CRASRV.properties 内の設定を使用して CARMA サーバーの始動および接続を行います。

CRASRV.properties は /etc/zexpl/ に置かれます。ただし、FEL.SFELSAMP (FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更を有効にするには、RSED 開始タスクを再始動する必要があります。

```
# 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
```

図 11. CRASRV.properties - CARMA 構成ファイル

clist.dsname

CARMA サーバーの始動方式を定義します。各種始動方式の詳細については、36 ページの『サーバー始動方式とアクティブ RAM の選択』を参照してください。

- *CRASTART は、CRASTART を使用して、CARMA サーバーを RSE 内のサブタスクとして始動するよう指示します。*CRASTART を指定する場合は、crastart.* ディレクティブも指定する必要があります。指定しない場合、デフォルト値が使用されます。
- *ISPF は、ISPF の TSO/ISPF クライアント・ゲートウェイを使用して CARMA サーバーを始動するよう指示します。この始動方式は推奨されなくなりました。
- それ以外のすべての値は、TSO のような命名規則を使用して、CRASUBMT CLIST のロケーションを定義します。単一引用符 (') を付けた場合、そのデータ・セット名は絶対参照となり、単一引用符 (') を付けなかった場合、そのデータ・セット名には TSO 接頭部ではなく、クライアントのユ

ーザー ID が接頭部として付加されます。後者の場合は、すべての CARMA ユーザーが独自の CRASUBMT CLIST を保守する必要があります。

デフォルトは、CARMA が構成されていないことを示すヌル・ストリングです。

crastart.configuration.file

CRASTART 構成ファイルの名前を指定します。デフォルトは `crastart.conf` です。このファイルは、CARMA サーバーを始動するために必要なデータ・セット割り振りとプログラム呼び出しを指定します。このディレクティブは、`clist.dsname` ディレクティブの値が `*CRASTART` である場合にのみ使用されます。ファイル名はいくつかの方法で指定できます。

- ヌル・ストリング。これは変数が指定されていないことを意味します。デフォルト値が使用されます。
- ファイル名のみ。これがデフォルトの方式です。CARMA は、構成ディレクトリー (デフォルトでは `/etc/zexpl`) を検索してファイルを見付けます。
- 相対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まらないもの)。CARMA が構成ディレクトリー (デフォルトでは `/etc/zexpl/`) を提供されたパスに追加して絶対パスにします。
- 絶対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まるもの)。CARMA は、指定されたファイル・ロケーションを使用します。

#port.start

`port.start` の値が 0 (ゼロ) である場合、CARMA は、CARMA と RSE サーバーの間の通信に一時ポートを使用します。このシナリオでは、TCP/IP はランダムな空きポート番号を割り当てます。`port.start` の値がゼロ以外の場合、その値は、CARMA と RSE サーバーの間の通信に使用されるポート範囲の開始点として解釈されます。この場合、`port.range` 変数も定義する必要があります。デフォルトのポートは 0 です。ポート範囲の開始を指定するには、コメントを外してカスタマイズします。このポート上の通信は、ご使用のホスト・システムだけに限定されます。

注: ポートを選択する前に、**NETSTAT** コマンドおよび **NETSTAT PORTL** コマンドを使用して、そのポートがシステム上で使用可能であることを確認してください。詳しくは、「ホスト構成リファレンス」(SC43-2912) の『予約済み TCP/IP ポート』を参照してください。

#port.range

`port.start` がゼロ以外の場合に CARMA 通信に使用される、`port.start` から始まるポートの範囲。デフォルトは 100 です。ポート範囲のサイズを指定するには、コメントを外してカスタマイズします。例えば、`port.start` が 5227 で、`port.range` が 100 の場合、CARMA はポート 5227 から 5326 (両端の値を含む) までを使用できます。各 CARMA 接続でポートが排他的に使用されるため、ポート範囲を指定することで同時 CARMA セッションの最大数が制限されます。

#user.exit

CARMA 始動時に実行されるユーザー指定コードを定義します。コメントを外し、実行するコードのデータ・セット名を指定します。

引用符 (') を付けた場合、そのデータ・セット名は絶対参照となり、引用符 (') を付けなかった場合、そのデータ・セット名には TSO 接頭部ではなく、クライアント・ユーザー ID が接頭部として付加されます。後者の場合は、すべての CARMA ユーザーが独自の出口コードを保守する必要があります。

FEL.SFELSAMP(CRAEXIT) というサンプルのユーザー出口が提供されています。このサンプルでは、ユーザー出口に渡される始動引数について説明しています。詳しくは、(オプション) CARMA ユーザー出口を参照してください。

startup.script.name

CARMA 始動スクリプトを定義します。デフォルトは `carma.startup.rex` です。この REXX exec は CARMA サーバーの始動をトリガーします。ファイル名はいくつかの方法で指定できます。

- ヌル・ストリング。これは変数が指定されていないことを意味します。この場合、デフォルト値が使用されます。
- ファイル名のみ。これがデフォルトの方式です。CARMA は、PATH 環境変数内のディレクトリーを検索してファイルを見付けます。Developer for z Systems 実行可能ファイルを保持しているディレクトリー (デフォルトでは `/usr/lpp/ibm/rdz/bin`) は自動的に PATH 環境変数に追加されます。
- 相対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まらないもの)。CARMA が構成ディレクトリー (デフォルトでは `/etc/zexpl/`) を提供されたパスに追加して絶対パスにします。
- 絶対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まるもの)。CARMA は、指定されたファイル・ロケーションを使用します。

#crastart.stub

CRASTART を呼び出すための z/OS UNIX スタブ。デフォルトは CRASTART です。このスタブは、MVS ベースの CRASTART ロード・モジュールを z/OS UNIX プロセスから使用できるようにします。特定のパスを指定するには、コメントを外してカスタマイズします。このディレクティブは、`clist.dsname` ディレクティブの値が *CRASTART である場合にのみ使用されます。ファイル名はいくつかの方法で指定できます。

- ヌル・ストリング。これは変数が指定されていないことを意味します。デフォルト値が使用されます。
- ファイル名のみ。これがデフォルトの方式です。CARMA は、PATH 環境変数内のディレクトリーを検索してファイルを見付けます。Developer for z Systems 実行可能ファイルを保持しているディレクトリー (デフォルトでは `/usr/lpp/ibm/rdz/bin`) は自動的に PATH 環境変数に追加されます。

- 相対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まらないもの)。CARMA が構成ディレクトリー (デフォルトでは /etc/zexpl/) を提供されたパスに追加して絶対パスにします。
- 絶対パス (ディレクトリーやファイル名などで先頭がスラッシュ (/) から始まるもの)。CARMA は、指定されたファイル・ロケーションを使用します。

#crastart.syslog

CRASTART で CARMA サーバーを始動するときに、どれくらいの量の情報をシステム・ログに書き込むかを指定します。デフォルトは Partial です。有効な値を以下の表に示します。

A (All)	すべてのトレース情報を SYSLOG に出力します。
P (Partial)	接続、切断、およびエラー情報のみを SYSLOG に出力します。
それ以外のすべて	エラー条件のみを SYSLOG に出力します。

システム・ログ・メッセージに必要な詳細レベルを指定するには、コメントを外してカスタマイズします。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

#crastart.timeout

アクティビティーがないために CARMA サーバーが終了するまでの時間の長さ (秒単位)。デフォルトは 420 (7 分) です。必要なタイムアウト値を指定するには、コメントを外してカスタマイズします。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

注: SMFPRMxx parmlib メンバー内の JWT パラメーターの設定が CRASRV.properties の crastart.timeout 値より小さい場合は、モジュール CRASERV のシステム異常終了 522 が発生します。このオカレンスによって、CARMA の操作が影響を受けることはありません。必要であれば、サーバーが自動的に再始動されるためです。

#crastart.steplib

rdz.env 内の STEPLIB ディレクティブを通じてアクセスする場合の CRASTART モジュールのロケーション。デフォルトは FEL.SFELLPA です。CRASTART モジュールを LPA および LINKLIST のいずれにも含めることができない場合は、このディレクティブのコメントを外してカスタマイズします。CRASTART モジュールが LPA 内にない場合は、プログラム制御および APF の問題が起きる可能性があります。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

#crastart.tasklib

crastart.conf 内の TASKLIB DD 名の代替名。デフォルトは TASKLIB です。使用している SCM または RAM にとって DD 名 TASKLIB が特殊な意味を持ち、STEPLIB の代わりとして使用できない場合は、このディレクティブのコメントを外してカスタマイズします。このディレクティブは、clist.dsname ディレクティブの値が *CRASTART である場合にのみ使用されます。

crastart*.conf、CRASTART サーバーの始動

RSE は CRASTART ロード・モジュールを始動します。このモジュールは crastart*.conf の定義を使用して、TSO バッチ・コマンドおよび ISPF バッチ・コマンドを実行するための有効な環境を作成します。Rational Developer for z Systems は、その環境を使用して CARMA サーバー CRASERV を実行します。

crastart*.conf は /etc/zexpl/ に置かれます。ただし、ジョブ FEL.SFELSAMP(FELSETUP) をカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

Developer for z Systems は、複数の crastart*.conf 構成ファイルを提供しています。これらサンプル・ファイルはそれぞれ、特定のカスタマイズ・シナリオ用に事前構成されています。

- crastart.endevor.conf は、CA Endevor® SCM RAM による CRASTART の始動用に構成されています。
- crastart.conf は、サンプル RAM による CRASTART の始動用に構成されています。

crastart*.conf ファイルの機能は、概念上は JCL ジョブ・ストリームに似ていますが、動作はもっと限定的です。

- 以下のサンプルに、有効な行形式を示します。
 - * comment
 - ddname=dsn1,dsn2,dsn3 * comment
 - ddname=SYSOUT(c) * comment
 - ddname=DUMMY * comment
 - -COMMAND=<any bpxwdyn command> * comment
 - PROGRAM = progname parms * comment

注: **BPXWDYN** コマンドは、「*REXX* および *z/OS UNIX* システム・サービスの使い方」(SA88-8644) に記載されており、複雑な割り振り構成が可能です。

- すべての入力は大文字に変更されます。
- 行の継続はサポートされていません。
- 行の長さに制限はありません。
- 等号 (=) の前後に 1 つ以上のブランク・スペースが許可されます。
- DD 割り振りは、関連する PROGRAM ステートメントの前になければなりません。
- ここで割り振られる DD 名は、プログラム実行の終わりで解放されます。これらは累積されません。
- 呼び出されたプログラムにより割り振られる DD 名は解放されません。
- 複数のデータ・セットを DD 名に連結できます。データ・セット名はコンマ (,) で区切る必要があり、連結はリストされた順序で検索されます。

- すべてのデータ・セット割り振りは DISP=SHR で行われますが、 -COMMAND を使用して行われる割り振りを除きます。
- インライン・データはサポートされません。すべてのデータがカタログされたファイルに入っている必要があります。
- 変数は、等号 (=) の右側でしか使用できません。
- 以下の変数がサポートされています。

&CRAUSER.	クライアント・ユーザー ID
&CRADATE.	Dyyyyddd 形式 (7 文字のユリウス日付) の現在日付。
&CRATIME.	Thhmmss 形式 (時分秒) の現在時刻。
&CRAPRM1.	ポート番号
システム・シンボル	任意の SYS1.PARMLIB(IEASYMxx) システム・シンボル
-<ddname>	前に定義した DD 名の前にハイフン (-) を付加すると、JCL の *.ddname 逆方向参照のように機能します。オリジナルの DD は、-COMMAND ステートメントを使用して割り振る必要があります。

注: TSO 接頭部の変数は存在しません。TSO は構成ファイルが解釈される時、アクティブではないからです。TSO 接頭部または使用できないその他の変数が必要な場合は、70 ページの『(オプション) カスタム割り振り exec』を参照してください。

55 ページの図 12 は、ISPF サービスを含む基本 crastart*.conf スケルトンを示しています。


```

* 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 * CRAALLOC
SYSTSIN = DUMMY
SYSTSPRT= SYSOUT(H)

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

```

図 12. *crastart*.conf*: *CRASTART* を使用した *CARMA* サーバーの始動

注:

- ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。
- **PROGRAM** 行を変更する場合は、ストリングの処理を簡素化するために、**PARM()** ステートメントの右丸括弧 (“)”) の前に少なくとも 1 つの空白を必ず置いてください。
- **CARMA TSO** 環境をカスタマイズするために、独自の **DD** ステートメントとデータ・セット連結を追加し、**TSO** ログオン・プロシーチャーを模倣することができます。
- **DD** 名 **TASKLIB** は、**JCL** の **STEPLIB** の役割を果たします。この **DD** 名は、**CRASRV.properties** の **crastart.tasklib** に指定された値に一致している必要があります。これについては、49 ページの『**CRASRV.properties**、**CARMA** への **RSE** インターフェース』で説明します。
- **TASKLIB** 割り振りには通常の **APF** 規則が当てはまります。連結の一部に非 **APF** 許可ライブラリーがあると、ライブラリーは **APF** 許可を失います。
- **SMFPRMxx parmlib** メンバー内の **JWT** パラメーターの設定が **CRASRV.properties** の **crastart.timeout** 値より小さい場合は、モジュール **CRASERV** のシステム異常終了 522 が発生します。システム異常終了によって **CARMA** の操作が影響を受けることはありません。必要であればサーバーが自動的に再始動するためです。
- サーバーの終了時に、**CARMA** サーバー始動の詳細は、**rsecomm.log** に示されます。**rsecomm.log** の詳細レベルの設定について詳しくは、「*IBM Explorer for z/OS Host Configuration Guide (SC27-8437)*」を参照してください。

CRASTART ログ・ファイルの収集

CRASTART は、TSO 環境を RSE の子プロセスとして作成します。これは、別個のアドレス・スペースとして稼働します。CARMA 出力を SYSOUT(*) に送信し続けるために、簡単ではないアクションが必要になる場合があります。このことにより、ログ・ファイルの収集が複雑になります。この問題は、以下の割り振りの例に示すように、ログ・ファイルをユーザー固有のデータ・セットに書き込むことで解決します。

```
-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)
```

注:

- ページ幅の制約上、サンプルのいくつかの行は次の行に折り返されています。インデントで始まる行はすべて、前の行の終わりに追加しなければなりません。
- ユーザー固有のログ・ファイルを作成できるようにするには、このログ・ファイルを -COMMAND ステートメントを使用して割り振る必要があります。
- 柔軟性を高める必要がある場合、例えばログを特定のユーザーのデータ・セットにのみ送信する場合など、ログ・データ・セットを割り振り exec に割り振ることもできます。割り振り exec の詳細については、70 ページの『(オプション) カスタム割り振り exec』を参照してください。

ログ・ファイルを SYSOUT に書き込む場合、z/OS UNIX プロセスによって割り振られた SYSOUT は、JES では特殊な出力として扱われることに留意してください。これは、APPC トランザクションによって割り振られた SYSOUT に似ています。

- CARMA サーバーがまだアクティブなときに、SDSF の DA コマンドを使用して出力を確認できます。ジョブには、ユーザーのユーザー ID の後にジョブ名としてのランダムな 1 桁の数字と、STC ジョブ ID を続けたものが与えられます。ユーザーは、ジョブ所有者になります。
- 非アクティブ状態またはユーザーによる接続の終了が原因で CARMA サーバーが終了する際に、出力が HOLD 出力クラスに書き込まれた場合は、SDSF の APPCON コマンドおよび H ALL コマンドを使用して出力を確認できます。ジョブ名、ジョブ ID、およびジョブ所有者は同じままです。各 DD は個別のスプール・ファイルとして表示され、どの DD であるかは示されません。
- FEJJCNGF で SEARCHALL=ON がアクティブであり、出力が HOLD 出力クラスのスプール上にある場合にも、JES ジョブ・モニターは出力を表示できます。SEARCHALL ディレクティブの詳細については、「IBM Explorer for z/OS Host Configuration Guide (SC27-8437)」を参照してください。

CRASUB*、バッチ実行依頼サーバー始動

RSE は CLIST CRASUB* を始動します。すると、始動された CRASUB* が、組み込み JCL を実行依頼して、TSO バッチ・コマンドおよび ISPF バッチ・コマンドを実行するための有効な環境を作成します。Rational Developer for z Systems は、その環境を使用して CARMA サーバー CRASERV を実行します。

CRASUB* は FEL.#CUST.CNTL に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

注: 変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

Developer for z Systems は、複数の CRASUB* JCL ジョブを提供します。これらサンプル・ファイルはそれぞれ、特定のカスタマイズ・シナリオ用に事前構成されています。

- CRASUBCA は、CA Endeavor® SCM RAM によるバッチ始動用に構成されています。
- CRASUBMT は、サンプル RAM によるバッチ始動用に構成されています。

58 ページの図 13 は、ISPF サービスを含む基本 CRASUB* スケルトンを示しています。

```

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

```

図 13. CRASUB*: バッチ実行依頼を使用した CARMA の始動

注:

- SYSTSIN データを変更する場合は、ストリングの処理を簡素化するために、
PARM() ステートメントの右丸括弧 (") の前に少なくとも 1 つの空白を必ず置いてください。
- CARMA TSO 環境をカスタマイズするために、独自の DD ステートメントとデータ・セット連結を追加し、TSO ログオン・プロシーチャーを模倣することができます。
- オプションで、CRASUB* CLIST 内の SET CRAPRM2=420 行のコメントを外して変更することにより、CARMA のタイムアウト値を変更できます。このタイムアウト値は、CARMA がクライアントからの次のコマンドを待つ秒数です。0 の値を設定すると、デフォルトのタイムアウト値になります。つまり、現行の 420 秒 (7 分) です。

- サーバーの終了時に、CARMA 始動プロセスの詳細は、rsecomm.log に示されます。rsecomm.log の詳細レベルの設定について詳しくは、「*IBM Explorer for z/OS Host Configuration Guide (SC27-8437)*」を参照してください。

CARMA VSAM データ・セット

CARMA サーバーは、3 つの VSAM データ・セットへの READ アクセス権限を必要とします。これらの VSAM データ・セットを作成し、取り込むためのサンプル・メンバーは、FEL.#CUST.JCL に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

注:

- (カスタムの) RAM の定義を既存の VSAM 構成にマージする必要がある場合は、サンプル・ジョブ FEL.#CUST.JCL(CRA#UADD) を参照してください。このジョブは、変更する CARMA VSAM ファイルごとにカスタマイズし、実行依頼する必要があります。さまざまな CARMA VSAM ファイルで使用されるレコード構造の詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660) を参照してください。
- VSAM から順次データ・セットへアクティブな定義を抽出するには、FEL.#CUST.JCL(CRA#UQRY) サンプル・ジョブを使用します。

CRADEF、構成データ・セット

VSAM データ・セットは、定義された RAM がサポートする機能を記述します。RAM 開発者はこのデータ・セットへの UPDATE アクセス権限が必要です。データ・セットは、以下のいずれかのサンプル・ジョブにより作成できます。

- CRA\$VCAD は CA Endevor® SCM RAM データをデータ・セットに取り込みます。
- CRA\$VDEF はサンプル RAM データをデータ・セットに取り込みます。

前述のサンプル・ジョブを使用して、定義された RAM を VSAM 作成中に使用不可にすることができます。これを行うことによって、単一のマスター入力ファイル (IBM によって提供されたもの、または RAM 開発者によってカスタマイズされたもの) を使用して、カスタマイズされた CARMA セットアップを作成できます。

CRAMSG、メッセージ・データ・セット

この VSAM データ・セットは、CARMA サーバー自体が発行したメッセージを保持します。データ・セットは、以下のいずれかのサンプル・ジョブにより作成できます。

- CRA\$VMSG は汎用サーバー・データをデータ・セットに取り込みます。

CRASTRS、カスタム・ストリング・データ・セット

この VSAM データ・セットは、定義された RAM が発行したメッセージを保持します。RAM 開発者はこのデータ・セットへの UPDATE アクセス権限が必要です。データ・セットは、以下のいずれかのサンプル・ジョブにより作成できます。

- CRA\$VCAS は CA Endevor® SCM RAM データをデータ・セットに取り込みます。
- CRA\$VSTR はサンプル RAM データをデータ・セットに取り込みます。

CARMA Repository Access Manager (RAM)

Repository Access Manager (RAM) は、z/OS ベースの Software Configuration Manager (SCM) 用のアプリケーション・プログラミング・インターフェース (API) です。Developer for z Systems またはユーザー作成アプリケーションは CARMA サーバーを始動でき、CARMA サーバーは RAM をロードし、サポートされる SCM があればそれにアクセスする標準インターフェースを提供します。

CARMA サーバーは、LINKLIST または STEPLIB/TASKLIB を介して、RAM ロード・モジュールを見つけることができなければなりません。

Developer for z Systems により提供される CRAR* RAM ロード・モジュールは FEL.SFELLOAD に置かれ、サンプルのソース・コードおよびコンパイル・ジョブは FEL.SFELSAMP に置かれます。ただし、Developer for z Systems の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。

以下のセクションには、Developer for z Systems で使用できる RAM に関するカスタマイズ上の注意があります。参照するサンプルのメンバーは FEL.#CUST.* に置かれます。ただし、FEL.SFELSAMP(FELSETUP) サンプル・ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

CARMA に関する詳細な情報、および提供されているサンプル RAM およびサンプル・ソース・コードの詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660) を参照してください。

CA Endeavor® SCM RAM

- CA Endeavor® SCM RAM は実動タイプの RAM です。
- CA Endeavor® SCM RAM は、Developer for z Systems クライアントが CA Endeavor® SCM エlementに直接アクセスできるようにします。
- ロード・モジュール名は CRARNDVR です。
- CA Endeavor® SCM RAM には、標準的な CARMA のセットアップと比べて多数の追加設定があります。CA Endeavor® SCM RAM をサポートする事前構成済みのセットアップのいずれかを開始点として使用し、ニーズに合わせてカスタマイズしてください。
- CA Endeavor® SCM RAM には、カスタマイズが可能な複数の構成ファイルがあります。詳しくは、62 ページの『CRACFG、CRASCL、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル』を参照してください。
- CA Endeavor® SCM RAM には割り振り exec FEL.SFELPROC(CRANDVRA) があり、これもカスタマイズできます。詳しくは、64 ページの『CRANDVRA、CA Endeavor® SCM RAM 初期割り振り exec』を参照してください。
- CA Endeavor® SCM RAM は、(バックグラウンドにおける) バッチ・モードでの CA Endeavor® SCM アクションの実行をサポートしています。詳しくは、64 ページの『CA Endeavor® SCM RAM バッチ・アクション』を参照してください。

CA Endeavor® SCM パッケージ RAM

- CA Endeavor® SCM パッケージ RAM は、実動タイプの RAM です。

- CA Endeavor® SCM パッケージ RAM は、Developer for z Systems クライアントが CA Endeavor® SCM パッケージに直接アクセスできるようにします。
- ロード・モジュール名は CRARPKGS です。
- CA Endeavor® SCM パッケージ RAM にはカスタマイズ可能な設定がなく、CA Endeavor® SCM RAM と組み合わせて使用する必要があります。

PDS RAM

- PDS RAM はサンプル RAM です。実稼働環境では使用しないでください。
- PDS RAM は、「リモート・システム」ビューの「MVS ファイル」->「ユーザー・データ・セット」と同様のデータ・セット・リストを提供します。
- ロード・モジュール名は CRARPDS です。
- PDS RAM を使用するには、ISPF サービスが使用可能である必要があります。
- PDS RAM は、追加の VSAM データ・セットが DD CRARAM1 に割り振られている必要があります。この VSAM データ・セットは、FEL.#CUST.JCL(CRA#VPDS) サンプル・ジョブに割り振って、このジョブを準備することができます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。
- ソース・コードおよびコンパイル・ジョブは FEL.SFELSAMP にあります。詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660)を参照してください。

スケルトン RAM

- スケルトン RAM はサンプル RAM です。実稼働環境では使用しないでください。
- スケルトン RAM は、ユーザーが独自の RAM を開発する際の開始点として使用できる機能フレームワークを提供します。
- ロード・モジュール名は CRARTEST です。
- ソース・コードおよびコンパイル・ジョブは FEL.SFELSAMP にあります。詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660)を参照してください。

SCLM RAM

- SCLM RAM はサンプル RAM です。実稼働環境では使用しないでください。
- SCLM RAM は、ISPF の Software Configuration Manager である SCLM への基本的な入り口となります。この RAM は、デフォルトでは使用可能に設定されていません。
- ロード・モジュール名は CRARSCLM です。
- SCLM RAM を使用するには、ISPF サービスが使用可能である必要があります。
- SCLM RAM は、追加の VSAM データ・セットが DD CRARAM2 に割り振られている必要があります。この VSAM データ・セットは、FEL.#CUST.JCL(CRA#VSLM) サンプル・ジョブに割り振って、このジョブを準備することができます。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。
- SCLM RAM を使用するには、さまざまなユーザー固有のデータ・セットが存在している必要があります。FEL.#CUST.JCL(CRA#ASLM) をカスタマイズして、これ

らのデータ・セットを割り振ります。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。各ユーザーは、CARMA を SCLM RAM と一緒に使用する前に、1 回だけ CRA#ASLM を実行依頼する必要があります。そうしなかった場合は、割り振りエラーになります。

- SCLM RAM は、デフォルトでは使用可能に設定されていません。RAM を使用可能にするには、これを、DD CRADEF と CRASTRS が参照する CARMA VSAM データ・セットに定義する必要があります。FEL.#CUST.JCL(CRA#UADD) サンプル・ジョブを使用して、FEL.SFELVSM2(CRA0SLMD) を CRADEF へ、FEL.SFELVSM2(CRA0SLMS) を CRASTRS へマージします。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。
- ソース・コードおよびコンパイル・ジョブは FEL.SFELSAMP にあります。詳細については、「*Common Access Repository Manager Developer's Guide*」(SC23-7660) を参照してください。

CRACFG、CRASCL、CRASHOW、および CRATMAP、CA Endeavor® SCM RAM 構成ファイル

以下の CA Endeavor® SCM RAM に固有の CARMA コンポーネントは、選択したサーバー始動方式に関係なく、カスタマイズすることができます。参照するサンプルのメンバーは、FEL.#CUST.PARMLIB に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

CRACFG、CA Endeavor® SCM RAM と SCM の相互作用

CRACFG は、CA Endeavor® SCM RAM が CA Endeavor® SCM と相互作用する方法を指定します。デフォルトを変更する場合、カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

```
# 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)
```

図 14. CRACFG - CA Endeavor® SCM RAM と SCM の相互作用

CRASCL、CA Endeavor® SCM RAM テンプレート SCL

CRASCL は、CA Endeavor® SCM Packages の処理時に使用できるアクションとオプションを制限できる、テンプレート SCL (ソフトウェア制御言語) です。

これを使用すると、テンプレート SCL に明示的にリストされたアクションとオプションのみを、Developer for z Systems Packages Editor で指定できるようになります。デフォルトを変更する場合、カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

CRASHOW、CA Endeavor® SCM RAM のデフォルト・フィルター

CRASHOW は、CA Endeavor® SCM の環境やシステムなどに対するデフォルト・フィルターを定義します。デフォルトを変更する場合、カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

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

図 15. CRASHOW - CA Endeavor® SCM RAM のデフォルト・フィルター

注: FILTER-DEP は、共通の CA Endeavor® SCM 変数ではなく、他の CA Endeavor® SCM リポジトリ・ロケーションへのフットプリントの参照によってエレメントの依存関係のスキャンを制御する Developer for z Systems 固有の変数です。

CRATMAP、CA Endeavor® SCM RAM のファイル拡張子のマッピング

CRATMAP は、CA Endeavor® SCM タイプとファイル拡張子のマッピングをオーバーライドします。デフォルトを変更する必要がある場合は、メンバー内のドキュメンテーションにある、カスタマイズに関する説明を参照してください。

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

図 16. CRATMAP: CA Endeavor® SCM RAM のデフォルト・フィルター

CRANDVRA および CRADYNDA、CA Endeavor® SCM RAM 割り振り exec

以下の CA Endeavor® SCM RAM に固有の CARMA コンポーネントは、選択したサーバー始動方式に関係なく、カスタマイズすることができます。

特定のデフォルト (データ・セット名など) がご使用のサイトの標準に適合しない場合は、これらの割り振り REXX exec のコピーをカスタマイズできます。exec は FEL.SFELPROC に置かれます。ただし、Developer for z Systems の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。割り振り exec の詳細については、70 ページの『(オプション) カスタム割り振り exec』を参照してください。

注: サンプルの割り振り REXX を新しいデータ・セットにコピーし、そのコピーをカスタマイズして、保守の適用時に上書きされないようにしてください。これを行う場合は、選択した CARMA 始動方式の SYSEXEC DD で SFELPROC への参照を更新してご使用の新しいデータ・セット名に合わせる必要があります。

CRANDVRA、CA Endeavor® SCM RAM 初期割り振り exec

バッチ実行依頼方式および CRASTART 始動方式はどちらも、CRANDVRA REXX exec を呼び出して、CA Endeavor® SCM RAM で使用するユーザー固有のデータ・セットを割り振ります。割り振りは別個の exec で行われます。これは、exec を使用した方が、バッチ実行依頼 CRASUBCA JCL や CRASTART crastart.endeavor.conf 構成ファイル内部で行う場合より高い柔軟性が得られるためです。また、割り振り exec はオプションのユーザー出口の呼び出しも行います。

DD	データ・セット名	タイプ
DEPEND	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.DEPEND	永続的
BROWSE	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.BROWSE	一時的
BROWSEV	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.BROWSEV	一時的
ENHCEDIT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDIT	一時的
ENHCEDITV	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.ENHCEDITV	一時的
CIPRINT	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.LISTING	一時的
SPCLLIST	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.SPCLLIST	一時的
PKGSCLS	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.PKGSCLS	一時的
CRABJCLO	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRABJCLO	一時的
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$NDVR.CRAPARM	一時的

CRADYNDA、CA Endeavor® SCM RAM 再割り振り exec

CA Endeavor® SCM は、さまざまなレコード長の変数ブロック・データ・セットを処理することができ、Developer for z Systems は特定のアクションで同一の割り振りを使用する必要があります。バッチ実行依頼方式および CRASTART 始動方式はどちらも、CRADYNDA REXX exec を呼び出して、正しい属性を持つ作業データ・セットを割り振ります。

CA Endeavor® SCM RAM バッチ・アクション

「エレメントの生成」などの CA Endeavor® SCM アクションは通常、CARMA サーバー・アドレス・スペースで「オンライン」で実行されます。使用する CA Endeavor® SCM プロシーチャーで TSO を呼び出す場合、この動作により問題が発

生します。理由は、TSO が既にアクティブである、つまり、SYSTSIN や SYSTSPRT などの必要な DD が使用されているからです。

この問題を解決するために、CA Endeavor® SCM RAM は「バッチ・アクション」をサポートしています。バッチ・アクションを使用可能にすると、CA Endeavor® SCM RAM は、「エレメントの生成」などのアクションを実行するために、カスタマイズ可能なバッチ・ジョブを実行依頼します。バッチ・ジョブを使用した場合、CA Endeavor® SCM プロシーチャーによる SYSTSIN や SYSTSPRT などの DD の割り振りが行われます。これは、実行依頼された JCL では TSO がアクティブである必要はないためです。

CA Endeavor® SCM RAM バッチ・アクションは、バックグラウンドで実行する CA Endeavor® SCM アクションの Developer for z Systems 版です。

バッチ・アクションでサポートされるアクションを実行するように要求が出されると、CA Endeavor® SCM RAM は CRASUBCA または crastart.endeavor.conf で CRABCFG DD の有無を検査し、この DD のためのセットアップが有効か検査します。CRABCFG が存在し、そのセットアップが有効である場合、アクションはバッチで実行されます。CRABCFG が存在しない場合、アクションはオンラインで実行されます。Developer for z Systems クライアントは、この動作をオーバーライドする機能を備えています。

次に例を示します。

```
/* 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
```

注:

- TSO が含まれない環境は、選択された CA Endeavor® SCM アクションでのみ使用可能です。バッチ・アクションは、この範囲外の TSO が含まれない環境をサポートしません。
- CRABCFG 構成ファイルは、サポートされている CA Endeavor® SCM アクションについて説明します。
- 機能サンプル・ジョブ FEL.#CUST.CNTL(CRABATCA) は、バッチ・アクションを実行するために用意されていますが、それらのバッチ・アクションの目的はこのサンプルが現行の CA Endeavor® SCM プロシーチャーを開始するようにカスタマイズされていることを示すことです。
- バッチ・アクション JCL の実行依頼に使用されるクラスに、十分な JES イニシエーターがあることを確認してください。
- SYSPLEX 環境で JES を使用する場合、ジョブが現在のシステムで稼働すること、または完了の情報が Developer for z Systems をホスティングしているシステムに送られることを確認します。これで、CA Endeavor® SCM RAM は状況をチェックできます。
- Developer for z Systems クライアントは、実行依頼の前に、カスタマイズした JOB カードおよび追加の JCL ステートメントをバッチ・アクション JCL に指定することができます。

CRABCFG、CA Endeavor® SCM RAM バッチ・アクション構成

CRABCFG は、CA Endeavor® SCM RAM のバッチ・アクションに関連した構成変数を定義します。

CRABCFG は FEL.#CUST.PARMLIB に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

以下の CRABCFG サンプル・ファイルを参照してください。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。US コード・ページを使用する場合、コメント行は番号記号 (#) で始まります。ディレクティブとそれに割り当てられる値の後のコメントは、サポートされています。等号 (=) の前後のスペースはサポートされています。行の継続はサポートされていません。

注: 変更は、更新後に始動されたすべての CARMA サーバーについて有効になります。

```
# 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>
```

図 17. CRABCFG: CA Endeavor® SCM RAM バッチ・アクション構成

SKELETON-DD

バッチ・アクション・スケルトン JCL を保持する 1 つ以上の PDS(E) データ・セットを参照する DD ステートメントの名前。サンプル値は CRABSKEL です。必要であれば変更できます。この DD は、CRASUBCA または crastart.endeavor.conf で CARMA サーバーに対して定義する必要があります。

DEFAULT-JOBCARD

デフォルト JOB カードを持っているメンバーの名前。ユーザー固有の JOB カード (Developer for z Systems クライアントに格納されている) によって却下されない限り、スケルトン JCL では、<JOBCARD> キーはこのデフォルト JOB カードに置き換えられます。必要であれば変更できます。

GENERATE-ELEMENT およびその他の CA Endeavor® SCM アクション

キー名は、バッチ・アクションによってサポートされ、変更することができない CA Endeavor® SCM アクションを表します。それぞれのキーに割り当てられる値は、関連するスケルトン JCL のメンバー名です。すべてのキーのサンプル値は、CRABATCA です。必要であれば変更できます。

BSTIPT01-KEY

実際の CA Endeavor® SCM コマンド・ストリングの置換キー。サンプル値は <CRA_BSTIPT01> です。必要であれば変更できます。スケルトン JCL 内におけるこの置換キーの最初のオカレンス (コメントに含まれるものを除く) が、要求されたアクションを要求された要素に対して実行するように CA Endeavor® SCM に指示するコマンド・ストリングに置換されます。

CRABATCA、CA Endeavor® SCM RAM バッチ・アクション JCL

CRABATCA は、バッチ・アクションに使用されるサンプルのスケルトン JCL です。デフォルトを変更する場合は、メンバー内のドキュメンテーションでカスタマイズ手順を確認してください。

CRABATCA は FEL.#CUST.CNTL に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

変更は、すべての新規呼び出しについてアクティブになります。サーバーの再始動は必要ありません。

```
//<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>
/*
```

図 18. CRABATCA: CA Endeavor® SCM RAM バッチ・アクション JCL

CRABJOBC、CA Endeavor® SCM RAM バッチ・アクション JOB カード

CRABJOBC は、<JOB CARD> キーを指定するバッチ・アクション・スケルトン JCL に使用されるサンプルのデフォルト JOB カードです。デフォルトを変更する場合は、メンバー内のドキュメンテーションでカスタマイズ手順を確認してください。

CRABJOBC は FEL.#CUST.CNTL に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

変更は、すべての新規呼び出しについてアクティブになります。サーバーの再始動は必要ありません。

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

図 19. CRABJOBC: CA Endeavor® SCM RAM バッチ・アクション JOB カード

CRAALLOC、カスタム RAM 割り振り exec

バッチ実行依頼方式および CRASTART 始動方式はどちらも、CRAALLOC REXX exec を呼び出して、ユーザー作成 RAM で使用できるユーザー固有のデータ・セットを割り振ります。割り振りは別個の exec で行われます。これは、exec を使用した方が、バッチ実行依頼 CRASUBMT JCL や CRASTART crastart.conf 構成ファイル内部で行う場合より高い柔軟性が得られるためです。また、割り振り exec はオプションのユーザー出口の呼び出しも行います。

DD	データ・セット名	タイプ
CRAPARM	&SYSPREF..&SYSUID.. &SYSNAME..CRA\$CUST.CRAPARM	一時的

特定のデフォルト (データ・セット名など) がご使用のサイトの標準に適合しない場合は、この割り振り REXX exec のコピーをカスタマイズできます。CRAALLOC は FEL.SFELPROC に置かれます。ただし、Rational Developer for z Systems の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。

カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。割り振り exec の詳細については、(オプション) カスタム割り振り exec を参照してください。

注: サンプルの割り振り REXX を新しいデータ・セットにコピーし、そのコピーをカスタマイズして、保守の適用時に上書きされないようにしてください。これを行う場合は、選択した CARMA 始動方式の SYSEXEC DD で SFELPROC への参照を更新してご使用の新しいデータ・セット名に合わせる必要があります。

CARMA 戻りコード

CARMA は、さまざまなエラー・コードをクライアントに対して、またはホスト・システム・ログで報告できます。エラーとともに提供される詳細と、表 12 の情報は、エラーを見つけて解決に取り組むのに役立ちます。

表 12. CARMA 戻りコード

エラー範囲	エラー・タイプ
4 から 99	一般的な CARMA エラー
100 から 199	一般的な RAM エラー
200 から 399	CRASERV (CARMA サーバー) エラー
400 から 499	RSE (CARMA マイナー) エラー

表 12. CARMA 戻りコード (続き)

エラー範囲	エラー・タイプ
500 から 899	RAM 固有のエラー
900 から 999	TSO エラーおよび TCP/IP エラー

いくつかの一般的な戻りコードを以下に示します。

- 220: 非アクティブ・タイムアウトのため CARMA サーバーが終了します。これはエラーではありません。
- 990: CARMA マイナーが listen しているポートに CARMA サーバーが接続できません。

(オプション) 複数の RAM のサポート

CARMA は、複数の RAM を定義して同時に実行するための機能を備えています。ただし、1 ユーザーにつきアクティブな CARMA サーバーは 1 つのみであるため、複数の RAM がある場合でも、そのセットアップを機能させるために構成の変更が必要となることがあります。

RAM は、RAM 開発者によって CARMA 構成 VSAM データ・セット CRADEF 内で定義されます。CARMA サーバー CRASERV は、始動時に、定義されているすべての RAM を識別し、その情報を CARMA クライアントに送信します。これでユーザーは 1 つ以上の RAM を選択できるようになり、それらが CARMA サーバーにロードされます。

RAM は CARMA サーバーのプラグインとしてアクティブになるため、各 RAM の前提条件となるもの (データ・セット割り振りなど) がすべて CARMA サーバーのアドレス・スペースにあることを確認してください。この要件のために、Developer for z Systems に付属する CARMA 構成サンプル (CRASUBMT や crastart.conf など) の変更が必要になることがあります。

例

以下の例では、CA Endevor® SCM RAM を使用する既存のセットアップから CRASTART 始動方式を使用して始動し、サンプルの PDS RAM を追加します。

CA Endevor® SCM RAM 用の定義:

- FEL.SFELVSM2(CRA0VCAD): CRADEF 定義
- FEL.SFELVSM2(CRA0VCAS): CRASTRS 定義
- /etc/zexpl/crastart.endevor.conf: CRASTART 構成ファイル

PDS RAM 用の定義:

- FEL.SFELVSM2(CRA0VDEF): CRADEF 定義
- FEL.SFELVSM2(CRA0VSTR): CRASTRS 定義
- FEL.#CUST.CRARAM1: CRARAM1 定義

このプロセスは、システム・プログラマーがセットアップを完了するために必要とするデータと情報を、RAM 開発者が収集することから始まります。

1. PDS RAM に固有のデータを SFELVSM2 メンバーから抽出します。これらのメンバーは、PDS RAM だけでなくすべてのサンプル RAM の定義を保持しています。
2. このデータを CA Endeavor® SCM RAM SFELVSM2 メンバーとマージします。
3. PDS RAM 固有の前提条件のリストを作成します。
 - FEL.#CUST.CRARAM1 にリンクされた DD CRARAM1
 - TSO 環境

次に、システム・プログラマーがこのデータを使用して更新された CARMA VSAM データ・セットを作成し、前提条件情報を使用して、両方の RAM をサポートできる CRASTART 構成ファイルを作成します。

1. この結合データを、CRA\$VDEF および CRA\$VSTR ジョブへの入力として使用して、更新された CARMA 構成および CARMA カスタム情報 VSAM データ・セット (CRADEF と CRASTRS) を作成します。CRAMSG VSAM は CARMA サーバーに固有のものであり、したがって両方の RAM で同一です。
2. CRARAM1 定義を crastart.endeavor.conf に追加します。
`CRARAM1 = FEL.#CUST.CRARAM1`
3. crastart.endeavor.conf 内の PROGRAM ステートメントを検証して、両方の RAM に必要な環境を提供できることを確認します。
4. `PROGRAM=IKJEFT01 %CRANDVRA NDVRC1 PGM(CRASERV) PARM(&CRAPRM1. &CRAPRM2.)`
 - IKJEFT01: 無許可の環境で特定の許可された呼び出しを可能にするために使用され、また、CA Endeavor® SCM RAM 事前割り振り exec を実行する環境として使用される TSO。
 - %CRANDVRA: 一時的なユーザー固有の作業データ・セットを割り振る CA Endeavor® SCM RAM 事前割り振り exec (FEL.SFELPROC に配置)。
 - NDVRC1: TSO コマンドと ISPF コマンドを実行するメカニズムが組み込まれた、CA Endeavor® バックエンド。
 - PGM(CRASERV): CARMA サーバーを始動する、ISPF コマンド・フォーマットのコマンド。
 - PARM(&CRAPRM1. &CRAPRM2.): ISPF コマンド・フォーマットの CRASERV のパラメーター。&CRAPRM1 は使用されるポート、&CRAPRM2 はタイムアウト値です。

CA Endeavor® SCM RAM は ISPF 環境でアクティブになります。これは、PDS RAM に必要な TSO 環境も使用可能であることを意味します。

(オプション) カスタム割り振り exec

すべての CARMA サーバー始動方式には、データ・セットの割り振りに関する制約があります。例えば、TSO 接頭辞の置換は、JCL や CRASTART では使用できません。

ただし、TSO または ISPF の始動後、かつ CARMA の始動前に呼び出される exec を作成すると、TSO または ISPF で使用可能なすべての範囲の変数とサービスを使用して必要な割り振りを実行することが可能になります。

Rational Developer for z Systems は、この章で先に説明した各事前構成済みセットアップで割り振り exec を使用します。FEL.SFELPROC(CRANDVRA) (CA Endeavor® SCM RAM の割り振り exec) および FEL.SFELPROC(CRAALLOC) (カスタム RAM の割り振り exec) であり、exec は、ユーザーの TSO 接頭部を高位修飾子に持つ、カタログされた一時データ・セットを割り振ります。また、割り振り exec はオプションのユーザー出口の呼び出しも行います。

カスタマイズについての説明は、exec 内に記載されています。割り振り exec の変更はサポートされていますが、推奨されません。それは、PTF サービスによって exec が更新された場合に、カスタマイズを再度行う必要があるためです。可能であれば、代わりに CARMA ユーザー出口 (『(オプション) CARMA ユーザー出口』を参照) を使用してください。

注:

- 割り振り exec を更新するときは、CARMA 始動プロセスで既に行われた割り振りを、CRASTART または始動 JCL によって破壊することがないようにしてください。
- 割り振り exec により生成される出力は、CARMA サーバーの DD SYSTSPRT に示されます。

割り振り exec を更新するときは、CARMA 始動プロセスで既に行われた割り振りを、CRASTART または始動 JCL によって破壊することがないようにしてください。

以下のサンプルは、TSO のみを必要とする割り振り exec の始動方法を示しています。

```
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. )
//*
```

(オプション) CARMA ユーザー出口

CARMA はユーザー出口の呼び出しをサポートしており、CARMA サーバーの始動時に特殊な初期化を行ったり、シャットダウン時に特殊なクリーンアップを行ったりすることができます。ユーザー出口を使用することで、PTF サービスによって保守されている割り振り exec の変更の必要性が減ります。

ユーザー出口は割り振り exec によって呼び出され、2 回実行されます。初期化の呼び出しは、一時データ・セットが割り振られてから CARMA サーバーが呼び出されるまでの間に行われます。クリーンアップの呼び出しは、CARMA サーバーが終了してから一時ファイルが除去されるまでの間に行われます。最初の呼び出しが戻りコード 99 以上で終了すると、CARMA 始動は中断します。これは、CARMA サーバーも、このユーザー出口の 2 回目の呼び出しも実行されないことを意味します。

FEL.SFELSAMP(CRAEXIT) というサンプル・ユーザー出口が提供されます。ただし、Developer for z Systems の SMP/E インストール時に別の高位修飾子を使用した場合は除きます。このサンプル・ユーザー出口では、ユーザー出口に渡される以下の始動引数について詳しく説明しています。

始動引数	説明
(STARTUP) (ENDING)	出口呼び出しが CARMA サーバーの呼び出しの前であるか後であるかの標識。
EXIT_RC=rc	出口の前の呼び出しの戻りコード。 (STARTUP) 呼び出しのときは rc は常に 0 です。
CARMA_RC=rc	CARMA サーバーの呼び出しの戻りコード。 (STARTUP) 呼び出しのときは rc は常に 0 です。
...	CARMA サーバーの始動コマンドと始動引数。例: ISPSTART PGM(CRASERV) PARM(1312 420 EXIT=CRAEXIT CLIENT=9.0.1)

ユーザー出口によって生成された出力は、CARMA サーバーの DD SYSTSPRT に表示されます。

(オプション) IRXJCL と CRAXJCL

TSO (IKJEFTxx) を使用して CARMA サーバーを始動する場合に、使用する RAM でサービスを読み出し、さらにそのサービスで IRXJCL REXX バッチ・インターフェースを読み出すと、問題が発生することがあります。問題が発生する可能性があるのは、RAM によって読み出されたプロセッサが、以前は TSO なしで稼働していたかオンライン TSO 内でのみ稼働しており、それが DD SYSTSIN または SYSTSPRT を動的に割り振ったときです。この問題を回避するために、サンプル・プログラム CRAXJCL が提供されています。

プロセッサは、SYSTSIN または SYSTSPRT (IRXJCL に必須) を割り振ろうとした場合に失敗することがあります。これは、CARMA に必須のバッチ TSO が既にそれらの DD 名を割り振って開いているためです。CRAXJCL 置換モジュールは SYSTSIN および SYSTSPRT を DUMMY に割り振ろうとしますが、割り振りが失敗した場合に発生するエラーを無視します。その後、IRXJCL を呼び出して、実際の処理を行います。

このことは、TSO によって始動された CARMA 環境内でプロセッサを稼働した場合、SYSTSIN および SYSTSPRT への割り振りは CARMA によって使用される割り振りと同じものであることを意味します。TSO/CARMA の外部でプロセッサを稼働した場合、SYSTSIN および SYSTSPRT の割り振りは CRAXJCL によって作成されます。したがって、使用するプロセッサは、SYSTSIN へ割り振られたデータ・セットの内容に依存してはなりません。

IRXJCL の呼び出しは、「*TSO/E REXX 解説書*」(SA88-8635) で説明されているように、PARM フィールドを使用して REXX 名および始動パラメーターを渡すものと想定されています。これは、SYSTSIN が CARMA で安全に使用できることを意味します。IRXJCL によって SYSTSPRT へ送られたすべての出力は、CARMA のログに書き込まれます。

CRAXJCL 置換モジュールを呼び出すプロセッサは、CRAXJCL を呼び出す前に DD SYSTSIN または SYSTSPRT の割り振りを試みてはなりません。

CRAXJCL の作成

CRAXJCL 置換モジュールはソース形式で提供されています。これは、お客様がこのモジュールをカスタマイズして、SYSTSPRT に使用する固有の割り振りを指定する必要があるためです。SYSTSIN の割り振りは通常、ダミー・データ・セットに対して行う必要があります。

サンプルのアセンブラー・ソース・コードおよびサンプルのコンパイル/バインド・ジョブは、FEL.#CUST.ASM(CRAXJCL) および FEL.#CUST.JCL(CRA#CIRX) として使用可能です。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

メンバー内のドキュメンテーションを使用し、必要に応じて CRAXJCL アセンブラー・ソース・コードをカスタマイズしてください。後で、CRA#CIRX JCL をカスタマイズおよび実行依頼して、CRAXJCL ロード・モジュールを作成します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。

必要に応じ、IRXJCL を別の名前に変更できます。IRXJCL の新規名を呼び出してコンパイルするように CRAXJCL ソースを調整してから、CRAXJCL ロード・モジュールの名前を IRXJCL に変更します。このセットアップは、IRXJCL へのすべての呼び出しを変更するよりも容易である可能性があります。

第 4 章 SCLM Developer Toolkit

SCLM Developer Toolkit は、SCLM の機能を拡張するために必要なツールをクライアントに提供します。SCLM 自体はホスト・システム・ベースのソース・コード・マネージャーであり、ISPF の一部として組み込まれています。

SCLM Developer Toolkit は、SCLM へのインターフェースとして機能する Eclipse ベースのプラグインを備えています。これにより、従来型コード開発におけるすべての SCLM プロセスへのアクセスを可能にし、メインフレーム上の SCLM と同期したワークステーション上で完全な Java および Java EE 開発 (メインフレームからの Java EE コードのビルド、アセンブル、およびデプロイメントを含む) を行えるようにサポートします。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、SCLM 管理者、およびオプションとしてセキュリティ管理者の支援が必要になります。このタスクには、以下のリソースか特殊なカスタマイズ・タスクが必要です。

- APF および LINKLIST の更新
- Java EE サポート用の SCLM 言語変換プログラムを定義する
- Java EE サポート用の SCLM タイプを定義する
- (オプション) ユーザーに SCLM VSAM の更新を許可するセキュリティ規則
- (オプション) Ant のインストール

ご使用のサイトで SCLM Developer Toolkit の使用を開始するには、以下のタスクを行う必要があります。特に断りがない限り、すべてのタスクは必須です。

1. 前提条件と PARMLIB 更新を確認および調整します。詳細については、76 ページの『前提条件』を参照してください。
2. Developer for z Systems 構成ファイルをカスタマイズします。詳細については、以下を参照してください。
 - 76 ページの『SCLMDT 用の ISPF.conf の更新』
 - 77 ページの『SCLMDT 用の rdz.env の更新』
3. オプションとして、ロング/ショート・ネーム変換サポートを定義します。詳細については、78 ページの『(オプション) ロング/ショート・ネーム変換』を参照してください。
4. オプションとして、Java EE ビルド・サポートを使用するために Ant をインストールおよびカスタマイズします。詳細については、80 ページの『(オプション) Ant のインストールおよびカスタマイズ』を参照してください。
5. SCLM を更新して、SCLMDT 固有の部分を定義します。詳細については、81 ページの『SCLMDT 用の SCLM の更新』を参照してください。
6. オプションとして、SCLMDT 作業域の定期的な自動クリーンアップをセットアップします。詳細については、82 ページの『WORKAREA および /tmp からの古いファイルの除去』を参照してください。

前提条件

Developer for z Systems のソフトウェア要件 (前提条件および相互前提条件を含む) の完全なリストについては、Developer for z Systems 製品ページ (<http://www-03.ibm.com/software/products/en/developerforsystemz/>) の「**System Requirements**」タブを参照してください。ここには、SCLM Developer Toolkit で Java EE ビルドに必要な Ant 仕様についても説明されています。

重要: SCLM Developer Toolkit では、レガシー ISPF ゲートウェイを使用します。これは、z/OS 1.8 以降が必要であることを意味します。さらに、TSO コマンド・サービスに対話式 ISPF ゲートウェイを使用する場合は、レガシー ISPF ゲートウェイと対話式 ISPF ゲートウェイの両方を構成する必要があることも意味しています。

19 ページの『PARMLIB の変更』の説明にあるように、SCLM Developer Toolkit を使用するには、システム設定の追加のカスタマイズが必要です。これには以下の項目が含まれます。

- (BPXPRMxx) z/OS UNIX ユーザー ID の 1 つ当たりの最大プロセス数を増やします。
- (PROGxx) SYS1.LINKLIB および REXX ランタイム REXX.V1R4M0.SEAGLPA または REXX.V1R4M0.SEAGALT の APF 許可を与えます。
- (PROGxx/LPALSTxx) ISP.SISPLPA、ISP.SISPLoad、SYS1.LINKLIB、および REXX ランタイムを LINKLIST/LPALIB に入れます。

また、SCLM Developer Toolkit は SDSF または TSO **OUTPUT** コマンドを使用して、ジョブの完了状況とジョブの出力を取り出します。どちらの方法でも、以下の点にさらに注意が必要です。

- SDSF は、別途オーダーし、インストールし、構成する必要があります。
- TSO **OUTPUT** コマンドのデフォルトの設定では、各ユーザーは、自分の特定のユーザー ID で始まるジョブ出力だけを取り出すことができます。**OUTPUT** 機能に十分に使用するには、ユーザーが自分の所有するジョブ出力を、それが自分のユーザー ID で始まるものでなくても取り出すことができるように、サンプルの TSO/E 出口 IKJEFF53 を変更しなければならない場合があります。この出口について詳しくは、「TSO/E カスタマイズ」(SA88-8629) を参照してください。

ユーザーには、z/OS UNIX ディレクトリー /tmp/ および /var/zexpl/WORKAREA/ に対する READ、WRITE、および EXECUTE 権限が必要です。ディレクトリー WORKAREA/ は /var/zexpl/ にあります。ただし、IBM Explorer for z/OS のカスタマイズで異なる場所を指定した場合を除きます。

SCLMDT 用の ISPF.conf の更新

SCLM Developer Toolkit は標準の ISPF/SCLM スケルトンを使用するため、ISP.SISPSLIB スケルトン・ライブラリーが ISPF.conf 内の ISPSLIB 連結に割り振られるようにしてください。ISP.SISPSENU データ・セットの使用はオプションです。

ISPF.conf は /etc/zexpl/ にあります。ただし、IBM Explorer for z/OS のカスタマイズで異なる場所を指定した場合を除きます。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更は、更新後にホスト・システムに接続するすべてのクライアントについて有効になります。

次のサンプル・コードは ISPF.conf ファイルを示しています。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。コメント行は、アスタリスク (*) で始まります。同じ行にある連結にデータ・セットを追加し、名前同士をコンマ (,) で分離します。.

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

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

図 20. SCLMDT 用の ISPF.conf の更新

注:

- TSO 環境をカスタマイズするために、独自の DD のようなステートメントとデータ・セット連結を追加し、TSO ログオン・プロシーチャーを模倣することができます。詳しくは、「*IBM Explorer for z/OS Host Configuration Reference (SC27-8438)*」の『TSO 環境のカスタマイズ』を参照してください。
- バッチ・ビルドを行う場合は、ISPF/SCLM スケルトン・ライブラリーの前に、カスタマイズされたバージョンの FLMLIBS スケルトンが連結されるようにしてください。

```
ispslib=h1q.USERSKEL,ISP.SISPSLIB
```

SCLMDT 用の rdz.env の更新

SCLM Developer Toolkit は、rdz.env の中で設定されたいくつかのディレクティブを使用して、データ・セットおよびディレクトリーを見つけます。

rdz.env は /etc/zexpl/ に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更を有効にするには、IBM Explorer for z/OS RSED 開始タスクを再始動してください。

次のコード・サンプルは、rdz.env ファイル内の SCLMDT ディレクティブを示しています。このファイルは、使用するシステム環境に合わせてカスタマイズする必要があります。rdz.env のカスタマイズについて詳しくは、30 ページの『rdz.env、

環境構成ファイル』を参照してください。

```
_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
```

図 21. SCLMDT 用の *rdz.env* の更新

(オプション) ロング/ショート・ネーム変換

SCLM Developer Toolkit では、ロング・ネーム・ファイルを SCLM に保管することができます。ロング・ネーム・ファイルとは、8 文字を超える名前または大/小文字混合の名前を持つファイルのことです。ロング・ファイル・ネームの保管は、ロング・ファイル・ネーム名を、SCLM で使用する 8 文字のメンバー名に対応付けるマッピングが入った VSAM ファイルを使用することで実現しています。

注:

- z/OS 1.8 より前のバージョンの場合、この機能は APAR OA11426 に対応する基本 ISPF/SCLM PTF によって提供されます。
- ロング/ショート・ネーム変換は、IBM SCLM Administrator Toolkit などの、他の SCLM 関連製品でも使用されます。

LSTRANS.FILE (ロング/ショート・ネーム変換 VSAM) の作成

ロング/ショート・ネーム変換 VSAM を作成するには、ISP.SISPSAMP ISPF サンプル・ライブラリー内のサンプル・メンバー FLM02LST をカスタマイズして実行依頼します。本資料の構成ステップでは、以下のサンプルのセットアップ JCL に示すように、この VSAM に FEL.#CUST.LSTRANS.FILE という名前を付ける必要があります。

```

//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))
/*

```

図 22. FLM02LST: ロング/ショート・ネーム変換セットアップ JCL

注: ユーザーには、この VSAM データ・セットに対する UPDATE 権限が必要です。

rdz.env ロング/ショート・ネーム変換用に更新

ロング/ショート・ネーム変換を使用する前に、rdz.env 環境変数 `_SCLMDT_TRANTABLE` をコメント解除し、ロング/ショート・ネーム変換 VSAM の名前に一致するように設定します。

rdz.env は /etc/zexpl/ に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO **OEDIT** コマンドで編集できます。

注: 変更を有効にするには、RSED 開始タスクを再始動してください。

(オプション) Ant のインストールおよびカスタマイズ

このステップは、SCLM で Java EE ビルド・サポートを使用する場合にのみ、必要となります。

Apache Ant はオープン・ソースの Java ビルド・ツールであり、<http://ant.apache.org/> からダウンロードできます。Ant はテキスト・ファイルとスクリプトからなり、それらは ASCII フォーマットで配布されています。このため、z/OS UNIX で実行するためには、ASCII/EBCDIC 変換が必要です。

以下のステップを実行して、Ant を z/OS に実装し、Developer for z Systems に対して定義します。

- 最新の Ant 圧縮ファイルをバイナリー・フォーマットで z/OS UNIX ファイル・システムにダウンロードします。.zip 版の ANT をダウンロードします。
- z/OS UNIX コマンド行セッションを開き、TSO **OMVS** コマンドを使用するなどして、インストールを続行します。
- **mkdir -p /home-dir** コマンドを使用して Ant インストールのホーム・ディレクトリーを作成し、それを **cd /home-dir** コマンドで現行ディレクトリーにします。
- JAR 抽出コマンド **jar -xf apache-ant-1.7.1.zip** を使用して、ファイルを現行ディレクトリーに抽出してください。jar コマンドを使用するには、Java bin ディレクトリーがローカルの z/OS UNIX パスに存在していなければなりません。そうならない場合は、コマンドを Java bin ロケーションで完全修飾 (例えば、/usr/lpp/java/J6.0/bin/jar -xf apache-ant-1.7.1.zip) してください。
- すべての Ant テキスト・ファイルを EBCDIC に変換するため、オプションとして /usr/lpp/ibm/rdz/samples/BWBTRANT サンプル・スクリプトをカスタマイズして実行します。

注: このスクリプトは 1 回だけ実行してください。複数回実行すると、Ant インストールが壊れます。

- 変換が成功したかどうかを検査するために、ANT ディレクトリー内のテキスト・ファイル、例えば apache-ant-1.7.1/README などを見つけて開きます。ファイルが読み取り可能であれば、変換は正常に行われています。
- **chmod -R 755 *** コマンドを使用して、すべてのユーザーが ANT ディレクトリー内のファイルの読み取りと実行を行うことができるようにします。

- Ant を使用する前に、環境変数 `JAVA_HOME` および `ANT_HOME` を設定します。
 - (rse.env 内の) `JAVA_HOME` は Java ホーム・ディレクトリーを指す必要があります。例:


```
JAVA_HOME=/usr/lpp/java/IBM/J6.0
```
 - (rse.env 内の) `ANT_HOME` は Ant ホーム・ディレクトリーを指す必要があります。例:


```
ANT_HOME=/usr/lpp/Apache/Ant/apache-ant-1.7.1
```

次に例を示します。

- `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`

Ant の初期化が正常に行われたことをテストするには、次のようにします。

- Ant および Java bin ディレクトリーを環境変数 `PATH` に追加します。

例:

```
export PATH=/usr/lpp/Apache/Ant/apache-ant-1.7.1/bin:$PATH
export PATH=/usr/lpp/java/J6.0/bin:$PATH
```

- 正常にインストールが終了したら、バージョンを表示するために、`ant -version` を実行します。

例:

```
ant -version
```

注: この方法による `PATH` ステートメントの設定はテスト時に必要なものであって、実際の運用に際しては不要です。

SCLMDT 用の SCLM の更新

SCLM 自体も、SCLM Developer Toolkit を処理するためにはカスタマイズが必要です。以下の必要なカスタマイズ・タスクの詳細については、「*IBM Rational Developer for z Systems SCLM Developer Toolkit 管理者ガイド*」(SC88-5664) を参照してください。

- Java EE サポート用の言語変換プログラムを定義する
- Java EE サポート用の SCLM タイプを定義する

カスタマイズ・タスクおよびプロジェクト定義タスクを完了するためには、SCLM 管理者は、82 ページの表 13 に示す Developer for z Systems のカスタマイズ可能値を知っている必要があります。

表 13. SCLM 管理者チェックリスト

説明	<ul style="list-style-type: none"> デフォルト値 正解の入手先 	値
Developer for z Systems サンプル・ライブラリー	<ul style="list-style-type: none"> FEL.SFELSAMV SMP/E インストール 	
Developer for z Systems サンプル・ディレクトリー	<ul style="list-style-type: none"> /usr/lpp/ibm/rdz/samples SMP/E インストール 	
Java bin ディレクトリー	<ul style="list-style-type: none"> /usr/lpp/java/J6.0/bin rse.env - \$JAVA_HOME/bin 	
Ant bin ディレクトリー	<ul style="list-style-type: none"> /usr/lpp/Apache/Ant/apache-ant-1.7.1/bin rdz.env - \$ANT_HOME/bin 	
WORKAREA ホーム・ディレクトリー	<ul style="list-style-type: none"> /var/zexpl rse.env - \$CGI_ISPWORK 	
SCLMDT プロジェクト構成ホーム・ディレクトリー	<ul style="list-style-type: none"> /var/rdz/sclmdt rdz.env - \$_SCLMDT_CONF_HOME 	
ロング/ショート・ネーム変換 VSAM	<ul style="list-style-type: none"> FEL.#CUST.LSTRANS.FILE rdz.env - \$_SCLMDT_TRANTABLE 	

WORKAREA および /tmp からの古いファイルの除去

SCLM Developer Toolkit とレガシー ISPF ゲートウェイは同じ WORKAREA および /tmp ディレクトリーを共用します。これらは両方とも定期的なクリーンアップを必要とする場合があります。z/OS UNIX には、ファイルが入っているディレクトリーとファイルの経過日数に基づいてファイルを削除する、skulker というシェール・スクリプトがあります。指定された日時にコマンドを実行する z/OS UNIX cron デーモンと結合すれば、定期的にターゲット・ディレクトリーを空にする自動化ツールをセットアップできます。skulker スクリプトおよび cron デーモンの詳細については、「UNIX System Services コマンド解説書」(SA88-8641) を参照してください。

第 5 章 統合デバッガー

Developer for z Systems 統合デバッガー・ホスト・コンポーネントを使用すると、バージョン 9.0.1 以上のクライアントは、言語環境プログラム (Language Environment (LE)) ベースのさまざまなアプリケーションをデバッグできます。統合デバッガーには、z/OS 1.10 以上が必要です。

統合デバッガーのデータ・フローの概要については、「ホスト構成リファレンス (SC43-2912)」の『Developer for z Systems について』の章の『統合デバッガー』のセクションを参照してください。

要件およびチェックリスト

このカスタマイズ・タスクを完了するには、セキュリティー、TCP/IP、および CICS の管理者の支援が必要になります。このタスクでは、以下のリソースまたは特殊なカスタマイズ・タスクが必要となります。

- SVC の追加 (IPL はオプション)
- SVC 用に更新された LPA
- LINKLIST 更新
- APF 許可
- 開始タスクの定義
- セキュリティー・プロファイルおよびアクセス・リストの定義
- クライアント/ホスト通信およびホスト限定通信のための TCP/IP ポートの予約
- (オプション) CICS 領域 JCL の更新
- (オプション) CICS CSD の更新
- (オプション) IMS メッセージ領域 JCL の更新
- (オプション) DB2 ストアード・プロシージャのアドレス・スペース JCL の更新
- (オプション) ELAXF* リモート・ビルド・プロシージャの更新

サイトで統合デバッガーの使用を開始するには、以下のタスクを実行する必要があります。特に断りがない限り、すべてのタスクは必須です。

1. 統合デバッガーが機能するには、(IBM Explorer for z/OS の必須の RSED 開始タスクと共に) オプションの DBGMGR 開始タスクがアクティブになっている必要があります。DBGMGR 始動 JCL については、26 ページの『DBGMGR、デバッグ・マネージャー開始タスク』を参照してください。
2. 統合デバッガー構成は、DBGMGR 開始タスクの始動引数で管理されます。詳しくは、86 ページの『統合デバッガー構成パラメーター』を参照してください。
3. DBGMGR 開始タスクを実行するには、FEL.SFELAUTH ライブラリーに APF 許可がある必要があります。86 ページの『統合デバッガーの parmlib 更新』を参照してください。

4. 統合デバッガーは、Developer for z Systems 監視プログラム呼び出し (SVC) に依存しています。SVC ロード・モジュールは、リンク・バック域 (LPA) にロードされる必要があります。そうすると動的にアクティブ化できます。詳しくは、86 ページの『統合デバッガーの parmlib 更新』を参照してください。
5. 統合デバッガーには、アプリケーションからアクセスできる必要があります、STEPLIB 更新または LINKLIST 更新が必要です。詳しくは、86 ページの『統合デバッガーの parmlib 更新』を参照してください。
6. 統合デバッガーが機能するには、デバッグするアプリケーションのユーザー ID が有効な OMVS セグメントを持っている必要があります。詳しくは、88 ページの『統合デバッガーのセキュリティ更新』を参照してください。
7. 統合デバッガーは、いくつかのセキュリティ許可を必要とします。詳しくは、88 ページの『統合デバッガーのセキュリティ更新』を参照してください。
8. DBGMGR 開始タスクは、いくつかのセキュリティ許可を必要とします。詳しくは、88 ページの『統合デバッガーのセキュリティ更新』を参照してください。

CICS トランザクションをデバッグするために必要なステップは、以下のとおりです。

1. 統合デバッガーを使用すると、CICS トランザクションをデバッグできます。そのためには、統合デバッガーと IP CICS ソケットを CICS に定義する必要があります。詳しくは、90 ページの『統合デバッガーの CICS アップデート』を参照してください。

統合デバッガーは、特定のモジュールのデータ・セットから TEST ランタイム・オプションを読み取る、言語環境プログラム (Language Environment (LE)) のユーザー出口を使用できます。これは、IMS トランザクションや DB2 ストアード・プロシージャなど、サブシステムでアクティブになっているコードのデバッグで効果的です。このコードに動的に TEST ランタイム・オプションを提供することはできないためです。以下のステップは、LE ユーザー出口を使用している場合にのみ必要です。

1. LE ユーザー出口と、そのユーザー出口を呼び出すカスタマイズ済み LE 環境を作成します。詳しくは、89 ページの『統合デバッガーの言語環境プログラム (Language Environment) の更新』を参照してください。
2. カスタマイズされた LE 環境を IMS トランザクションに定義します。詳しくは、91 ページの『統合デバッガーの IMS の更新』を参照してください。
3. カスタマイズされた LE 環境を DB2 ストアード・プロシージャに定義します。詳しくは、91 ページの『統合デバッガーの DB2 ストアード・プロシージャの更新』を参照してください。

注:

- 統合デバッガーは、CICS CADP を使用して、CICS トランザクションに対する TEST ランタイム・オプションを提供します。CADP について詳しくは、CICS TS の資料を参照してください。ただし、必要に応じて、LE ユーザー出口メカニズムを使用することもできます。
- LE ユーザー出口メカニズムは、標準アプリケーションで使用することもできます。詳細については、92 ページの『統合デバッガーの ELAXF* の更新』を参照してください。

統合デバッガーと COBOL v4

COBOL v3.4 および v4 で作成されたプログラムをデバッグするために、統合デバッガーには、リストされたデータ・セットへのアクセス権限が必要です。データ・セット名は、環境変数 AQE_DBG_V4LIST、または DD AQEV4LST を介して指定できます。どちらも存在しない場合、統合デバッガーは、実行可能ファイルのデータ・セットの最後の修飾子 (例、.LOAD) を .LISTING に置換することによって、データ・セット名を作成します。ご使用のサイトでどちらの方法が使用可能であるかを調べるには、開発者にご相談ください。

統合デバッガーおよびその他の言語環境ベースのデバッガー

通常、所定のアプリケーション、CICS 領域、DB2 ストアード・プロシージャ、または IMS トランザクションでアクティブにできる言語環境 (LE) ベースのデバッガー (統合デバッガーなど) は、1 つだけです。LE ベースのデバッガーであることを示す分かりやすい指標は、アプリケーションで使用されるべき CEEEVDBG ロード・モジュールまたは別名をそのデバッガーが提供することです。

ただし、統合デバッガーは、アプリケーションによって統合デバッガーが最初にロードされていれば、IBM Debug Tool for z/OS と共存させることができます。

統合デバッガーと暗号化通信

Developer for z Systems クライアントで暗号化を使用して RSE デーモンと通信している場合、(クライアント・ベースの) デバッグ・エンジンも、デフォルトで、(ホスト・ベースの) デバッグ・マネージャーと通信する際に暗号化を使用します。デフォルトでは、デバッグ・エンジンは Developer for z Systems クライアントと同じ証明書を使用します。

これは、RSE デーモンの開始タスクと、デバッグ・マネージャーの開始タスクが、通信の暗号化に関して同様にセットアップされていると想定されるということです。以下の代替シナリオは、RSE デーモンとデバッグ・マネージャーの暗号化設定が異なる状況において、有効です。

- 暗号化セッションのセットアップが失敗した場合、デバッグ・エンジンは、確認後に非暗号化通信の使用を試みることができます。
- デバッグ・エンジンは、Developer for z Systems クライアント以外の認証証明書を使用できます。

RSE デーモンとは異なり、デバッグ・マネージャーでは暗号化通信はネイティブ・サポートされていません。デバッグ・マネージャーは、暗号化通信について、Application Transparent Transport Layer Security (AT-TLS) と呼ばれる TCP/IP サービスに依存しています。段階的なセットアップ手順の説明については、「*IBM Rational Developer for z Systems* ホスト構成リファレンス (SC43-2912)」の『AT-TLS のセットアップ』を参照してください。

統合デバグガーと SYSPLEX

Developer for z Systems は SYSPLEX に認識されないため、Developer for z Systems クライアントの接続先と同じシステム上で、デバグ対象のプログラムがアクティブである必要があります。これには、バッチ・ジョブを介して開始されたデバグ・セッションが含まれます。

z/OS 2.1 では、ジョブ・カードに SYSAFF=* または SYSTEM=* を指定して、実行依頼されたシステム上でジョブが実行されるようにします。古いシステムでは、JESPLEX を使用して SYSPLEX 内の複数の JES サブシステムを結合する場合、正しいシステム名を明示的に指定する必要があります。

統合デバグガー構成パラメーター

統合デバグガーでは、DBGMR 始動 JCL で以下の変数を構成できます。DBGMR 始動 JCL については、26 ページの『DBGMR、デバグ・マネージャー開始タスク』を参照してください。

- タイムゾーン・オフセット (デフォルトは EST5DST)
- 外部 (クライアント/ホスト) 通信に使用されるポート番号 (デフォルトは 5335)
- 内部 (ホスト限定) 通信に使用されるポート (デフォルトは 5336)
- 統合デバグガーに使用される SVC 番号 (デフォルトは 251)
- ロード・ライブラリーの高位修飾子 (デフォルトは FEL)

統合デバグガーの parmlib 更新

- DBGMR 開始タスクは、Developer for z Systems クライアントがホストに接続される前にアクティブになっている必要があります。IPL 時に自動的にサーバーを開始する方法についての詳細は、20 ページの『COMMNDxx への開始タスクの追加』を参照してください。
- DBGMR 開始タスクを実行するには、FEL.SFELAUTH ライブラリーに APF 許可がある必要があります。詳しくは、22 ページの『PROGxx での APF 許可』を参照してください。
- 言語環境プログラム (LE) は、統合デバグガーを起動できる必要があります。そのため、FEL.SFELAUTH ライブラリーを LINKLIST に配置するか、またはデバグするアプリケーションの STEPLIB に配置する必要があります。詳しくは、23 ページの『PROGxx での LINKLIST 定義』を参照してください。

注:

- LINKLIST を使用するときには、CEEVDBG ロード・モジュールを保持する他の LE ベース・デバグガーのライブラリーの前に FEL.SFELAUTH があることを確認してください。例えば、IBM Debug Tool for z/OS は hlq.SEQA* ライブラリーを使用します。
- 競合を回避するには、LINKLIST で定義される LE ベースのデバグガーは 1 つでなければなりません。
- 統合デバグガーは、アプリケーションによって統合デバグガーが最初にロードされていれば、IBM Debug Tool for z/OS と共存させることができます。

– FEL.SFELAUTH 内の統合デバッガー・ロード・モジュールは、デバッグ・マネージャー開始タスクを除いて、許可済みで実行する必要はありません。ここには、ロード・モジュールが常駐しているので、それらを許可済みの環境で使用できます。

- 統合デバッガーは z/OS Binder を使用します。これは、SYS1.MIGLIB が LINKLIST (または STEPLIB) になければならないことを意味します。詳しくは、24 ページの『必要な LINKLIST 定義と LPA 定義』を参照してください。
- 統合デバッガーは z/OS Binder API を使用します。この API は z/OS 1.10 以降で /usr/lib/iewbndd.so として使用でき、z/OS 1.13 以降でも SYS1.SIEAMIGE(IEWBNDD) として使用できます。これは、z/OS 1.13 以上では、SYS1.SIEAMIGE が LINKLIST (または STEPLIB) 内にある必要があることを示しています。詳しくは、24 ページの『必要な LINKLIST 定義と LPA 定義』を参照してください。

注: z/OS 1.13 以上のシステムで SYS1.SIEAMIGE が LINKLIST または STEPLIB に含まれていない場合、統合デバッガーは以下のメッセージを発行し、/usr/lib/iewbndd.so の使用を試みます。

```
CEE3501S The module //IEWBNDD was not found
```

- 統合デバッガーは、システムに Developer for z Systems 監視プログラム呼び出し (SVC) が定義されていることを必要とします。デフォルトの SVC 番号は 251 です。関連するロード・モジュール FEL.SFELLPA(AQESVC03) が、LPA にロードされて動的にアクティブ化される必要があります。詳しくは、20 ページの『IEASVCxx での SVC 定義』、および 21 ページの『LPALSTxx での LPA 定義』を参照してください。

注: バージョン 9.1.1 より前のバージョンの Developer for z Systems では、現在使用されていない別の SVC、AQESVC01 を使用していました。これは、ご使用のホスト・システムにバージョン 9.1.1 以上の Developer for z Systems しかない場合には、削除できます。

統合デバッガーの SVC 更新

統合デバッガーは、システムに Developer for z Systems 監視プログラム呼び出し (SVC)、FEL.SFELLPA(AQESVC03) が定義されていることを必要とします。理想的な環境では、システムの IPL 時に、この SVC がアクティブ化されるか、または更新されます。ただし、IPL を常にオプションとして選択できるわけではないので、統合デバッガーがデバッグ・マネージャー開始タスクを使用して動的インストールまたは更新を行います。

始動時に、デバッグ・マネージャー開始タスク DBGMGR は、以下の項目を検証してから、88 ページの表 14に記載されているように関連アクションを実行します。

- SVC が定義されているかどうか
- アクティブ SVC (定義されている場合) および新規 SVC (SFELLPA 内) のバージョン
- SVC=svc_number 始動引数が指定されているかどうか (svc_number は必要な SVC 番号)

表 14. 動的統合デバッガー SVC 更新

	SVC 定義済み	SVC 未定義
アクティブ SVC と新規 SVC のバージョンが一致します	<ul style="list-style-type: none"> アクションなし。SVC はすでにアクティブで最新です SVC 始動引数は無視されます 	<ul style="list-style-type: none"> SFELLPA ロード・モジュールを使用して動的に SVC を定義します SVC 始動引数は必須です
アクティブ SVC と新規 SVC のバージョンが一致しません	<ul style="list-style-type: none"> SFELLPA ロード・モジュールを使用して SVC を更新します SVC 始動引数は無視されます 	<ul style="list-style-type: none"> SFELLPA ロード・モジュールを使用して動的に SVC を定義します SVC 始動引数は必須です

統合デバッガーの TCP/IP 更新

統合デバッガーは 2 つの TCP/IP ポートを使用します。

- クライアント/ホスト通信用のポート (デフォルトは 5335)。このポートでの通信を暗号化することができます。
- ホスト限定の通信用のポート (デフォルトは 5336)。

統合デバッガーのセキュリティー更新

統合デバッガーは以下のセキュリティー定義を必要とします。詳しくは、109 ページの『第 9 章 セキュリティー定義』を参照してください。

- デバッグ対象のアプリケーションを実行しているユーザー ID の OMVS セグメント (これには、CICS 領域のユーザー ID が含まれます)
- DBGMR 開始タスク
- 開始タスク・ユーザー ID の BPX.SERVER 許可
- 開始タスク・ロード・ライブラリーのプログラム制御
- 問題プログラム状態をデバッグしているユーザーの AQE.AUTHDEBUG.STDPGM 許可
- 問題プログラム状態の許可済みプログラムをデバッグしているユーザーの AQE.AUTHDEBUG.AUTHPGM 許可

注: 統合デバッガーがない状態の、既存の Developer for z Systems セットアップからのマイグレーションを単純化するために、RACF コマンドを伴うサンプル JCL FEL.SFELSAMP(AQERACF) が提供されています。これは統合デバッガーに関連するセキュリティー定義だけを定義します。

統合デバッガーの言語環境プログラム (Language Environment) の更新

統合デバッガーは、特定のモジュールのデータ・セットから TEST ランタイム・オプションを読み取る、言語環境プログラム (Language Environment (LE)) のユーザー出口を使用できます。これは、IMS トランザクションや DB2 ストアード・プロシージャなど、サブシステムでアクティブになっているコードのデバッグで効果的です。このコードに動的に TEST ランタイム・オプションを提供することはできないためです。

このプロセス・フローについて、次のステップで説明します。

1. LE がモジュールの実行を依頼されます。
2. LE は、引数としてモジュール名を指定してユーザー出口を呼び出します。
3. ユーザー出口は、TEST ランタイム・オプション・データ・セットを開き、モジュールの TEST ランタイム・オプションを見つけます。
4. ユーザー出口は、LE に TEST ランタイム・オプションを返します。
5. LE は、指定された TEST ランタイム・オプションを使ってモジュールを実行します。

注:

- IBM Debug Tool for z/OS は、ここで説明するユーザー出口メカニズムの拡張バージョンを使用します。共存をサポートし、TEST ランタイム・オプションを保守する労力が重複することを回避するため、統合デバッガーは、ユーザー出口オプションおよび TEST ランタイム・オプションのデータ・セット名について、デバッグ・ツールと同じデフォルトを使用します。
- ユーザー出口では、TEST ランタイム・オプションのデータ・セット名で、ユーザー ID およびモジュール名に対する変数置換をサポートしています。
- Developer for z Systems クライアント GUI を使用して TEST ランタイム・オプションのデータ・セットを保守するには、Developer for z Systems 以上のホストおよびクライアントが必要です。
- 統合デバッガーは、CICS CADP を使用して、CICS トランザクションに対する TEST ランタイム・オプションを提供します。CADP について詳しくは、CICS TS の資料を参照してください。

以下のステップは、LE ユーザー出口を使用している場合にのみ必要です。

- AQED3CXT をカスタマイズおよび実行依頼して、ユーザー出口を作成します。AQED3CXT は FEL.#CUST.JCL にあります。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。
- AQED3CEE をカスタマイズおよび実行依頼して、ユーザー出口を呼び出すカスタマイズ済み LE ロード・モジュールを作成します。AQED3CEE は FEL.#CUST.JCL にあります。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。デフォルトでは、カスタマイズした LE ロード・モジュール (CEE*) は FEL.#CUST.LOAD に置かれています。

注: パフォーマンス上の理由から、カスタマイズした LE ロード・モジュールが入っているロード・ライブラリーを LINKLIST には配置しないでください。購入したソフトウェアも含めて、すべての LE 対応ロード・モジュールに対してユーザー出口が呼び出されることになります。

統合デバッガーの CICS アップデート

統合デバッガーで CICS トランザクションをデバッグするには、以下の CICS アップデートが必要です。

- CICS システム初期設定 (SIT) パラメーターの更新:
 - DEBUGTOOL=YES を指定します。
 - TCPIP=YES を指定します。
 - LINKLIST によって、DFHRPL DD 連結からロード・モジュールを取り出す場合、LLACOPY=YES を指定します。
- CICS JCL 更新:
 - EXEC ステートメントで REGION=0M を指定します。
 - 領域の DFHRPL DD ステートメントで FEL.SFELAUTH ロード・ライブラリーを定義します。SIT パラメーター LLACOPY=YES が指定されている場合、ライブラリーは LINKLIST に置くこともできます。
 - 領域の DFHRPL DD ステートメントで SYS1.MIGLIB ロード・ライブラリーを定義します。SIT パラメーター LLACOPY=YES が指定されている場合、ライブラリーは LINKLIST に置くこともできます。
 - z/OS 1.13 以上の場合、領域の DFHRPL DD ステートメントで SYS1.SIEAMIGE ロード・ライブラリーを定義します。SIT パラメーター LLACOPY=YES が指定されている場合、ライブラリーは LINKLIST に置くこともできます。詳細については、86 ページの『統合デバッガーの parmlib 更新』にある z/OS Binder API の情報を参照してください。

注:

- CICS 領域のユーザー ID には、SIT パラメーター LLACOPY=YES が意図したとおりに機能するために、FACILITY クラスの CSVLLA.dataset プロファイルに対する UPDATE 許可が必要です。
- COBOL v4 で作成されたプログラムをデバッグするために、統合デバッガーには、リストされているデータ・セット (PDS または PDS/E) へのアクセス権限が必要です。データ・セット名は、環境変数 AQE_DBG_V4LIST、または DD AQEV4LST を介して指定できます。どちらも存在しない場合、統合デバッガーは、実行可能ファイルのデータ・セットの最後の修飾子 (例、.LOAD) を .LISTING に置換することによって、データ・セット名を作成します。ご使用のサイトでどちらの方法が使用可能であるか、開発者に問い合わせてください。
- CICS CSD アップデート:

AQECSD サンプル CSD 更新ジョブの説明に従って、デバッガーを CICS 領域に定義します。AQECSD は FEL.#CUST.JCL にあります。ただし、ジョブ

FEL.SFELSAMP(FELSETUP) をカスタマイズして実行依頼したときに別のロケーションを指定した場合を除きます。詳細については、17 ページの『カスタマイズのセットアップ』を参照してください。

注:

- 統合デバッガーでは、ご使用の CICS 領域で TCP/IP ベースの通信が有効になっている必要があります。これを行う場合に推奨される方法は、z/OS Communication Server IP CICS ソケットです。このフィーチャーの構成については、「*Communication Server IP CICS ソケット・ガイド (SC88-9053)*」を参照してください。
- 統合デバッガーは、CICS CADP を使用して、CICS トランザクションに対する TEST ランタイム・オプションを提供します。CADP について詳しくは、CICSTS の資料を参照してください。

統合デバッガーの IMS の更新

統合デバッガーで IMS トランザクションをデバッグするには、以下の IMS の更新が必要です。

- IMS メッセージ領域 (DFSMPR) JCL の更新:
 - 領域の EXEC ステートメントで REGION=0M を指定します。
 - FEL.SFELAUTH ロード・ライブラリーが LINKLIST に存在しない場合、このライブラリーを STEPLIB に追加します。統合デバッガー以外の他の製品で必要とされている場合を除いて、すべての STEPLIST データ・セットで APF 許可を保守する必要はありません。
 - 言語環境プログラム (Language Environment (LE)) ユーザー出口を使用している場合、カスタマイズされた LE ロード・モジュールが入っているロード・ライブラリー (デフォルトでは FEL.#CUST.LOAD) を STEPLIB に追加します。

統合デバッガーの DB2 ストアード・プロシージャの更新

統合デバッガーで DB2 ストアード・プロシージャをデバッグするには、以下の DB2 関連の更新を行う必要があります。

- DB2 ストアード・プロシージャのアドレス・スペースの開始タスク JCL に対する更新
 - EXEC ステートメントで REGION=0M を指定します。
 - FEL.SFELAUTH ロード・ライブラリーが LINKLIST に存在しない場合、このライブラリーを STEPLIB に追加します。統合デバッガー以外の他の製品で必要とされている場合を除いて、すべての STEPLIST データ・セットで APF 許可を保守する必要はありません。
 - 言語環境プログラム (Language Environment (LE)) ユーザー出口を使用している場合、カスタマイズされた LE ロード・モジュールが入っているロード・ライブラリー (デフォルトでは FEL.#CUST.LOAD) を STEPLIB に追加します。

統合デバッガーの ELAXF* の更新

以下の更新は、Developer for z Systems の提供するリモート・ビルド・プロシージャで統合デバッガーを使用するために必要です。これらの JCL プロシージャの機能および場所について詳しくは、27 ページの『ELAXF* リモート・ビルド・プロシージャ』を参照してください。

- 以下のようにして、ELAXFDCL、ELAXFG0、および ELAXFTS0 JCL プロシージャに更新します。
 - EXEC ステートメントで REGION=0M を指定します。
 - FEL.SFELAUTH ロード・ライブラリーが LINKLIST に存在しない場合、このライブラリーを STEPLIB に追加します。統合デバッガー以外の他の製品で必要とされている場合を除いて、すべての STEPLIST データ・セットで APF 許可を保守する必要はありません。
 - 言語環境プログラム (Language Environment (LE)) ユーザー出口を使用している場合、カスタマイズされた LE ロード・モジュールが入っているロード・ライブラリー (デフォルトでは FEL.#CUST.LOAD) を STEPLIB に追加します。

第 6 章 ホスト・ベースのコード分析

Developer for z Systems クライアントと同様に、Developer for z Systems ホストは別の製品として提供されている実行中のコード分析ツール、IBM Rational Developer for z Systems Host Utilities をサポートします。このコード分析をホスト上で実行する利点は、それを日常のバッチ処理として組み込むことができる点です。

以下のコード分析ツールを、ホスト上で使用することができます。

- ・コード・レビュー: コード・レビューは、異なる重大度レベルを持つ規則を使用して、ソース・コードをスキャンし、規則違反を報告します。
- ・コード・カバレッジ: 実行中のプログラムを分析し、実行可能な行の総数に対する実行済みの行についてのレポートを生成します。

要件およびチェックリスト

ホスト・ベースのコード分析ツールをご使用のサイトで使用開始する際には、他の管理者の支援は必要ありません。ただし、以下のタスクを実行する必要があります。特に断りがない限り、すべてのタスクは必須です。

1. IBM Rational Developer for z Systems Host Utilities をインストールします
(「*Program Directory for IBM Rational Developer for z Systems Host Utilities*」(GI88-4326) を参照)。提供されているデフォルトを使用すると、製品は高位修飾子の AKG および z/OS UNIX パス/usr/lpp/rdzutil を使用してインストールされます。
2. AKG.SAKGSAMP(AKGSETUP) をカスタマイズして実行依頼することで、提供されているサンプルのカスタマイズ可能なコピーを作成します。このジョブは、以下のタスクを実行します。
 - ・ AKG.#CUST.PROCLIB を作成し、これにサンプル SYS1.PROCLIB メンバーを取り込みます。
 - ・ AKG.#CUST.JCL を作成し、これにサンプル構成 JCL を取り込みます。

注: IBM Debug Tool for z/OS は、コード・カバレッジ・コンポーネントの前提条件です。

コード・レビュー

コード・レビューは、ソース・コードをスキャンし、異なる重大度レベルを持つ規則を使用して、規則違反を報告します。ツールには COBOL および PL/I の規則プロバイダーが付属していますが、それ以外の規則プロバイダーを追加することもできます。

Developer for z Systems Host Utilities は、バッチ・モードでのコード・レビュー・サービスの呼び出しを簡素化するために、サンプル・プロシージャ AKGCR を提供しています。AKGCR は AKG.#CUST.PROCLIB に置かれます。ただし、AKG.SAKGSAMP(AKGSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。

サンプル・プロシージャー AKG.#CUST.PROCLIB(AKGCR) を、メンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーします。

AKGCR プロシージャーをシステム・プロシージャー・ライブラリーにコピーできない場合は、呼び出しジョブに JCLLIB カードを JOB カードの直後に追加するように Developer for z Systems ユーザーに依頼します。

```
//MYJOB    JOB <job parameters>
//PROCS    JCLLIB ORDER=(AKG.#CUST.PROCLIB)
```

コード・レビュー・プロセスの変更

Developer for z Systems コード・レビューは、サード・パーティーのコードを、レビュー・プロセスの一部にすることを可能にします。例えば、C/C++ コードを分析する規則プロバイダーを提供したり、サイト固有のコーディング規則を認識するように COBOL 規則プロバイダーを拡張したりすることができます。

ホスト・ベースのコード・レビューは、Developer for z Systems クライアントと同じく Eclipse プロセスです。そのため、開発サポート・チームがクライアント上でコード・レビュー用に行った機能拡張は、ホスト上でも再利用することができます。

機能拡張は、Eclipse プラグインまたは Eclipse フィーチャーで構成されます。それらをアクティブ化するには、AKGCRADD 構成ジョブで記述されているように、それらを既存のコードで使用できるようにします。AKGCRADD は AKG.#CUST.JCL に置かれます。ただし、AKG.SAKGSAMP(AKGSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。

コード・カバレッジ

コード・カバレッジは、実行中のプログラムを分析し、実行可能な行の総数に対する実行済みの行についてのレポートを生成します。コード・カバレッジは、一時ポートを使用して、IBM Debug Tool for z/OS、または Developer for z Systems クライアントの統合デバッガー・デバッグ・エンジンとの TCP/IP 接続をセットアップすることに注意してください。

Developer for z Systems Host Utilities には、コード・カバレッジをバッチ・モードで呼び出す 2 とおりの方法があります。単一のプログラム実行を処理するサンプル JCL プロシージャーを使用する方法と、複数のプログラム実行を処理できる永続的にアクティブなコード・カバレッジ・コレクターを開始および停止するためのスクリプト・セットを使用する方法です。

コード・カバレッジの単一の呼び出し

AKGCC サンプル・プロシージャーは、コード・カバレッジ・コレクターを開始して、単一プログラム実行を分析させ、コレクターを停止し、結果をアーカイブして後で利用できるようにするための手段となります。

AKGCC は AKG.#CUST.PROCLIB に置かれます。ただし、AKG.SAKGSAMP(AKGSETUP) ジョブをカスタマイズして実行依頼したときに別のロケーションを指定した場合は除きます。

サンプル・プロシージャ AKG.#CUST.PROCLIB(AKGCC) を、メンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーします。

AKGCC プロシージャをシステム・プロシージャ・ライブラリーにコピーできない場合、JCLLIB カード (JOB カードの直後) を呼び出しジョブに追加するように Developer for z Systems ユーザーに依頼します。

```
//MYJOB      JOB <job parameters>
//PROCS      JCLLIB ORDER=(AKG.#CUST.PROCLIB)
```

コード・カバレッジの複数の呼び出し

AKGCC プロシージャにはすべてをこれで処理できる利便性がありますが、ソフトウェア開発プロセスのセットアップ方法によっては、プログラム分析のたびにリソースと時間を使ってコード・カバレッジ・コレクターを開始しなければならないことの方が重大かもしれません。

Developer for z Systems Host Utilities には、アクティブな状態を維持するコード・カバレッジ・コレクターを開始するための `ccstart` スクリプトが用意されています。開始したコレクターを、コード・カバレッジの複数の呼び出しで使用できます。コレクターを停止するには、`ccstop` スクリプトを使用できます。

これらのスクリプト (`ccstart`、`ccstop`、および `codecov`) は、Developer for z Systems Host Utilities をデフォルトのロケーションにインストールした場合、`/usr/lpp/rdzutil/bin/` にあります。これらのスクリプトの使用法については、Developer for z Systems IBM Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html) を参照してください。

一般的な使用シナリオを以下に示します。

1. 固定ポート番号を使用するオプションと出力を既知のロケーションにリダイレクトするオプションを指定して `ccstart` を呼び出します。
2. 始動オプションで、分析対象のプログラムを必要な回数呼び出します。
`TEST(,,,TCPIP&<hostip>:<port>)`。
3. `ccstop` を呼び出します。
4. Developer for z Systems クライアントから出力ロケーションにアクセスして、レポートを表示します。

コード・カバレッジの出力

コード・カバレッジの出力は、Developer for z Systems クライアントにインポートすることを目的としているため、z/OS UNIX ファイルに書き込まれます。またコード・カバレッジは、以前の実行結果を使用し、それらを現在の実行結果と組み合わせて複数のコード・パスをカバーする単一のレポートを作成することができます。

これらの理由により、Developer for z Systems Host Utilities はコード・カバレッジの実行の出力を削除しようとしなないため、出力は長期にわたって累積されます。

z/OS UNIX には、ファイルが入っているディレクトリーとファイルの経過日数に基づいてファイルを削除する、`skulker` というシェル・スクリプトがあります。指定された日時にコマンドを実行する z/OS UNIX `cron` デーモンと結合すれば、定期的にターゲット・ディレクトリーを空にする自動化ツールをセットアップできます。

skulker スクリプトおよび cron デーモンの詳細については、「*UNIX System Services* コマンド解説書」(SA88-8641) を参照してください。

第 7 章 その他のカスタマイズ・タスク

このセクションは、さまざまなオプションのカスタマイズ・タスクを結合したものです。必要なサービスを構成するには、該当するセクションの説明に従ってください。

Developer for z Systems 構成ファイルへのカスタマイズ

- 『include.conf、C/C++ コンテンツ・アシスト用の強制インクルード』

他の製品への Developer for z Systems 関連のカスタマイズ：

- 98 ページの『z/OS UNIX サブプロジェクト』
- 99 ページの『インクルード・プリプロセッサのサポート』
- 100 ページの『Enterprise COBOL および PL/I での xUnit サポート』
- 101 ページの『エンタープライズ・サービス・ツール・サポート』
- 102 ページの『CICS 双方向言語サポート』
- 103 ページの『エンタープライズ・サービス・ツールの診断用 IRZ メッセージ』
- 104 ページの『FEKRNPLI、PL/I リモート・プリプロセッサ』

include.conf、C/C++ コンテンツ・アシスト用の強制インクルード

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

C/C++ のコンテンツ・アシストは、include.conf の定義を使用して、指定されたファイルまたはメンバーの強制インクルードを実行できます。強制インクルードは、ファイルまたはメンバーがプリプロセッサ・ディレクティブを使用してソース・コードに組み込まれたかどうかに関係なく、コンテンツ・アシスト操作の実行時に構文解析されるファイルまたはディレクトリー、データ・セット、またはデータ・セット・メンバーで構成されます。

ファイルは、使用される前に、rdz.env で include.c 変数または include.cpp 変数によって参照される必要があります。このように rdz.env で参照されることは、C および C++ で使用されるように別のファイルを指定できることを意味します。rdz.env の変数はデフォルトで使用不可になっています。

サンプルの include.conf は /etc/zexpl/ に置かれます。ただし、ジョブ FEL.SFELSAMP(FELSETUP) をカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳細については、17 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO OEDIT コマンドで編集できます。

定義は列 1 から始める必要があります。US コード・ページを使用する場合、コメント行はポンド記号 (#) で始まります。データ行に入れることができるのは、ディレクトリー、ファイル、データ・セット、またはメンバーの名前だけです。同じ行

にコメントを入れることはできません。行の継続はサポートされていません。

```
# 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
```

図 23. *include.conf* - C/C++ コンテンツ・アシスト用の強制インクルード

z/OS UNIX サブプロジェクト

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

REXEC (リモート実行) は、クライアントがホスト・システム上でコマンドを実行できるようにする TCP/IP サービスです。SSH (セキュア・シェル) も同様のサービスですが、すべての通信が暗号化されます。Developer for z Systems は、いずれかのサービスを使用して、z/OS UNIX サブプロジェクトでリモート (ホスト・ベース) アクションを実行します。

注:

- Developer for z Systems は、TSO バージョンでなく、z/OS UNIX バージョンの REXEC を使用します。
- REXEC/SSH がデフォルト・ポートを使用するように構成されていない場合、Developer for z Systems クライアントは z/OS UNIX サブプロジェクトに使用する正しいポートを定義する必要があります。この構成を行うには、「ウィンド

ウ」 > 「設定」 > 「z/OS ソリューション」 > 「USS サブプロジェクト」 > 「リモート・アクション・オプション」 設定ページを選択します。使用するポートについては、『REXEC または SSH のセットアップ』を参照してください。

REXEC または SSH のセットアップ

REXEC および SSH は、別の TCP/IP サービスである、INETD (Internet Daemon) によって提供されるサービスに依存します。INETD、REXEC、および SSH をセットアップするために必要なステップについては、「*Communications Server IP 構成ガイド*」(SC88-8926) に説明があります。詳細およびその他のセットアップ方法については、Developer for z Systems ライブラリー (<http://www-01.ibm.com/support/docview.wss?uid=swg27038517>) にあるホワイト・ペーパー「*Using INETD, REXEC and SSH with Developer for System z*」(SC14-7301) を参照してください。

REXEC で使用される共通のポートは 512 です。使用されるポートを確認するには、`/etc/inetd.conf` および `/etc/services` を参照してください。

- `/etc/inetd.conf` で、`rexecd` サーバー (7 番目のワード) のサービス名 (最初のワード。この例では `exec`) を見つけます。

```
exec stream tcp nowait OMVSKERN /usr/sbin/orexecd rexecd -LV
```

- `/etc/services` で、そのサービス名 (最初のワード) に接続しているポート (2 番目のワード。この例では 512) を見つけます。

```
exec      512/tcp      #REXEC      Command Server
```

同じ原則が SSH にも適用されます。その共通のポートは 22 で、サーバー名は `sshd` です。

インクルード・プリプロセッサのサポート

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

Developer for z Systems は、一部のサード・パーティーのインクルード・ステートメントを含む COBOL および PL/I インクルード・ステートメントの解釈と拡張をサポートします。Developer for z Systems では、サンプルの REXX `exec`、FEKRNPLI も提供され、これらは Developer for z Systems クライアントによって呼び出すことができ、PL/I コンパイラーを呼び出して PL/I ソースを拡張します。

FEKRNPLI は `FEL.#CUST.CNTL` に置かれます。ただし、`FEL.SFELSAMP(FELSETUP)` ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『第 2 章 基本的なカスタマイズ』を参照してください。

サンプルの `FEL.#CUST.CNTL(FEKRNPLI) exec` を、このメンバーの中で説明されているようにカスタマイズします。以下の情報を指定する必要があります。

- `compiler_hlq`: PL/I コンパイラーの高位修飾子

Developer for z Systems クライアントは、TSO コマンド・サービスを使用して exec を実行します。このことは、FEKRNPLI exec が TSO コマンド・サービスの SYSPROC または SYSEXEC 連結の中に置かれている場合、ユーザーは exec の正確なロケーションを知っている必要がないことを意味します。ユーザーが知っている必要があるのは名前だけです。レガシー ISPF ゲートウェイを使用する場合は、ISPF.conf の中で SYSPROC または SYSEXEC 連結が定義されます。このファイルのカスタマイズについては、「IBM Explorer for z/OS Host Configuration Guide (SC27-8437)」で説明されています。

Enterprise COBOL および PL/I での xUnit サポート

このカスタマイズ・タスクには、支援は必要ありませんが、以下のリソースまたは特殊なカスタマイズ・タスクが必要となります。

- LINKLIST 更新
-

反復可能な自己検査単体テストを実行するためのコードを作成する上で開発者を支援するフレームワークは、まとめて xUnit と呼ばれます。Developer for z Systems は、zUnit という、Enterprise COBOL および PL/I コードの単体テストのためのフレームワークを備えています。

zUnit フレームワークを使用するには、開発者は、STEPLIB または LINKLIST を通じて、FEL.SFELLOAD ロード・ライブラリー内の AZU* および IAZU* ロード・モジュールにアクセスする必要があります。同様に、zUnit テスト実行機能の AZUTSTRN は、STEPLIB または LINKLIST のいずれかを通じて、次のようにさまざまなシステム・ライブラリーにアクセスする必要があります。

- CEE.SCEERUN および CEE.SCEERUN2 (LE ランタイム)
- SYS1.CSSLIB (呼び出し可能システム・サービス)
- SYS1.SIXML0D1 (XML ツールキット)

zUnit テスト実行機能は別のテスト・ケースを保持するロード・ライブラリーにもアクセスする必要があります。このライブラリーは大抵、開発者に固有のものです。

zUnit テスト実行機能の AZUTSTRN は、Developer for z Systems クライアントによってバッチ・モードで、TSO コマンド行から、および z/OS UNIX コマンド行から呼び出すことができます。

- Developer for z Systems は、バッチ・モードでの zUnit テスト実行機能の呼び出しを簡素化するためにサンプル・プロシージャ AZUZUNIT を提供しています。AZUZUNIT は FEL.#CUST.PROCLIB に置かれます。ただし、FEL.SFELSAMP (FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

サンプル・プロシージャ FEL.#CUST.PROCLIB(AZUZUNIT) を、このメンバー内で説明されているようにカスタマイズし、SYS1.PROCLIB にコピーしてください。

プロシージャの名前とプロシージャ内のステップの名前は、Developer for z Systems クライアントに付属しているデフォルトのプロパティーに一致していま

す。プロシージャーの名前またはプロシージャー内のステップの名前を変更する場合は、すべてのクライアント上の対応するプロパティ・ファイルを更新する必要があります。プロシージャー名とステップ名は変更しないでください。

AZUZUNIT プロシージャーをシステム・プロシージャー・ライブラリーにコピーできない場合は、呼び出しジョブに JCLLIB カードを JOB カードの直後に追加するように Developer for z Systems ユーザーに依頼します。

```
//MYJOB    JOB <job parameters>
//PROCS    JCLLIB ORDER=(FEL.#CUST.PROCLIB)
```

- z/OS UNIX から (/usr/lpp/ibm/rdz/bin/zunit スクリプトを使用して) zUnit テスト実行機能呼び出す場合、rdz.env の STEPLIB ディレクティブで、必要な非 LINKLIST データ・セットを指定して、開発者のセットアップ作業を簡素化することができます。

rdz.env は /etc/zexpl/ に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。このファイルは、TSO OEDIT コマンドで編集できます。

zunit スクリプトを使用すると、ユーザーはそのスクリプトによって使用される STEPLIB ディレクティブに追加するデータ・セットを指定できます。

- FEL.SFELPROC(FEKZUNIT) exec を使用して TSO コマンド行から zUnit テスト・ランナーを呼び出す場合、システム・ライブラリーが LINKLIST に存在していなければなりません。存在しない場合、開発者は zUnit テスト・ランナーを呼び出すたびに、システム・データ・セット名を指定する必要があります。開発者のために、このデータ・セットの **TSOLIB** 割り振りを行うラッパー exec を作成することもできます。このラッパー exec のコーディング方法の例として、FEKZUNIT だけで使用することができます。

zUnit テスト・ランナーは、テスト・レポートの自動再フォーマット設定を可能にします。Developer for z Systems には、サンプルの変換 (Ant または jUnit フォーマットへの変換など) が用意されており、/usr/lpp/ibm/rdz/samples/zunit/xsd および /usr/lpp/ibm/rdz/samples/zunit/xsl に置かれています (Developer for z Systems をデフォルト・ロケーション /usr/lpp/ibm/rdz にインストールした場合)。

エンタープライズ・サービス・ツール・サポート

このカスタマイズ・タスクには、支援や特殊リソース、または特殊なカスタマイズ・タスクは必要ありません。

Developer for z Systems クライアントには、エンタープライズ・サービス・ツールと呼ばれるコード生成コンポーネントがあります。生成されるコードのタイプによっては、このコードは Developer for z Systems ホスト・システム・インストールで提供される機能に依存しています。これらのホスト・システム機能を使用可能にする方法を、以下のセクションで説明しています。

- 102 ページの『CICS 双方向言語サポート』
- 103 ページの『エンタープライズ・サービス・ツールの診断用 IRZ メッセージ』

注: エンタープライズ・サービス・ツールには、サービス・フロー・モデラー (SFM) やエンタープライズ用 XML サービスなどの複数のツールが含まれます。

CICS 双方向言語サポート

このカスタマイズ・タスクを完了するには、CICS 管理者の支援が必要になります。このタスクには、以下のリソースまたは特殊なカスタマイズ・タスクが必要です。

- CICS 領域 JCL を更新する
 - CICS に対してプログラムを定義する
-

Developer for z Systems エンタープライズ・サービス・ツールのコンポーネントは、さまざまな形式のアラビア語およびヘブライ語のインターフェース・メッセージをサポートしているほか、すべてのエディターおよびビューで双方向言語データの表示と編集をサポートしています。端末アプリケーションでは、左から右と、右から左の両方の画面がサポートされ、数値フィールド、および画面とは反対の向きのフィールドもサポートされます。

そのほかの双方向言語フィーチャーおよび機能には、以下のものがあります。

- エンタープライズ・サービス・ツールのサービス・リクエストは、インターフェース・メッセージの双方向属性を動的に指定します。
- サービス・フロー内の双方向データ処理は、双方向属性 (テキスト・タイプ、テキスト方向、数値スワッピング、対称スワッピングなど) に基づきます。これらの属性は、インターフェース・フローと端末フローのどちらの場合でも、さまざまなフロー作成のステージで指定できます。
- エンタープライズ・サービス・ツールの生成ランタイム・コードには、異なる双方向属性を持つメッセージ内のフィールド間でのデータの変換が含まれます。

さらに、エンタープライズ・サービス・ツールの生成コードは、CICS SFR (サービス・フロー・ランタイム) 以外の環境での bidi 変換をサポートできます。その一例がバッチ・アプリケーションです。エンタープライズ・サービス・ツールの生成ウィザードで適切な bidi 変換オプションを指定し、生成されたプログラムを適切な双方向変換ライブラリー FEL.SFELLOAD とリンクすることにより、エンタープライズ・サービス・ツールの生成プログラムに、双方向変換ルーチンの呼び出しを組み込ませることができます。

CICS 双方向言語サポートをアクティブにするには、以下のタスクを実行します。

1. FEL.SFELLOAD ロード・モジュール FEJBDCMP および FEJBDTRX を CICS RPL 連結 (DD ステートメント DFHRPL) の中に置きます。この作業は、適用された保守が自動的に CICS で使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。

重要: インストール・データ・セットを連結せず、モジュールを新規または既存のデータ・セットにコピーする場合は、このモジュールが DLL であり、PDSE ライブラリー内に存在しなければならないことに留意してください。

2. 適切な CEDA コマンドを使用して、FEJBDCMP および FEJBDTRX をプログラムとして CICS に対して定義します。

エンタープライズ・サービス・ツールの診断用 IRZ メッセージ

このカスタマイズ・タスクには、支援は必要ありませんが、以下のリソースまたは特殊なカスタマイズ・タスクが必要となります。

- LINKLIST 更新
- CICS 領域 JCL 更新
- CICS 領域 CSD 更新

Developer for z Systems クライアントには、エンタープライズ・サービス・ツールと呼ばれるコード生成コンポーネントがあります。エンタープライズ・サービス・ツールを使用して生成したコードから診断用エラー・メッセージを出すためには、FEL.SFELMOD ロード・ライブラリー内のすべての IRZM* モジュールおよび IIRZ* モジュールを、その生成コードで使用できるようにする必要があります。エンタープライズ・サービス・ツールでは、以下の環境向けにコードを生成できます。

- CICS
- IMS
- MVS バッチ

生成されたコードを CICS トランザクションで実行する場合は、以下の手順を実行します。

- FEL.SFELMOD 内のすべての IRZM* および IIRZ* モジュールを、CICS 領域の DFHRPL DD に追加します。この作業は、適用された保守が自動的に使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。
- IRZCSD ジョブをカスタマイズおよび実行依頼して、CICS 領域の CICS システム定義 (CSD) を更新します。カスタマイズの手順については、メンバー内のドキュメンテーションを参照してください。IRZCSD は FEL.#CUST.JCL に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

それ以外の状態では、STEPLIB または LINKLIST を通じて、FEL.SFELMOD 内のすべての IRZM* モジュールと IIRZ* モジュールを使用可能にします。この作業は、適用された保守が自動的に使用可能になるように、インストール・データ・セットを連結に追加する方法で行ってください。

STEPLIB を使用する場合は、LINKLIST を介して使用できないモジュールを、コードを実行するタスクの STEPLIB ディレクティブで定義します。

ロード・モジュールを使用できず、生成コードでエラーが発生した場合は、次のメッセージが出されます。

IRZ9999S Failed to retrieve the text of a Language Environment runtime message. Check that the Language Environment runtime message module for facility IRZ is installed in DFHRPL or STEPLIB.

注:

- モジュール `FEL.SFELLMOD(IRZPWSIO)` は、トップダウン IMS MPP コード生成中に静的にリンクされます。したがって、このモジュールは、生成されたコードの実行時に使用可能であってはなりません。使用可能にできるのはコンパイル中のみです。
- バージョン 9.0.1 では、`FEL.SFELLMOD(IRZPWSIO)` および関連する `FEL.SFELSAMP(IRZPWSH)` サンプル PL/I インクルード・メンバーが Developer for z Systems から IMS バージョン 12 に移動されました。これらのパーツは、それぞれ `IMS.SDFSRESL(DFSPWSIO)` および `IMS.SDFSSMPL(DFSPWSH)` に名前変更されています。

FEKRNPLI、PL/I リモート・プリプロセッサー

このカスタマイズ・タスクには、支援は必要ありませんが、以下のリソースまたは特殊なカスタマイズ・タスクが必要となります。

Developer for z Systems には、Developer for z Systems クライアントから呼び出すことのできるサンプル PL/I プリプロセッサーが用意されています。

サンプルのメンバー `FEKRNPLI` は `FEL.#CUST.CNTL` に置かれます。ただし、ジョブ `FEL.SFELSAMP(FELSETUP)` をカスタマイズして実行依頼したときに、別のロケーションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。メンバー内に記載されているようにサンプル PL/I プリプロセッサー (REXX) `exec FEL.#CUST.CNTL(FEKRNPLI)` をカスタマイズし、Developer for z Systems ユーザーに `exec` の場所を通知します。

第 8 章 インストール検査

製品のカスタマイズの完了後、この章で説明するインストール検査プログラム (IVP) を使用して、主要な製品コンポーネントのセットアップが正常であることを検査できます。

開始タスクの検査

DBGMGR、デバッグ・マネージャー

オプションの DBGMGR 開始タスクまたはユーザー・ジョブを開始します。正常に開始されると、サーバーは以下のコンソール・メッセージを出します。clientport は外部 (クライアント/ホスト) 通信に使用されるポートの番号、hostport は内部 (ホスト限定) 通信に使用されるポート番号です。

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

ジョブが戻りコード 66 を伴って終了する場合は、FEL.SFELAUTH に APF 許可がありません。

注: 続けて他のデバッグ関連 IVP テストを行う前に、DBGMGR を開始してください。

サービスの検査

Developer for z Systems インストールは、基本およびオプションのサービス用に、いくつかのインストール検査プログラム (IVP) を提供します。IVP スクリプトは、インストール・ディレクトリー (デフォルトでは /usr/lpp/ibm/rdz/bin/) に置かれます。

表 15. サービス用の IVP

fekfivpc	106 ページの『CARMA 接続』
fekfivps	107 ページの『SCLMDT 接続』

以下のセクションで説明するタスクでは、ユーザーが z/OS UNIX でアクティブである必要があります。そのためには、**OMVS** TSO コマンド を発行します。TSO に戻るには、**exit** コマンドを使用します。

IVP を実行するユーザー ID には大きな領域サイズが必要です。これは、大量のメモリーを必要とする機能 (Java など) が実行されるからです。領域サイズは、131072 キロバイト (128 メガバイト) 以上に設定してください。

次の例のエラーは、領域サイズが不十分であることを明示していますが、他のエラーが発生する可能性もあります。例えば Java を始動できない場合があります。

```
CEE5213S The signal SIGPIPE was received.
%z/OS UNIX command%: command was killed by signal number 13
  %line-number% **      %REXX command%
      +++ RC(137) +++
```


注: IVP テストを開始する前に、z/OS Explorer および Developer for z Systems の開始タスクをアクティブにしておく必要があります。

IVP の初期化

このセクションのサンプル・コマンドはすべて、特定の環境変数が設定されていることを必要とします。これにより、IVP スクリプトは PATH ステートメントを通じて入手でき、カスタマイズされた構成ファイルのロケーションが判明します。**pwd** コマンドおよび **cd** コマンドを使用して、現行ディレクトリを検査し、カスタマイズした構成ファイルがあるディレクトリに変更してください。その後、次のサンプルのように **ivpinit** シェル・スクリプトを使用して RSE 環境変数を設定できます。\$ は z/OS UNIX プロンプトです。

```
$ 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/ispf/bin:/bin:/usr/lpp/ibm/rdz/bin:/bin
```

./ivpinit の最初のピリオド (.) は、シェルを現行環境で実行するための z/OS UNIX コマンドです。これにより、シェル内で設定された環境変数が、シェルを出た後も有効になります。2 番目のピリオド (.) は現行ディレクトリを参照しています。

注:

- **./ivpinit** を **fekfivp*** スクリプトの前に実行しない場合は、呼び出すときにこれらのスクリプトへのパスを指定する必要があります。以下に例を示します。

```
/usr/lpp/ibm/rdz/bin/fekfivpc
```

また、**./ivpinit** が最初に実行されない場合、すべての **fekfivp*** スクリプトは、カスタマイズされた構成ファイルを保持するディレクトリの場所を要求します

CARMA 接続

次のコマンドを実行することにより、CARMA への接続を検査します。

```
fekfivpc
```

このコマンドから成功メッセージが返されなければなりません。

注: IVP が失敗した場合は、**/tmp/fekfivpc.log** の内容を検査します。このログには RSE と CARMA の間の通信が記録されており、失敗の根本原因の究明に役立つ情報が含まれている可能性があります。

fekfivpc には、以下に示すオプションの非定位置パラメーターがあります。

-noram

デフォルトでは、**fekfivpc** は、**CRADEF VSAM** データ・セット内で最初に定義された **RAM** を始動します。**RAM** をテストすることが望ましくない場合もあります。例えば、サード・パーティーの **RAM** が最初にリストさ

れていて、予期しない入力をその RAM が必要とするような場合です。そのような場合、`-noram` 始動引数を使用して、IVP テストの RAM 固有のステップを省略することができます。

SCLMDT 接続

次のコマンドを実行することにより、SCLM Developer Toolkit への接続を検査します。

`fekfivps`

このコマンドから成功メッセージが返されなければなりません。

`fekfivps` には、以下に示すオプションの非定位置パラメーターがあります。

-file `fekfivps` は、数百行に上る大量の出力を生成する場合があります。`-file` パラメーターは、この出力をファイル `$TMPDIR/fekfivps.log` へ送信します。ここで、`$TMPDIR` は `rse.env` 内の `TEMPDIR` ディレクティブの値で、デフォルトは `/tmp` です。

-debug `-debug` パラメーターは、詳細なテスト出力を作成します。IBM サポートから指示された場合以外は、このオプションを使用しないでください。

第 9 章 セキュリティー定義

サンプル FELRACF ジョブをカスタマイズし、実行依頼してください。このメンバーには、Developer for z Systems 用の基本セキュリティー定義を作成する、サンプルの RACF コマンドが含まれています。サンプル AQERACF ジョブをカスタマイズして実行依頼します。このジョブには、統合デバッガーのセキュリティー定義を作成するサンプル RACF コマンドが含まれています。

FELRACF および AQERACF は FEL.#CUST.JCL に置かれます。ただし、FEL.SFELSAMP(FELSETUP) ジョブをカスタマイズして実行依頼したときに、別のローションを指定した場合は除きます。詳しくは、17 ページの『カスタマイズのセットアップ』を参照してください。

RACF コマンドの詳細については、「RACF コマンド言語解説書」(SA88-8617) を参照してください。

要件およびチェックリスト

セキュリティーのセットアップを完了するために、セキュリティー管理者は、表 16 に示す値を知っている必要があります。これらの値は、前のステップである Rational Developer for z Systems のインストールとカスタマイズで定義されています。

表 16. セキュリティー・セットアップの変動要素

説明	デフォルト値 正解の入手先	値
Developer for z Systems 製品 の高位修飾子	<ul style="list-style-type: none">FELSMP/E インストール	
Developer for z Systems カス タマイズの高位修飾子	<ul style="list-style-type: none">FEL.#CUSTFEL.SFELSAMP(FELSETUP) (17 ページの『カスタマイ ズのセットアップ』を参 照)	
統合デバッガー開始タスク名	<ul style="list-style-type: none">DBGMGRFEL.#CUST.PROCLIB (DBGMGR) - 26 ページの 『PROCLIB の変更』の説 明を参照	

以下のリストは、Developer for z Systems の基本的なセキュリティー・セットアップを完了するために必要な操作の概要を示したものです。以下の各セクションで説明されているように、これらの要件を満たすために、求めるセキュリティー・レベルに応じてさまざまな方式を使用できます。

- 110 ページの『セキュリティーの設定およびクラスをアクティブにする』

- 『Developer for z Systems 開始タスクの定義』
- 111 ページの『セキュアな z/OS UNIX サーバーとしてデバッグ・マネージャーを定義する』
- 111 ページの『デバッグ・マネージャーの MVS プログラム制御ライブラリーを定義する』
- 112 ページの『統合デバッガーへのアクセスの定義』
- 113 ページの『データ・セット・プロファイルを定義する』
- 114 ページの『セキュリティ設定の検査』

セキュリティの設定およびクラスをアクティブにする

Developer for z Systems は、セキュアで制御されたホスト・システム環境をクライアントに保証するために、さまざまなセキュリティ・メカニズムを使用しています。そのため、以下のサンプルの RACF コマンドで示すように、いくつかのクラスおよびセキュリティ設定をアクティブにする必要があります。

- 現行の設定を表示する
 - SETROPTS LIST
- 統合デバッガー用の機能クラスをアクティブにする
 - SETROPTS GENERIC(FACILITY)
 - SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
- 統合デバッガー用の開始タスク定義をアクティブにする
 - SETROPTS GENERIC(STARTED)
 - RDEFINE STARTED ** STDATA(USER(=MEMBER) GROUP(STCGROUP) TRACE(YES))
 - SETROPTS CLASSACT(STARTED) RACLIST(STARTED)
- 統合デバッガー用のプログラム制御をアクティブにする
 - RDEFINE PROGRAM ** ADDMEM('SYS1.CMDLIB'//NOPADCHK) UACC(READ)
 - SETROPTS WHEN(PROGRAM)

注: すでに PROGRAM クラス内に * プロファイルがある場合は、** プロファイルを作成しないでください。セキュリティ・ソフトウェアによって使用される検索パスが、分かりにくく、複雑なものになります。その場合は、既存の * 定義と新しい ** 定義をマージする必要があります。 ** プロファイルの使用については、「*Security Server RACF セキュリティ管理者のガイド*」(SA88-8613) に説明があります。

重要: 「WHEN PROGRAM」がアクティブの場合、一部の製品 (FTP など) はプログラムで制御することが必要です。このプログラム制御は、実動システム上でアクティブにする前にテストしてください。

Developer for z Systems 開始タスクの定義

以下のサンプル RACF コマンドは、保護されたユーザー ID (STCDBM) および STCGROUP グループを割り当てて DBGMGR 開始タスクを作成します。

•

```

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 DBGMR.* DATA('DEBUG MANAGER')
STDATA(USER(STCDBM) GROUP(STCGROUP) TRUSTED(NO))

• SETROPTS RACLIST(STARTED) REFRESH

```

注:

- NOPASSWORD キーワードを指定することにより、開始タスクのユーザー ID が必ず保護されるようにしてください。
- デバッグ・マネージャー開始タスク (DBGMR) は、統合デバッガー機能によってのみ使用されます。

セキュアな z/OS UNIX サーバーとしてデバッグ・マネージャーを定義する

統合デバッガーは、デバッグ・スレッドのセキュリティ環境を作成または削除するために、BPX.SERVER プロファイルに対する UPDATE アクセス権を必要とします。UID(0) を使用したこの要求の回避はサポートされていません。この許可は、オプションの統合デバッガー機能を使用する場合にのみ必要となります。

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

重要: BPX.SERVER プロファイルを定義すると、z/OS UNIX 全体が UNIX レベルのセキュリティから、より安全な z/OS UNIX レベルのセキュリティに切り替わります。この切り替えによって、他の z/OS UNIX アプリケーションと操作が影響を受ける場合もあります。セキュリティは、実動システム上でアクティブにする前にテストしてください。各種セキュリティ・レベルの詳細については、「*UNIX System Services 計画*」(GA88-8639)を参照してください。

デバッグ・マネージャーの MVS プログラム制御ライブラリーを定義する

BPX.SERVER に対する権限を持つサーバーは、クリーンなプログラム制御環境で実行する必要があります。この要件は、デバッグ・マネージャーによって呼び出されるすべてのプログラムも、プログラムで制御する必要があることを意味します。MVS ロード・ライブラリーの場合、プログラム制御はセキュリティ・ソフトウェアによって管理されます。

デバッグ・マネージャーは、システム・ライブラリー、言語環境のランタイム、および Developer for z Systems の (ISP.SISPLoad) ロード・ライブラリーを使用します。

- 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

注: すでに PROGRAM クラス内に * プロファイルがある場合は、** プロファイルを使用しないでください。このプロファイルを使用すると、セキュリティー・ソフトウェアで使用される検索パスが分かりにくく複雑になります。その場合は、既存の * 定義と新しい ** 定義をマージする必要があります。** プロファイルの使用については、「*Security Server RACF セキュリティー管理者のガイド*」(SA88-8613) に説明があります。

オプションのサービスを使用できるようにするには、以下の前提条件の追加ライブラリーがプログラムで制御されるようにする必要があります。このリストには、Developer for z Systems と対話する製品 (IBM Explorer for z/OS など) に固有のデータ・セットは含まれていません。

- 代替 REXX ランタイム・ライブラリー、SCLM Developer Toolkit 用
 - REXX.*.SEAGALT

注: LPA 配置用に設計されたライブラリーは、LINKLIST または STEPLIB によってアクセスされる場合、追加のプログラム制御権限も必要とします。この資料では、以下の LPA ライブラリーの使用法について説明します。

- REXX ランタイム・ライブラリー、SCLM Developer Toolkit 用
 - REXX.*.SEAGLPA
- Developer for z Systems、CARMA 用
 - FEL.SFELLPA

統合デバッガーへのアクセスの定義

ユーザーが問題プログラム状態のプログラムのデバッグに統合デバッガーを使用できるようにするには、リストされている AQE.AUTHDEBUG.* プロファイルのいずれかに対する READ アクセス権が必要です。AQE.AUTHDEBUG.AUTHPGM プロファイルへのアクセスが許可されているユーザーは、APF 許可済みのプログラムをデバッグすることもできます。#apf プレースホルダーは、許可済みプログラムのデバッグが許可されているユーザーの、有効なユーザー ID または RACF グループ名に置き換えてください。

- 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

注: バージョン 9.1.1 より前のバージョンの IBM Rational Developer for System z では、現在使用されていない別の FACILITY クラス・プロファイル AQE.AUTHDEBUG.WRITEBUFFER を使用していました。これは、ご使用のホスト・システムにバージョン 9.1.1 以上の IBM Rational Developer for System z しかない場合には、削除できます。

データ・セット・プロファイルを定義する

ほとんどの Developer for z Systems データ・セットでは、ユーザーの場合は READ アクセス権、システム・プログラマーの場合は ALTER で十分です。#sysprog プレースホルダーは、有効なユーザー ID または RACF グループ名に置き換えてください。また、正しいデータ・セット名については、製品をインストールおよび構成したシステム・プログラマーに問い合わせてください。FEK は、インストール時に使用されたデフォルトの高位修飾子で、FEL.#CUST はカスタマイズ・プロセスで作成されたデータ・セットのデフォルトの高位修飾子です。

```
•
|
|      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
```

注:

- FEL.SFELAUTH には APF 許可があるため、このデータ・セットを更新されないよう保護してください。
- この資料内および FELRACF ジョブ内のサンプル・コマンドでは、拡張総称命名 (EGN) がアクティブであることを想定しています。EGN がアクティブな場合、** 修飾子を使用して、DATASET クラス内の任意の数の修飾子を表すことができます。使用しているシステムで EGN がアクティブでない場合は、** を * に置き換えてください。EGN の詳細については、「*Security Server RACF セキュリティー管理者のガイド*」(SA88-8613) を参照してください。

一部の Developer for z Systems コンポーネントには、追加のセキュリティー・データ・セット・プロファイルが必要です。#sysprog および #ram-developer の各プレースホルダーは、有効なユーザー ID または RACF グループ名に置き換えてください。

- SCLM Developer Toolkit のロング/ショート・ネーム変換を使用している場合は、ユーザーにマッピング VSAM の FEL.#CUST.LSTRANS.FILE に対する UPDATE アクセス権が必要です。

```
—
|
|      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
```

- CARMA RAM (Repository Access Manager) 開発者には、CARMA VSAM である FEL.#CUST.CRA* に対する UPDATE アクセス権が必要です。

—

```

|          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

```

セキュリティ設定の検査

セキュリティに関連するカスタマイズの結果を表示するには、以下のサンプル・コマンドを使用します。

- セキュリティーの設定とクラス
 - SETROPTS LIST
- 開始タスク
 - LISTGRP STCGROUP OMVS
 - LISTUSER STCDBM OMVS
 - RLIST STARTED DBGMGR.* ALL STDATA
- セキュアな z/OS UNIX サーバーとしてのデバッグ・マネージャー
 - RLIST FACILITY BPX.SERVER ALL
- デバッグ・マネージャーの MVS プログラム制御ライブラリー
 - RLIST PROGRAM ** ALL
- 統合デバッガーのアクセス
 - RLIST FACILITY AQE.** ALL
- データ・セット・プロファイル
 - LISTGRP FEL
 - LISTDSD PREFIX(FEL) ALL

第 10 章 マイグレーション・ガイド

マイグレーションに関する考慮事項

ここでは、本製品の以前のリリースと比較したインストールおよび構成上の変更点に重点を置いて説明します。また、このリリースへのマイグレーションに関する一般的なガイドラインも示します。詳しくは、本書の関連セクションを参照してください。

- 以前に IBM Rational Developer for z Systems を使用していた場合は、このバージョンの IBM Rational Developer for z Systems にアップグレードする前に、関連するカスタマイズ済みのファイルを保存してください。
- Developer for z Systems の複数インスタンスを実行する予定の場合は、「ホスト構成リファレンス」(SC43-2912) の『複数のインスタンスの実行』を参照してください。
- 3 つ以上離れたリリースにマイグレーションするシナリオの場合は、古い方のリリースは存在しないかのように、再びカスタマイズを行う必要があります。

前に構成したファイルのバックアップ

以前に Developer for z Systems を使用していた場合は、このバージョンの IBM Developer for z Systems をインストールする前に、関連するカスタマイズ済みのファイルを保存してください。

カスタマイズ可能な Developer for z Systems ファイルは、以下のロケーションにあります。

- すべての旧バージョン
 - FEK.SFEKSAMP。一部のメンバーは、FEKSETUP サンプル・ジョブによって FEK.#CUST.* にコピーされます。ここで、* は PARMLIB、PROCLIB、JCL、CNTL、ASM、および COBOL に相当します。
 - FEK.SFEKSAMV
 - /usr/lpp/rdz/samples/。一部のファイルは、FEKSETUP サンプル・ジョブによって /etc/rdz/ および /var/rdz/sc1mdt/* にコピーされます。ここで、* は CONFIG/、CONFIG/PROJECT/、および CONFIG/script/ に相当します。

以前の Developer for z Systems セットアップでは、他の製品によって所有されている構成ファイルへの変更も記載されています。

- バージョン 9.0.1 までの旧バージョン
 - SYS1.PARMLIB(IEASCVxx)
- すべての旧バージョン
 - SYS1.PARMLIB(BPXPRMxx)
 - SYS1.PARMLIB(COMMNDxx)
 - SYS1.PARMLIB(LPALSTxx)
 - SYS1.PARMLIB(PROGxx)

– (WLM)

DB2 ストアード・プロシージャのアプリケーション環境を割り当てます。

バージョン 9.5 からバージョン 9.5.1 へのマイグレーション

バージョン 9.5.1 では、RSE および JES ジョブ・モニターの関連機能が IBM Rational Developer for z Systems から別の製品 IBM Explorer for z/OS に移動されました。IBM Explorer for z/OS は、新たに IBM Rational Developer for z Systems の前提条件となりました。

新しい MVS データ・セット名および z/OS UNIX パスを確実に使用するために、新しい z/OS Explorer および Developer for z Systems サンプルを使用してすべてのカスタマイズ設定をやり直すことを、IBM は強くお勧めします。

SMP/E データ・セット:

- z/OS Explorer は、Developer for z Systems の SMP/E 前提条件です。つまり、z/OS Explorer および Developer for z Systems が同じ CSI にインストールされている必要があります。
- Developer for z Systems Host Utilities には SMP/E 関連の変更点はありません。

MVS データ・セット:

- z/OS Explorer は FEK 高位修飾子および SFEK* データ・セット名を使用します。ここではマイグレーション操作は必要ありません。
- Developer for z Systems は新しい高位修飾子 FEL および新しい SFEL* データ・セット名を使用します。FEK データ・セットへの現在の参照を確認し、必要に応じて代わりに FEL を使用するよう更新してください。
 - DATASET および PROGRAM クラス内のセキュリティー定義
 - 開始タスク、リモート・ビルド・プロシージャ、および zUnit 用の PROCLIB
 - CARMA を開始するための構成
 - CARMA VSAM を処理するための JCL
 - LE ユーザー出口を処理するための JCL
- Developer for z Systems Host Utilities は AKG 高位修飾子および SAKG* データ・セット名を使用します。マイグレーション操作は必要ありません。

z/OS UNIX パス:

- IBM Explorer for z/OS は、新しい z/OS UNIX パス /usr/lpp/ibm/zexpl、/etc/zexpl、および /var/zexpl を使用します。古いパス (/usr/lpp/rdz、/etc/rdz、および /var/rdz) への現在の参照を確認し、必要に応じて代わりに新しいパスを使用するよう更新してください。
 - 開始タスク用の PROCLIB
 - /etc/rdz への JCL 参照
 - /etc/rdz/rsed.envvars が /etc/zexpl/rse.env に置き換えられたことに注意してください

- ログ、クライアントへのプッシュ、および ISPF WORKAREA が /var/zexpl に移動したことに注意してください
- Developer for z Systems は、新しい z/OS UNIX パス /usr/lpp/ibm/rdz および /etc/zexpl を使用します。 /var/rdz パスも引き続き使用します。古いパス (/usr/lpp/rdz、/etc/rdz) への現在の参照を確認し、必要に応じて代わりに新しいパスを使用するよう更新してください。
 - サンプル zUnit 出力コンバーターを利用するための JCL
 - /etc/rdz/rsed.envvars が /etc/zexpl/rdz.env に置き換えられたことに注意してください
 - SCLMDT が /var/rdz/sclmdt に残っていることに注意してください
- Developer for z Systems Host Utilities は /usr/lpp/rdzutil パスを使用します。マイグレーション操作は必要ありません。

IBM Rational Developer for z Systems、FMID HHOP951

- SMP/E による MVS および z/OS UNIX のコンポーネントのデフォルトのインストール場所が FEK.SFEK* および /usr/lpp/rdz/* から FEL.SFEL* および /usr/lpp/ibm/rdz/* に変更されました。FEK 高位修飾子が IBM Explorer for z/OS によって使用されるようになりました。
- MVS および z/OS UNIX のコンポーネントのデフォルトのカスタマイズ場所が /etc/rdz から /etc/zexpl に変更されました。
- IBM Explorer for z/OS に移動された機能
 - ホスト構成ユーティリティーが IBM Explorer for z/OS に移動されました
 - RSE、JES ジョブ・モニター、およびレガシー ISPF ゲートウェイのセットアップ・アクティビティーは残っています
 - CARMA、統合デバッガー、および SCLMDT のセットアップ・アクティビティーは除去されています
 - JES ジョブ・モニターが IBM Explorer for z/OS に移動されました
 - CNTL(FEJTZO)
 - PROCLIB(JMON)
 - SAMPLIB(FEJCNFG)
 - クライアントへのプッシュが IBM Explorer for z/OS に移動されました
 - /etc/rdz/pushtoclient.properties が /etc/zexpl/pushtoclient.properties に移動されました
 - /etc/zexpl/pushtoclient.properties が /var/rdz から /var/zexpl に変更されました
 - RSE が IBM Explorer for z/OS に移動されました
 - PROCLIB(RSED)
 - /etc/rdz/rsed.envvars が /etc/zexpl/rse.env に移動され、名前変更されました
 - /etc/rdz/ISPF.conf が /etc/zexpl/ISPF.conf に移動されました
 - /etc/rdz/rsecomm.properties が /etc/zexpl/rsecomm.properties に移動されました

- /etc/rdz/ssl.properties が /etc/zexpl/ssl.properties に移動されました
- /etc/zexpl/rse.env には z/OS Explorer 機能のカスタマイズ定義のみが保持されます
- システム・ツールが IBM Explorer for z/OS に移動されました
- JCL(FEKRACF) には z/OS Explorer 機能のセキュリティー定義のみが保持されます
- JCL(FEKSETUP) には z/OS Explorer 機能のカスタマイズ定義のみが保持されます
- JCL(FEKSETUP) が usr/lpp/rdz から /usr/lpp/ibm/zexpl に、/etc/rdz から /etc/zexpl に、/var/rdz から /var/zexpl にそれぞれ変更されています
- 引き続き IBM Rational Developer for z Systems に残る機能
 - C/C++ コンテンツ・アシスト
 - /etc/rdz/include.conf が /etc/zexpl/include.conf に移動されました
 - CARMA
 - CNTL(CRASUBCA) が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
 - CNTL(CRASUBMT) が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
 - JCL(CRA\$VCAD) が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
 - JCL(CRA\$VCAS) が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
 - JCL(CRA\$VDEF) が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
 - JCL(CRA\$VMSG) が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
 - JCL(CRA\$VSTR) が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
 - JCL(CRA#CIRX) が FEK.* から FEL.* に変更されています
 - JCL(CRA#UADD) が FEK.* から FEL.* に変更されています
 - JCL(CRA#UQRY) が FEK.* から FEL.* に変更されています
 - JCL(CRA#VPDS) が FEK.* から FEL.* に変更されています
 - JCL(CRA#VSLM) が FEK.* から FEL.* に変更されています
 - /etc/rdz/CRASRV.properties が /etc/zexpl/CRASRV.properties に移動されました
 - /etc/rdz/crastart.conf が /etc/zexpl/crastart.conf に移動されました
 - /etc/rdz/crastart.endevor.conf が /etc/zexpl/crastart.endevor.conf に移動されました
 - /etc/zexpl/crastart.conf が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています

- /etc/zexpl/crastart.endevor.conf が FEK.* から FEL.* に、SFEK* から SFEL* に変更されています
- 統合デバッガー
 - JCL(AQED3CEE) が FEK.* から FEL.* に変更されています
- これにより、LE ユーザー出口のデフォルトの保管場所が更新されるため、ユーザーは LE ユーザー出口の使用方法を更新する必要がある場合があります
- JCL(AQED3CXT) が FEK.* から FEL.* に変更されています
- JCL(AQERACF) が FEK.SFEK* から FEL.SFEL* に変更されています
- PROCLIB(DBGMGR) が FEK.SFEK* から FEL.SFEL* に変更されています
- リモート・ビルド・プロシージャー
 - PROCLIB(ELAXFADT) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFASM) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFCOC) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFCOP) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFCOT) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFCP1) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFDCL) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFGO) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFLNK) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFPLP) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFPLT) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFPL1) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFPP1) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFTSO) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFUOP) が FEK から FEL (LODPRFX および STEPLIB) に変更
 - PROCLIB(ELAXFSP) が FEK.* から FEL.* に変更

- PROCLIB(ELAXFSQL) が FEK.* から FEL.* に変更
- SQL(FEKTEP2) が SQL(FELTEP2) に移動
- SQL(FEKTIAD) が SQL(FELTIAD) に移動
- システム・ツール
 - (新規) JCL(FELRACF) は Developer for z Systems 機能のセキュリティー定義を保持します
 - (新規) JCL(FELSETUP) は Developer for z Systems 機能のカスタマイズ定義を保持します
 - (新規) /etc/zexpl/rdz.env は、以前に /etc/rdz/rsed.envvars に配置されていた Developer for z Systems 機能のカスタマイズ定義を保持します
- zUnit
 - PROCLIB(AZUZUNIT) が FEK から FEL (LODPRFX および STEPLIB) に変更されています

場合によっては、新しい Developer for z Systems パス
 /usr/lpp/ibm/rdz/samples/zunit/* を使用するようプロシージャ呼び出しを更新する必要があります

構成可能なファイル

表 17 は、バージョン 9.5.1 でカスタマイズされる Developer for z Systems ファイルの概要を示しています。Developer for z Systems のサンプル・ライブラリー FEL.SFELSAMP、FEL.SFELSAMPV、および /usr/lpp/ibm/rdz/samples/ には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルの CARMA ソース・コードおよびそれらをコンパイルするジョブなど) が含まれています。

注: サンプル・ジョブ FELSETUP は、リストされているすべてのメンバーを別のデータ・セットとディレクトリー (デフォルトでは FEL.#CUST.* および /etc/zexpl/*) にコピーします。

表 17. バージョン 9.5.1 のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FELSETUP	FEL.SFELSAMP [FEL.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	新規 (構成が必要)
DBGMGR	FEL.SFELSAMP [FEL.#CUST.JCL]	デバッグ・マネージャー用の JCL	新しいデータ・セット/パス名を使用
AQEJCL	FEL.SFELSAMP [FEL.#CUST.PROCLIB(DBGMGR)]	DBGMGR メンバーの名前	DBGMBR メンバーを参照

表 17. バージョン 9.5.1 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
ELAXF*	FEL.SFELSAMP [FEL.#CUST.PROCLIB]	リモート・プロジェクト・ビルドなどのための JCL	新しいデータ・セット/パス名を使用
FELRACF	FEL.SFELSAMP [FEL.#CUST.JCL]	セキュリティ定義の JCL	新規 (構成が必要)
AQERACF	FEL.SFELSAMP [FEL.#CUST.JCL]	デバッグ・マネージャー用の JCL	新しいデータ・セット/パス名を使用
CRA\$VMSG	FEL.SFELSAMP [FEL.#CUST.JCL]	CARMA メッセージ VSAM を作成するための JCL	新しいデータ・セット/パス名を使用
CRA\$VDEF	FEL.SFELSAMP [FEL.#CUST.JCL]	CARMA 構成 VSAM を作成するための JCL	新しいデータ・セット/パス名を使用
CRA\$VSTR	FEL.SFELSAMP [FEL.#CUST.JCL]	CARMA カスタム情報 VSAM を作成するための JCL	新しいデータ・セット/パス名を使用
CRA\$VCAD	FEL.SFELSAMP [FEL.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA 構成 VSAM を作成するための JCL	新しいデータ・セット/パス名を使用
CRA\$VCAS	FEL.SFELSAMP [FEL.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA カスタム情報 VSAM を作成するための JCL	新しいデータ・セット/パス名を使用
CRASUBMT	FEL.SFELSAMP [FEL.#CUST.CNTL]	CARMA バッチ始動 CLIST	新しいデータ・セット/パス名を使用
CRASUBCA	FEL.SFELSAMP [FEL.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バッチ始動 CLIST	新しいデータ・セット/パス名を使用
CRACFG	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 対話構成	なし

表 17. バージョン 9.5.1 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRABCFG	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA バッ チ・アクション 構成	なし
CRABATCA	FEL.SFELSAMP [FEL.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バッ チ・アクション の JCL	なし
CRASCL	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	CA Endeavor® SCM 用テンプ レート SCL	なし
CRASHOW	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRATMAP	FEL.SFELSAMP [FEL.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRANDVRA	FEL.SFELPROC	CA Endeavor® SCM RAM 用の CARMA 割り振 り REXX	なし
CRADYNDA	FEL.SFELPROC	CA Endeavor® SCM RAM 用の CARMA 割り振 り REXX	なし
CRAALLOC	FEL.SFELPROC	CARMA 割り振 り REXX	なし
CRA#VSLM	FEL.SFELSAMP [FEL.#CUST.JCL]	SCLM RAM の メッセージ VSAM を作成す るための JCL	新しいデータ・ セット/パス名を 使用
CRA#ASLM	FEL.SFELSAMP [FEL.#CUST.JCL]	SCLM RAM の データ・セット を作成するた めの JCL	新しいデータ・ セット/パス名を 使用
CRA#VPDS	FEL.SFELSAMP [FEL.#CUST.JCL]	PDS RAM のメ ッセージ VSAM を作成するた めの JCL	新しいデータ・ セット/パス名を 使用
CRA#UADD	FEL.SFELSAMP [FEL.#CUST.JCL]	RAM 定義をマ ージするための JCL	新しいデータ・ セット/パス名を 使用

表 17. バージョン 9.5.1 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA#UQRY	FEL.SFELSAMP [FEL.#CUST.JCL]	RAM 定義を抽出するための JCL	新しいデータ・セット/パス名を使用
CRAXJCL	FEL.SFELSAMP [FEL.#CUST.ASM]	IRXJCL 置換用のサンプル・ソース・コード	なし
CRA#CIRX	FEL.SFELSAMP [FEL.#CUST.JCL]	CRAXJCL をコンパイルするための JCL	新しいデータ・セット/パス名を使用
AQECSD	FEL.SFELSAMP [FEL.#CUST.JCL]	統合デバッガーを CICS 領域に対して定義するための JCL	なし
AQED3CEE	FEL.SFELSAMP [FEL.#CUST.JCL]	カスタマイズされた LE ランタイム・モジュールを作成するための JCL	新しいデータ・セット/パス名を使用
AQED3CXT	FEL.SFELSAMP [FEL.#CUST.JCL]	LE ユーザー出口を作成するための JCL	新しいデータ・セット/パス名を使用
FEKTEP2	FEL.SFELSAMP [FEL.#CUST.SQL]	ELAXF* で使用される SQL コマンド入力	なし
FEKTIAD	FEL.SFELSAMP [FEL.#CUST.JCL]	ELAXF* で使用される SQL コマンド入力	なし
AZUZUNIT	FEL.SFELSAMP [FEL.#CUST.PROCLIB]	zUnit プロシージャの JCL	新しいデータ・セット/パス名を使用
FEKRNPLI	FEL.SFELSAMP [FEL.#CUST.CNTL]	プリプロセッサ・フレームワーク内部から PL/I コンパイラーを呼び出すための REXX	なし
IRZCSD	FEL.SFELSAMP [FEL.#CUST.JCL]	エンタープライズ・サービス・ツールを CICS に定義するための JCL	なし

表 17. バージョン 9.5.1 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
rdz.env	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	RSE 環境変数	古いコピーをこのコピーに置き換え、カスタマイズを再実行する必要がある。
CRASRV.properties	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	CARMA 構成ファイル	なし
crastart.conf	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	CRASTART を使用するための CARMA 構成ファイル	新しいデータ・セット/パス名を使用
crastart.endevor.conf	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	CA Endeavor® SCM RAM 用に CRASTART を使用するための CARMA 構成ファイル	新しいデータ・セット/パス名を使用
include.conf	/usr/lpp/ibm/rdz/samples/ [/etc/zexpl/]	C/C++ コンテンツ・アシスト用の強制インクルード	なし

表 18 は、バージョン 9.5.1 で z/OS Explorer に移動されたカスタマイズ可能なファイルの概要を示しています。

注: サンプル・ジョブ FEKSETUP は、リストされているすべてのメンバーを別のデータ・セットとディレクトリー (デフォルトでは FEK.#CUST.* および /etc/zexpl/*) にコピーします。

表 18. バージョン 9.5.1 z/OS Explorer のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	使用されなくなったファイルに対するアクションを削除し、新規ファイル用のアクションを追加するために更新
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JES ジョブ・モニター用の JCL	なし
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	JMON メンバーの名前	JMON メンバーを参照

表 18. バージョン 9.5.1 z/OS Explorer のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	RSE デーモンの JCL	なし
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	RSED メンバー の名称	RSED メンバー を参照
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	セキュリティ 定義の JCL	Developer for z Systems 固有の データが除去さ れました
FEJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES ジョブ・モ ニター構成ファ イル	なし
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	TSO 実行依頼用 の JCL	なし
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイル を収集するた めの JCL	新しいデータ・ セット/パス名を 使用 Developer for z Systems 固有の データが除去さ れました
rse.env	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	RSE 環境変数	rsed.envvars の 古いコピーをこ のコピーに置き 換え、再びカス タマイズする必 要があります。
ISPF.conf	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	ISPF ゲートウェ イ構成ファイル	新しいデータ・ セット/パス名を 使用
ssl.properties	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	RSE 暗号化通信 構成ファイル	なし
rsecomm.properties	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	RSE トレース構 成ファイル	なし
pushtoclient. properties	/usr/lpp/ibm/zexpl/samples/ [/etc/zexpl/]	クライアント構 成ファイルに情 報をブッシュす る	新しいデータ・ セット/パス名を 使用

IBM Rational Developer for z Systems Host Utilities、FMID HAKG951

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは変更されておらず、したがって AKG.* および /usr/lpp/rdutil/* のままです。

構成可能なファイル

表 19 は、バージョン 9.5.1 でカスタマイズされるファイルの概要を示しています。Developer for z Systems Host Utilities のサンプル・ライブラリーの AKG.SAKGSAMP および /usr/lpp/rdutil/samples には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルのコード・レビューの後処理スクリプトなど) が含まれます。

注: サンプル・ジョブ AKGSETUP は、リストされているすべてのメンバーを別のデータ・セット (デフォルトでは AKG.#CUST.*) にコピーします。

表 19. Host Utilities バージョン 9.0 のカスタマイズ

メンバーまたはファイル	デフォルト・ロケーション	目的	マイグレーションの注
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	データ・セットを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	なし
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・カバレッジの JCL	なし
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・レビューの JCL	なし
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	サード・パーティーのコードをコード・レビューに追加するための JCL	なし

バージョン 9.1 からバージョン 9.5 へのマイグレーション

これらの注は、基本バージョン 9.1 からバージョン 9.5 へのマイグレーションのためのものです。これには、バージョン 9.1 のメンテナンスの一部として既に文書化されている変更点も含まれています。メンテナンス・ストリームの一部である、したがって既に実装されている可能性がある、変更内容には、それが導入された時点のリリースでマークが付いています。

既存の rsed.envvars (デフォルトでは /etc/rdz 内にあります) を、新たに提供されたサンプル (デフォルトでは /usr/lpp/rdz/samples 内にあります) で置き換えて、カスタマイズをやり直すことをお勧めします。

IBM Rational Developer for z Systems、FMID HHOP950

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは、変更されておらず、FEK.* および /usr/lpp/rdz/* のままです。
- バージョン 9.5 以降
 - Application Deployment Manager は、製品から削除されました。以下の定義が削除されました。
 - SFEKLOAD(ADN*)
 - FEK.#CUST.JCL(ADN*)
 - FEK.#CUST.COBOLO(ADN*)
 - FEK.#CUST.ADNREPF0 [ADNVCRD で作成された VSAM]
 - FEK.#CUST.ADNMANF0 [ADNVMFST で作成された VSAM]
 - FEK.#CUST.ADN* データ・セット・セキュリティー・プロファイル [FEKRACF で実行]
 - CICS CSD [定義
ADNCSDAR、ADNCSDRS、ADNCSDTX、ADNCSDWS、ADNVMFST 、
AQECSD で実行]
 - usr/lpp/rdz/cicsts
 - CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRACFG は新規オプション DYNAMIC-VB-DATASET-ALLOC を定義します。
 - CRACFG は新規オプション DATASET-ALLOC-OVERRIDE を定義します。
 - エンタープライズ・サービス・ツール: 新規カスタマイズ可能メンバー
 - IRZCSD: CICS 領域 CSD 更新
 - 統合デバッガー: CICS トランザクションのデバッグに IP CICS ソケットが必要になりました。
 - PL/I: 新規カスタマイズ可能メンバー
 - FEKRNPLI: PL/I プリプロセッサ REXX
 - リモート・ビルド・プロシージャ: 以下のカスタマイズ可能メンバーが変更されました。
 - ELAXFDCL: SFEKAUTH を 統合デバッガーの STEPLIB に追加
 - ELAXFTSO: SFEKAUTH を統合デバッガーの STEPLIB に追加
 - RSE: 以下の新しいオペレーター・コマンドが追加されました。
 - F rsed,APPL=SEND 'message',[USER={ userid | (userid,userid,...)}]
 - F rsed,APPL=SEND 'message',[PID={ pid | (pid,pid,...)}]
 - F rsed,APPL=SEND CLEARALL
 - RSE: オプション・ディレクティブが rsed.envvars に追加されました:
 - CGI_CEATSO
 - CGI_CEATSO_KEEPAIVE
 - GSK_PROTOCOL_SSLV3
 - GSK_V3_CIPHER_SPECS

- RSE: オプション・ディレクティブが `rsed.envvars` から削除されました:
 - `DSTORE_SSL_ALGORITHM`
- RSE: `rsed.envvars` 内で、以下の必須ディレクティブに編集不能というマークが付けられています。
 - `_RSE_HOST_CODEPAGE`
- RSE: オペレーター・コマンド、`z/OS UNIX` コマンド、または `TSO SEND` を使用してクライアントにメッセージを送信するためのサポートが追加されました。
- RSE: 対話式 `ISPF` ゲートウェイのサポートが追加されました。
- `zUnit`: 以下のカスタマイズ可能メンバーが変更されました。
 - `AZUZUNIT`: コード・カバレッジ・サポートが追加されました。
- バージョン 9.1.1.1 以降
 - エラー・フィードバック: `FEK.SFEKPROC(FEKFERRF)` が `/usr/lpp/rdz/bin/fekferrf.rex` に移動されました。
 - `CARMA`: 以下の新しいカスタマイズ可能メンバーが追加されました。
 - `CRASCL`: `CA Endevor® SCM` 用サンプル `CARMA` テンプレート `SCL`。
 - `CARMA`: 以下のカスタマイズ可能メンバーが変更されました。
 - `CRACFG` は新規オプション `PACKAGE-EDITING-OPTION` を定義します。
 - `CRACFG` は新規オプション `SCL-REQUIRED` を定義します。
 - `CRACFG` は新規オプション `SCL-DATASET-TEMPLATE` を定義します。
 - 統合デバッガー: 以下のカスタマイズ可能メンバーが変更されました。
 - `LE` ユーザー出口用の `AQED3CXT` 新規デバッグ・レベル
 - `AQED3CEE` により正しいユーザー出口が `LE` ランタイムに追加されるようになりました。
 - RSE: オプションの `LPA` モジュールは必須になります。
 - `FEKDSI` が `LPA/LINKLIST/STEPLIB` を介して使用可能である必要があります。
 - RSE: 以下の新しいオペレーター・コマンドが追加されました。
 - `F rsed,APPL=DEBUG {PROCESS | P},{STORAGE | STOR}[,PID=pid]`
 - `F rsed,APPL=DEBUG {DAEMON | D},{STORAGE | STOR}`
 - RSE: 製品コード 5697-CDT の `IBM Enterprise COBOL Suite for z/OS` の一部としての購入時に製品登録が必要です。
 - `SYS1.PARMLIB`: 以下の新規 `PARMLIB` メンバーが更新されています。
 - `IFAPRDxx`: 製品登録
- バージョン 9.1.1.0 以降
 - `CARMA`: `CA Endevor® SCM RAM` 用の `CRADEF` ファイルおよび `CRASTRS VSAM` ファイルが更新されました。
 - `CARMA`: 以下のカスタマイズ可能メンバーが変更されました。
 - `CRACFG` は新規オプション `ALTERNATIVE-ALLOC` を定義します。
 - `CRABCFG` は新規アクション `TRANSFER-ELEMENT` を定義します。
 - `CRASUBCA` に `APIHJC` が追加されました。

- crastart.endevor.conf に APIHJC が追加されました。
- 統合デバッガー: IBM Debug Tool for z/OS に対する共存サポート
- 統合デバッガー: 以下のカスタマイズ可能メンバーが変更されました。
 - AQECSD (CICS CSD 更新) は新規リソースを定義します。
 - AQERACF は新規 AQE.** プロファイルを定義します。
 - FEKRACF は新規 AQE.** プロファイルを定義します。
 - 統合デバッガー: 必須 CICS CSD 定義が追加されました。
 - AQEM および AQED 一時データ・キュー
 - プログラム
 - AQEW3Z3、AQEW3Z6、AQEL3Z3、AQEL3Z6、AQEL6Z3、AQEL6Z6
 - 統合デバッガー: 一部の CICS CSD 定義が削除されました。
 - CIGZ および CIBM 一時データ・キュー
 - プログラム AQEEV006
- 統合デバッガー: 以下のカスタマイズ可能メンバーが変更されました。
 - AQESTC [DBGMGR] 開始タスクの STEPLIB に SFEKLPA が加わりました
- 統合デバッガー: 必須 SVC が追加されました。
 - IEASVCxx parmlib が AQESVC03 用に変更されました (動的にロード可能)
- 統合デバッガー: オプションの SVC が削除されました。
 - IEASVCxx parmlib が AQESVC01 用に変更されました。
- 統合デバッガー: 必須 RACF プロファイルが追加されました。
 - AQE.AUTHDEBUG.STDPGM
 - AQE.AUTHDEBUG.AUTHPGM
- 統合デバッガー: オプション RACF プロファイルが削除されました。
 - AQE.AUTHDEBUG.WRITEBUFFER
- ログの収集: クライアントがホスト・ログを収集できます
- RSE: パスフレーズ・サポートが追加されました。
- RSE: オプション・ディレクティブが rsed.envvars に追加されました:
 - search.server.limit.scanned_objects
 - search.server.limit.errcount
- RSE: オプション・ディレクティブが rsed.envvars から削除されました:
 - enable.saf.check
 - RSE_DSICALL
 - search.server.limit.datasets

構成可能なファイル

130 ページの表 20 は、バージョン 9.5 でカスタマイズされるファイルの概要を示しています。Developer for z Systems のサンプル・ライブラリー FEK.SFEKSAMP、FEK.SFEKSAMV、および /usr/lpp/rdz/samples/ には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルの CARMA ソース・コードおよびそれらをコンパイルするジョブなど) が含まれています。

注: サンプル・ジョブ FEKSETUP は、リストされているすべてのメンバーを別のデータ・セットおよびディレクトリー (デフォルトでは FEK.#CUST.* および /etc/rdz/*) にコピーします。

表 20. バージョン 9.5 のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	使用されなくなったファイルに対するアクションを削除し、新規ファイル用のアクションを追加するために更新
JMON	FEK.SFEKSAMP (FEJJJCL) [FEK.#CUST.PROCLIB]	JES ジョブ・モニター用の JCL	なし
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (JMON)]	JMON メンバーの名前	JMON メンバーを参照
RSED	FEK.SFEKSAMP (FEKRSED) [FEK.#CUST.PROCLIB]	RSE デーモンの JCL	なし
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (RSED)]	RSED メンバーの名前	RSED メンバーを参照
DBGMGR	FEK.SFEKSAMP [FEK.#CUST.JCL]	デバッグ・マネージャー用の JCL	STEPLIB に SFEKLPA が追加された
AQEJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB (DBGMGR)]	DBGMGR メンバーの名前	DBGMBR メンバーを参照
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	リモート・プロジェクト・ビルドなどのための JCL	STEPLIB に SFEKAUTH が追加された
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	セキュリティ定義の JCL	新規 AQE.** プロファイル
AQERACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	デバッグ・マネージャー用の JCL	新規 AQE.** プロファイル
FEKPBITS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルのアクセス許可を変更するための JCL	なし
FEJJCNGF	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES ジョブ・モニター構成ファイル	なし
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	TSO 実行依頼用の JCL	なし

表 20. バージョン 9.5 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA メッセージ VSAM を作成するための JCL	なし
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA 構成 VSAM を作成するための JCL	なし
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA カスタム情報 VSAM を作成するための JCL	なし
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA 構成 VSAM を作成するための JCL	VSAM 入力に変更された
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA カスタム情報 VSAM を作成するための JCL	VSAM 入力に変更された
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA バッチ始動 CLIST	なし
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バッチ始動 CLIST	DD APIHJC および新規 DD が追加された
CRACFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 対話構成	オプションのディレクティブが追加された
CRBCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA バッチ・アクション構成	オプションのディレクティブが追加された

表 20. バージョン 9.5 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRBATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA パッチ・アクションの JCL	なし
CRASCL	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM 用テンプレート SCL	新規、構成はオプション
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRANDVRA	FEK.SFEKPROC	CA Endeavor® SCM RAM 用の CARMA 割り振り REXX	新規 DD の割り振り
CRADYNDA	FEK.SFEKPROC	CA Endeavor® SCM RAM 用の CARMA 割り振り REXX	新規、構成はオプション
CRAALLOC	FEK.SFEKPROC	CARMA 割り振り REXX	なし
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のメッセージ VSAM を作成するための JCL	なし
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のデータ・セットを作成するための JCL	なし
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	PDS RAM の メッセージ VSAM を作成するための JCL	なし
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を マージするための JCL	なし

表 20. バージョン 9.5 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を抽出するための JCL	なし
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	IRXJCL 置換用のサンプル・ソース・コード	なし
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRAXJCL をコンパイルするための JCL	なし
AQECSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	統合デバッガーを CICS 領域に対して定義するための JCL	新規リソースを定義する
AQED3CEE	FEK.SFEKSAMP [FEK.#CUST.JCL]	カスタマイズされた LE ランタイム・モジュールを作成するための JCL	正しい LE 出口が使用されるように更新された
AQED3CXT	FEK.SFEKSAMP [FEK.#CUST.JCL]	LE ユーザー出口を作成するための JCL	LE 出口用の新規デバッグ・レベル
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	ELAXF* で使用される SQL コマンド入力	なし
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	ELAXF* で使用される SQL コマンド入力	なし
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	zUnit プロシージャの JCL	STEPLIB に SFEKAUTH が追加された
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	プリプロセッサ・フレームワーク内部から PL/I コンパイラーを呼び出すための REXX	なし

表 20. バージョン 9.5 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
IRZCSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	エンタープライズ・サービス・ツールを CICS に定義するための JCL	新規、構成はオプション
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルを収集するための JCL	なし
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 環境変数	古いコピーをこのコピーに置き換え、カスタマイズを再実行する必要がある。
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF クライアント・ゲートウェイ構成ファイル	なし
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA 構成ファイル	なし
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CRASTART を使用するための CARMA 構成ファイル	なし
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CA Endeavor® SCM RAM 用に CRASTART を使用するための CARMA 構成ファイル	DD APIHJC および新規 DD が追加された
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	C/C++ コンテンツ・アシスト用の強制インクルード	なし
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 暗号化通信構成ファイル	なし
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE トレース構成ファイル	なし
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	クライアント構成ファイルに情報をプッシュする	なし

IBM Rational Developer for z Systems Host Utilities、FMID HAKG950

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは変更されておらず、したがって AKG.* および /usr/lpp/rdzutil/* のままです。
- バージョン 9.1.1.1 以降
 - コード・カバレッジ: 製品コード 5697-CDT の IBM Enterprise COBOL Suite for z/OS の一部としての購入時に必要な製品登録
 - コード・レビュー: 製品コード 5697-CDT の IBM Enterprise COBOL Suite for z/OS の一部としての購入時に必要な製品登録
- バージョン 9.1.1.0 以降
 - コード・レビュー: AKGCR プロシーチャーの新規オプション
 - SYSLIB: クライアントからインポートされたデータを必要とする PROPERTY が置き換えられました。
 - LIST: データ・セット/メンバーと、それぞれのファイル・タイプのリスト (PDS、 MEMBERS、 および EXTMAP が置き換えられました)
 - JUNIT: 新規出力形式

構成可能なファイル

表 21 は、バージョン 9.5 でカスタマイズされるファイルの概要を示しています。Developer for z Systems Host Utilities のサンプル・ライブラリーの AKG.SAKGSAMP および /usr/lpp/rdzutil/samples には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルのコード・レビューの後処理スクリプトなど) が含まれます。

注: サンプル・ジョブ AKGSETUP は、リストされているすべてのメンバーを別のデータ・セット (デフォルトでは AKG.#CUST.*) にコピーします。

表 21. Host Utilities バージョン 9.0 のカスタマイズ

メンバーまたはファイル	デフォルト・ロケーション	目的	マイグレーションの注
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	データ・セットを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	なし
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・カバレッジの JCL	なし
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・レビューの JCL	新規 DDs SYSLIB、LIST、および JUNIT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	サード・パーティーのコードをコード・レビューに追加するための JCL	なし

バージョン 9.1 のマイグレーションに関する注意事項

以下のマイグレーションに関する注意事項は、IBM Rational Developer for z Systems バージョン 9.1 に固有のものであります。これらの注意事項は、IBM Rational Developer for z Systems バージョン 9.1.0 からバージョン 9.1.1 へのマイグレーションに該当し、バージョン 9.1.0 のマイグレーションに関する既存の注意事項を補足するものです。

記載している変更内容はすべて、バージョン 9.1.1 以降に該当します。

IBM Rational Developer for z Systems、FMID HHOP910

- CARMA: CA Endeavor® SCM RAM 用の CRADEF ファイルおよび CRASTRS VSAM ファイルが更新されました。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRACFG は新規オプション ALTERNATIVE-ALLOC を定義します。
 - CRABCFG は新規アクション TRANSFER-ELEMENT を定義します。
 - CRASUBCA に APIHJC が追加されました。
 - crastart.endevor.conf に APIHJC が追加されました。
- 統合デバッガー: IBM Debug Tool for z/OS に対する共存サポート
- 統合デバッガー: 以下のカスタマイズ可能メンバーが変更されました。
 - DBGMGR 開始タスクは SVC 番号を使用しなくなりました。
 - AQECSD (CICS CSD 更新) は新規リソースを定義します。
 - AQERACF は新規 AQE.** プロファイルを定義します。
 - FEKRACF は新規 AQE.** プロファイルを定義します。
 - 統合デバッガー: 必須 CICS CSD 定義が追加されました。
 - AQEM および AQED 一時データ・キュー
 - プログラム AQEW3Z3、AQEW3Z6、AQEL3Z3、AQEL3Z6、AQEL6Z3、AQEL6Z6
 - 統合デバッガー: 一部の CICS CSD 定義が削除されました。
 - CIGZ および CIBM 一時データ・キュー
 - プログラム AQEEV006
- 統合デバッガー: 必須 SVC が追加されました。
 - IEASVCxx parmlib が AQESVC03 用に変更されました。
- 統合デバッガー: オプションの SVC が削除されました。
 - IEASVCxx parmlib が AQESVC01 用に変更されました。
- 統合デバッガー: 必須 RACF プロファイルが追加されました。
 - AQE.AUTHDEBUG.STDPGM
 - AQE.AUTHDEBUG.AUTHPGM
- 統合デバッガー: オプション RACF プロファイルが削除されました。
 - AQE.AUTHDEBUG.WRITEBUFFER
- RSE: パスフレーズ・サポートが追加されました。
- RSE: オプション・ディレクティブが rsed.envvars に追加されました:

- search.server.limit.scanned_objects
- search.server.limit.errcount
- RSE: オプション・ディレクティブが rsed.envvars から削除されました:
 - enable.saf.check
 - RSE_DSICALL
 - search.server.limit.datasets

IBM Rational Developer for z Systems Host Utilities、FMID HAKG910

- コード・レビュー: AKGCR プロシーチャーの新規オプション
 - SYSLIB: クライアントからインポートされたデータを必要とする PROPERTY が置き換えられました。
 - LIST: データ・セット/メンバーと、それぞれのファイル・タイプのリスト (PDS、 MEMBERS、 および EXTMAP が置き換えられました)
 - JUNIT: 新規出力形式

バージョン 9.0 からバージョン 9.1 へのマイグレーション

これらの注は、基本バージョン 9.0 からバージョン 9.1 へのマイグレーションのためのものです。これには、バージョン 9.0 のメンテナンスの一部として既に文書化されている変更点も含まれています。メンテナンス・ストリームの一部である、したがって既に実装されている可能性がある、変更内容には、それが導入された時点のリリースでマークが付いています。

既存の rsed.envvars (デフォルトでは /etc/rdz 内にあります) を、新たに提供されたサンプル (デフォルトでは /usr/lpp/rdz/samples 内にあります) で置き換えて、カスタマイズをやり直すことをお勧めします。

IBM Rational Developer for z Systems、FMID HHOP910

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは、変更されておらず、FEK.* および /usr/lpp/rdz/* のままです。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRASUBMT
 - CRASUBCA
- カスタマイズ: FEKSETUP JCL が新規メンバーを処理するようになりました。
 - AQED3CEE: FEK.#CUST.JCL(AQED3CEE) にコピーされる
 - AQED3CXT: FEK.#CUST.JCL(AQED3CXT) にコピーされる
 - FEKPBITS: FEK.#CUST.JCL(FEKPBITS) にコピーされる
- RSE: 以下の新しいオプション・ディレクティブが rsed.envvars に追加されました。
 - (_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: 以下の新しいオペレーター・コマンドが追加されました。
 - F rsed,APPL=LOG {USER | AUDIT | NOSERVER | OWNER}
- RSE: 以下の新しいコンソール・メッセージが追加されました。
 - FEK220I = ホスト・ログは {0} に書き込まれます (Host logs are written onto {0})
 - FEK221E = {0} は {1} が原因で中断しました ({0} was interrupted because of {1})
 - FEK301E = {0} (uid:{1}) は {2} (file_owner uid:{3}) のディレクトリーを所有しません ({0} (uid:{1}) does not own the directory of {2} (file_owner uid:{3}))
 - FEK302E = LOGS コマンドの要求者 {0} には、{1} のプロファイルにアクセスする権限がありません (The requester, {0}, of the LOGS command does not have authority to access the profile of {1})
 - FEK303E = シンボリック・リンク {0} はログ・ディレクトリーとして使用できません (The symbolic link, {0}, cannot be used as a log directory)
 - FEK304W = 無効な {0}、{1} が指定されました。代わりに、デフォルト・モード {3} が使用されます。 (Invalid {0}, {1}, was specified. The default mode, {3}, is used instead.)
 - FEK305E = ID {0} は、{1} にアクセスするための適切な特権を保持していません。 (The ID, {0}, does not have appropriate privileges to access {1}.)
- セキュリティー: 以下のセキュア・ログ・ファイルに対するサポートが追加されました。
 - FEKPBITS: 既存のログ・インフラストラクチャーの許可を変更するためのスクリプト
 - 新規動作: サーバー・ログが \$daemon.log/server に配置されるようになりました。
 - 新規動作: ログは制限付きアクセス許可で作成されます。
- セキュリティー: 以下の新しいセキュリティー・プロファイルのサポートが追加されました。
 - FEK.CMD.LOGS.**
 - FEK.REJECT.*.UPDATES.system.group
- バージョン 9.0.1.1 以降
 - RSE: 以下の新しいオプション・ディレクティブが rsed.envvars に追加されました。
 - (_RSE_JAVAOPTS) -Ddebug.miner.autoreconnect
- バージョン 9.0.1 以降
 - CARMA: CA Endeavor® SCM RAM 用の CRADEF VSAM ファイルが更新されました。

- CARMA: LPA 内にある CRAFTSTART ロード・モジュールは更新されており、LPA の更新が必要です。
- CARMA: CARMA 始動時にユーザー出口を実行するためのサポートが追加されました。
- CARMA: RAM 処理始動引数のサポートが追加されました。
- CARMA: 以下の新しいカスタマイズ可能メンバーが追加されました。
 - CRAEXIT: サンプル CARMA ユーザー出口。
 - CRAALLOC: カスタム RAM CARMA 呼び出し用の割り振り exec。
 - CRACFG: CA Endeavor® SCM RAM 使用法構成ファイル。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRASRV.properties
 - crastart.conf
 - crastart.endevor.conf
 - CRASUBMT
 - CRASUBCA
 - CRANDVRA
- CARMA: crastart.endevor.conf および CRASUBCA の CA Endeavor® SCM RAM に対して DD ステートメントがさらに追加されました。
 - CRAPARM (CRANDVRA により割り振られる)
 - CRACFG
- CARMA: crastart.conf および CRASUBMT の CA Endeavor® SCM RAM 以外のものに対して DD ステートメントがさらに追加されました。
 - CRAPARM (CRAALLOC により割り振られる)
- カスタマイズ: FEKSETUP JCL が新規メンバーを処理するようになりました。
 - CRACFG: FEK.#CUST.PARMLIB(CRACFG) にコピーされる
 - AQEJCL: FEK.#CUST.PROCLIB(DBGMGR) にコピーされる
 - AQECSD: FEK.#CUST.JCL(AQECSD) にコピーされる
- 統合デバッガー: 新しいオプション・サービス
 - IEASVCxx、LPALSTxx、および PROGxx (APF および LINKLIST) PARMLIB 更新
 - DBGMGR: 開始タスク JCL
 - AQECSD: CICS CSD を更新するサンプル JCL
 - AQERACF: 統合デバッガーのみにセキュリティ・セットアップを実行するサンプル JCL
- RSE: 更新された PROCLIB メンバー
 - ELAXFGO
- RSE: 以下の新しいオプション・ディレクティブが rsecomm.properties に追加されました。
 - USER
- RSE: 新しいオペレーター・コマンド
 - F rsed,APPL=TRACE {USER | SERVER | CLEAR}

- RSE: 以下の新しいオプション・ディレクティブが `rsed.envvars` に追加されました。
 - (`_RSE_JAVAOPTS`) `-Dsearch.server.limit.timeout`
 - (`_RSE_JAVAOPTS`) `-Dkeep.all.logs`
 - (`_RSE_JAVAOPTS`) `-Daudit.users`
 - `RSE_UBLD_DD`
 - `RSE_UBLD_STEPLIB`
- RSE: 以下の新しいコンソール・メッセージが追加されました。
 - `FEK910I = FEK107E = {0}` のディスクがほとんど満杯です。既存の `{1}` が削除されました
- zUnit: 以下の新しいオプションの始動引数が追加されました。
 - `CLOCALE / -l`

注: 統合デバッガーがない状態の、既存の Developer for z Systems セットアップからのマイグレーションを単純化するために、RACF コマンドを伴うサンプル JCL `FEK.SFEKSAMP(AQERACF)` が提供されています。これは統合デバッガーに関連するセキュリティー定義だけを定義します。

構成可能なファイル

表 22 は、バージョン 9.1.0 でカスタマイズされるファイルの概要を示しています。Developer for z Systems のサンプル・ライブラリー `FEK.SFEKSAMP`、`FEK.SFEKSAMV`、および `/usr/lpp/rdz/samples/` には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルの `CARMA` ソース・コードおよびそれらをコンパイルするジョブなど) が含まれています。

注: サンプル・ジョブ `FEKSETUP` は、リストされているすべてのメンバーを別のデータ・セットおよびディレクトリー (デフォルトでは `FEK.#CUST.*` および `/etc/rdz/*`) にコピーします。

表 22. バージョン 9.1.0 のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	使用されなくなったファイルに対するアクションを削除し、新規ファイル用のアクションを追加するために更新
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JES ジョブ・モニター用の JCL	なし
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	JMON メンバーの名前	JMON メンバーを参照

表 22. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	RSE デーモンの JCL	なし
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	RSED メンバーの名前	RSED メンバーを参照
DBGMGR	FEK.SFEKSAMP [FEK.#CUST.JCL]	デバッグ・マネージャー用の JCL	新規、カスタマイズはオプション
AQEJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(DBGMGR)]	DBGMGR メンバーの名前	DBGMBR メンバーを参照
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	リモート・プロジェクト・ビルドなどのための JCL	ELAXFGO には STEPLIB の SFEKAUTH が含まれる
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	セキュリティ定義の JCL	新規、開始タスク DBGMGR
AQERACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	デバッグ・マネージャー用の JCL	新規、カスタマイズはオプション
FEKPBITS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルのアクセス許可を変更するための JCL	新規、カスタマイズはオプション
FEJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES ジョブ・モニター構成ファイル	なし
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	TSO 実行依頼用の JCL	なし
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA メッセージ VSAM を作成するための JCL	なし
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA 構成 VSAM を作成するための JCL	なし
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA カスタム情報 VSAM を作成するための JCL	なし

表 22. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用 に CARMA 構 成 VSAM を 作成するた めの JCL	VSAM 入力 が変更され た
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用 に CARMA カ スタム情報 VSAM を作 成するための JCL	VSAM 入力 が変更され た
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA バッ チ始動 CLIST	8 つの始動 引数、DD CRAPARM お よび割り振 り exec のサ ポートを追 加
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用 の CARMA バ ッチ始動 CLIST	8 つの始動 引数、DD CRAPARM お よび新規 DD のサポート を追加
CRACFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用 の CARMA 対 話構成	新規、構成 はオプション
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用 の CARMA バ ッチ・アクシ ョン構成	なし
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用 の CARMA バ ッチ・アクシ ョンの JCL	なし
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用 の CARMA 構 成	なし

表 22. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構成	なし
CRANDVRA	FEK.SFEKPROC	CA Endeavor® SCM RAM 用の CARMA 割り振り REXX	ユーザー出口のサポ ートが追加され、新 規 DD を割り振る
CRAALLOC	FEK.SFEKPROC	CARMA 割り 振り REXX	新規、構成は オプション
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のメッセージ VSAM を作成 するための JCL	なし
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のデータ・セ ットを作成す るための JCL	なし
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	PDS RAM の メッセージ VSAM を作成 するための JCL	なし
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を マージするた めの JCL	なし
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を 抽出するた めの JCL	なし
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	IRXJCL 置換 用のサンプ ル・ソース・ コード	なし
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRAXJCL を コンパイルす るための JCL	なし
AQECSD	FEK.SFEKSAMP [FEK.#CUST.JCL]	統合デバッ ガーを CICS 領 域に対して定 義するための JCL	新規、構成は オプション

表 22. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
AQED3CEE	FEK.SFEKSAMP [FEK.#CUST.JCL]	カスタマイズされた LE ランタイム・モジュールを作成するための JCL	新規、構成はオプション
AQED3CXT	FEK.SFEKSAMP [FEK.#CUST.JCL]	LE ユーザー出口を作成するための JCL	新規、構成はオプション
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	RESTful CRD サーバーを主 CICS 領域に対して定義するための JCL	なし
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザクション ID を CICS 領域に対して定義するための JCL	なし
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザクション ID を作成するための JCL	なし
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	ADNMSGHS をコンパイルするための JCL	なし
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	パイプライン・メッセージ・ハンドラー用のサンプル・ソース・コード	なし
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD リポジトリを作成するための JCL	なし
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Web サービス CRD サーバーを主 CICS 領域に対して定義するための JCL	なし

表 22. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD サーバーを非主 CICS 領域に対して定義するための JCL	なし
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD のデフォルトを更新するための JCL	なし
ADNVMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	マニフェスト・リポジトリを作成し、定義するための JCL	なし
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	ELAXF* で使用される SQL コマンド入力	なし
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	ELAXF* で使用される SQL コマンド入力	なし
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	zUnit プロシージャの JCL	なし
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	プリプロセッサ・フレームワーク内部から PL/I コンパイラーを呼び出すための REXX	なし
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルを収集するための JCL	なし
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 環境変数	古いコピーをこのコピーに置き換え、カスタマイズを再実行する必要がある。
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF クライアント・ゲートウェイ構成ファイル	なし
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA 構成ファイル	ユーザー出口のサポートを追加

表 22. バージョン 9.1.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CRASTART を使用するた めの CARMA 構成ファイル	8 つの始動引 数、DD CRAPARM お よび割り振り exec のサポー トを追加
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CA Endevor® SCM RAM 用 に CRASTART を使用するた めの CARMA 構成ファイル	8 つの始動引 数、DD CRAPARM お よび新規 DD のサポートを 追加
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	C/C++ コンテ ンツ・アシス ト用の強制イ ンクルード	なし
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 暗号化通 信構成ファイ ル	なし
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE トレース 構成ファイル	USER キーワ ードのサポー トを追加
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	クライアント 構成ファイ ルに情報をプッ シュする	なし

IBM Rational Developer for z Systems Host Utilities、FMID HAKG910

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは変更されておらず、したがって AKG.* および /usr/lpp/rdzutil/* のままです。
- コード・レビュー: 追加の DD ステートメントが AKGCR に追加されました。
 - BEXPORT
 - BIMPORT

構成可能なファイル

147 ページの表 23 は、バージョン 9.1 でカスタマイズされるファイルの概要を示しています。Developer for z Systems Host Utilities のサンプル・ライブラリーの AKG.SAKGSAMP および /usr/lpp/rdzutil/samples には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルのコード・レビューの後処理スクリプトなど) が含まれます。

注: サンプル・ジョブ AKGSETUP は、リストされているすべてのメンバーを別のデータ・セット (デフォルトでは AKG.#CUST.*) にコピーします。

表 23. Host Utilities バージョン 9.0 のカスタマイズ

メンバーまたはファイル	デフォルト・ロケーション	目的	マイグレーションの注
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	データ・セットを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	なし
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・カバレッジの JCL	なし
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・レビューの JCL	新規 DD BIMPORT および BEXPORT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	サード・パーティーのコードをコード・レビューに追加するための JCL	なし

バージョン 8.5 からバージョン 9.0 へのマイグレーション

これらの注は、基本バージョン 8.5 からバージョン 9.0 へのマイグレーションのためのものです。これには、バージョン 8.5 のメンテナンスの一部として既に文書化されている変更点も含まれています。メンテナンス・ストリームの一部である、したがって既に実装されている可能性がある、変更内容には、それが導入された時点のリリースでマークが付いています。

IBM Rational Developer for z Systems、FMID HHOP900

- SMP/E による MVS および z/OS UNIX コンポーネントのデフォルトのインストール・ロケーションは、変更されておらず、したがって FEK.* および /usr/lpp/rdz/* のままです。
- CARMA: カスタマイズ可能な CA Endeavor® SCM バッチ・アクション (バージョン 8.5.1 より) に対する新しいサポートを使用するには、CA Endeavor® SCM RAM 用の CRADEF ファイルおよび CRASTRS VSAM ファイルを更新する必要があります。
- CARMA: CRADEF VSAM 作成時に RAM を使用不可にするサポートが追加されました (バージョン 8.5.1 以降)。
- CARMA: CRASRV.properties の相対ファイル参照のサポートが追加されました (バージョン 8.5.1 以降)。
- CARMA: 新しいサンプル・メンバーが追加されました。
 - CRABJOBC: CA Endeavor® SCM バッチ・アクション用のデフォルト JOB カード (バージョン 8.5.1 より)。
- CARMA: 以下のカスタマイズ可能メンバーが変更されました。
 - CRASRV.properties (バージョン 8.5.1 以降)
 - carma.startup.rex (バージョン 8.5.1 以降)

- CR\$VCAD (バージョン 8.5.1 以降)
- CR\$VDEF (バージョン 8.5.1 以降)
- CRABATCA (バージョン 8.5.1 以降)
- CRABCFG (バージョン 8.5.1 以降)
- CRANDVRA (バージョン 8.5.1 以降)
- CARMA: crastart.endevor.conf および CRASUBCA の CA Endeavor® SCM RAM に対して DD ステートメントがさらに追加されました。
 - CRABJCLO (CRANDVRA により割り振られる) (バージョン 8.5.1 以降)
 - ENHCEDIT (CRANDVRA により割り振られる) (バージョン 8.5.1 以降)
- カスタマイズ: FEKSETUP JCL が新規メンバーを処理するようになりました。
 - CRABJOB: FEK.#CUST.CNTL(CRABJOB) にコピーされます (バージョン 8.5.1 以降)
 - ELAXFSP: FEK.#CUST.PROCLIB(ELAXFSP) にコピーされます (バージョン 9.0 以降)
 - ELAXFSQL: FEK.#CUST.PROCLIB(ELAXFSQL) にコピーされます (バージョン 9.0 以降)
 - FEKTEP2: FEK.#CUST.SQL(FEKTEP2) にコピーされます (バージョン 9.0 以降)
 - FEKTIAD: FEK.#CUST.SQL(FEKTEP2) にコピーされます (バージョン 9.0 以降)
- Fault Analyzer 統合: FAI のサポートは廃止されました。この変更は、まだ FAI を使用している古いクライアントとは非互換です。
- JES ジョブ・モニター - JMON 開始タスクに対して新しいオペレーター・コマンドが追加されました。
 - MODIFY USERS (バージョン 8.5.1 以降)
 - MODIFY -T{N | E | I | V} (バージョン 8.5.1 以降)
 - MODIFY -M{N | E | W | I | V} (バージョン 8.5.1 以降)
 - MODIFY TRACE {N | E | I | V} (バージョン 9.0 以降)
 - MODIFY MESSAGE {N | E | W | I | V} (バージョン 9.0 以降)
- JES ジョブ・モニター - 以下の新しいオプション・ディレクティブが FEJCNFG に追加されました。
 - LOOPBACK_ONLY (バージョン 9.0 以降)
- JES ジョブ・モニター - 以下のオプション・ディレクティブが FEJCNFG から削除されました。
 - _BPXK_SETIBMOPT_TRANSPORT (バージョン 9.0 以降)
- 問題判別: FEKLOGS JCL は、ユーザー・ログを収集するために、複数のユーザー ID の指定をサポートするようになりました (バージョン 8.5.1 より)。
- 問題判別: FEKLOGS JCL は DD REFORMAT を使用して再フォーマットされたログを収集し、より迅速に問題を判別します (バージョン 8.5.1 より)。
- 問題判別: 以下のカスタマイズ可能メンバーが変更されました。
 - FEKLOGS (バージョン 8.5.1 以降)

- RSE - 以下の新しいオペレーター・コマンドが RSED 開始タスクに追加されました。
 - MODIFY DISPLAY OWNER,DATASET=dataset (バージョン 9.0 以降)
 - MODIFY DEBUG GC,PID=pid (バージョン 9.0 以降)
- RSE: 以下のカスタマイズ不可の新しいディレクティブが `rsed.envvars` に追加されました。
 - `_CMDSERV_BASE_HOME` (バージョン 8.5.1 以降)
 - `_CMDSERV_CONF_HOME` (バージョン 8.5.1 以降)
 - `_CMDSERV_WORK_HOME` (バージョン 8.5.1 以降)
 - `RSE_DSN_SFEKLOAD` (バージョン 9.0 以降)
 - `(_RSE_JAVAOPTS) -Dlock.info.timeout` (バージョン 9.0 以降)
 - `(_RSE_JAVAOPTS) -DDSTORE_INITIAL_SIZE` (バージョン 9.0 以降)
 - `(_RSE_JAVAOPTS) -DDSTORE_MAX_FREE` (バージョン 9.0 以降)
- RSE: 以下の新しい必須のディレクティブが `rsed.envvars` に追加されました。
 - `RSE_HLQ` (バージョン 9.0 以降)
- RSE: 以下の新しいオプション・ディレクティブが `rsed.envvars` に追加されました。
 - `(_RSE_JAVAOPTS) -DRSE_DSICALL` (バージョン 8.5.1 以降)
 - `(_RSE_JAVAOPTS) -DDISABLE_REMOTE_INDEX_SEARCH` (バージョン 8.5.1 以降)
 - `(_RSE_JAVAOPTS) -DDISABLE_TEXT_SEARCH` (バージョン 9.0 以降)
 - `(_RSE_JAVAOPTS) -Dsearch.server.limit.hits` (バージョン 9.0 以降)
 - `(_RSE_JAVAOPTS) -Dsearch.server.limit.datasets` (バージョン 9.0 以降)
 - `(_RSE_JAVAOPTS) -Dsearch.server.limit.lines` (バージョン 9.0 以降)
 - `(_RSE_JAVAOPTS) -DDSTORE_SSL_ALGORITHM` (バージョン 9.0 以降)
- RSE: `rsed.envvars` の以下のカスタマイズ不可のディレクティブのデフォルト値が変更されました。
 - `(_RSE_JAVAOPTS) -DSPIRIT_EXPIRY_TIME` (バージョン 9.0 以降)
- RSE: `rsed.envvars` の以下のオプション・ディレクティブのデフォルト値が変更されました。
 - `(_RSE_JAVAOPTS) -Xms` (バージョン 8.5.1 以降)
 - `(_RSE_JAVAOPTS) -Xmx` (バージョン 8.5.1 以降)
 - `(_RSE_JAVAOPTS) -Dmaximum.clients` (バージョン 8.5.1 以降)
 - `(_RSE_JAVAOPTS) -Dmaximum.threads` (バージョン 8.5.1 以降)
 - `CGI_ISPPREF` (バージョン 9.0 以降)
- セキュリティー: 以下の新しいセキュリティー・プロファイルのサポートが追加されました。
 - `FEK.USR.**` (バージョン 8.5.1 以降)

構成可能なファイル

150 ページの表 24 は、バージョン 9.0 でカスタマイズされるファイルの概要を示しています。Developer for z Systems のサンプル・ライブラリー `FEK.SFEKSAMP`、

FEK.SFEKSAMV、および /usr/lpp/rdz/samples/ には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルの CARMA ソース・コードおよびそれらをコンパイルするジョブなど) が含まれています。

以下のメンバーおよびファイルはもうカスタマイズ可能でないか、使用されなくなっています。

- LOCKD 開始タスク
- ELAXMSAM サンプル DB2 ストアード・プロシージャ
- DB2 ストアード・プロシージャ用の ELAXMJCL サンプル JCL

注: サンプル・ジョブ FEKSETUP は、リストされているすべてのメンバーを別のデータ・セットおよびディレクトリー (デフォルトでは FEK.#CUST.* および /etc/rdz/*) にコピーします。

表 24. バージョン 9.0 のカスタマイズ

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKSETUP	FEK.SFEKSAMP [FEK.#CUST.JCL]	データ・セットおよびディレクトリーを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	使用されなくなったファイルに対するアクションを削除し、新規ファイル用のアクションを追加するために更新
JMON	FEK.SFEKSAMP(FEJJJCL) [FEK.#CUST.PROCLIB]	JES ジョブ・モニター用の JCL	なし
FEJJJCL	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(JMON)]	JMON メンバーの名前	JMON メンバーを参照
RSED	FEK.SFEKSAMP(FEKRSED) [FEK.#CUST.PROCLIB]	RSE デーモンの JCL	なし
FEKRSED	FEK.SFEKSAMP [FEK.#CUST.PROCLIB(RSED)]	RSED メンバーの名前	RSED メンバーを参照
ELAXF*	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	リモート・プロジェクト・ビルドなどのための JCL	ELAXFSP および ELAXFSQL が追加され、ELAXFCOC および ELAXFCPI は COBOL バージョン 5 をサポートするように更新された
FEKRACF	FEK.SFEKSAMP [FEK.#CUST.JCL]	セキュリティ定義の JCL	なし

表 24. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEJJCNFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	JES ジョ ブ・モニター 構成ファイル	オプションの 新規ディレク ティブが追加 された。既存 のオプション のディレクテ ィブは削除さ れた。
FEJTSO	FEK.SFEKSAMP [FEK.#CUST.CNTL]	TSO 実行依 頼用の JCL	なし
CRA\$VMSG	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA メ ッセージ VSAM を作 成するための JCL	なし
CRA\$VDEF	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA 構 成 VSAM を 作成するた めの JCL	RAM を除外す るためのサポ ートが追加さ れた
CRA\$VSTR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CARMA カ スタム情報 VSAM を作 成するための JCL	なし
CRA\$VCAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA 構 成 VSAM を 作成するた めの JCL	RAM を除外す るためのサポ ートが追加さ れ、VSAM 入 力が変更され た。
CRA\$VCAS	FEK.SFEKSAMP [FEK.#CUST.JCL]	CA Endeavor® SCM RAM 用に CARMA カ スタム情報 VSAM を作 成するための JCL	VSAM 入力 が変更された
CRASUBMT	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CARMA バ ッチ始動 CLIST	なし

表 24. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRASUBCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バ ッチ始動 CLIST	なし
CRABCFG	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA バ ッチ・アクシ ョン構成	新しいディレ クティブが追 加された
CRABATCA	FEK.SFEKSAMP [FEK.#CUST.CNTL]	CA Endeavor® SCM RAM 用の CARMA バ ッチ・アクシ ョンの JCL	変数ジョブ・ カードのサポ ートが追加さ れた
CRASHOW	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構 成	なし
CRATMAP	FEK.SFEKSAMP [FEK.#CUST.PARMLIB]	CA Endeavor® SCM RAM 用の CARMA 構 成	なし
CRANDVRA	FEK.SFEKPROC	CA Endeavor® SCM RAM 用の CARMA 割 り振り REXX	新しい DD 割 り振りが追加 された
CRA#VSLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のメッセージ VSAM を作 成するための JCL	なし
CRA#ASLM	FEK.SFEKSAMP [FEK.#CUST.JCL]	SCLM RAM のデータ・セ ットを作成す るための JCL	なし

表 24. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
CRA#VPDS	FEK.SFEKSAMP [FEK.#CUST.JCL]	PDS RAM のメッセージ VSAM を作 成するための JCL	なし
CRA#UADD	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を マージするた めの JCL	なし
CRA#UQRY	FEK.SFEKSAMP [FEK.#CUST.JCL]	RAM 定義を 抽出するた めの JCL	なし
CRAXJCL	FEK.SFEKSAMP [FEK.#CUST.ASM]	IRXJCL 置換 用のサンプ ル・ソース・ コード	なし
CRA#CIRX	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRAXJCL を コンパイルす るための JCL	なし
ADNCSDRS	FEK.SFEKSAMP [FEK.#CUST.JCL]	RESTful CRD サーバ ーを主 CICS 領域に対して 定義するた めの JCL	なし
ADNCSDTX	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザ クション ID を CICS 領 域に対して定 義するた めの JCL	なし
ADNTXNC	FEK.SFEKSAMP [FEK.#CUST.JCL]	代替トランザ クション ID を作成するた めの JCL	なし
ADNMSGHC	FEK.SFEKSAMP [FEK.#CUST.JCL]	ADNMSGHS をコンパイル するための JCL	なし
ADNMSGHS	FEK.SFEKSAMP [FEK.#CUST.COBOL]	パイプライン・メッセー ジ・ハンドラ ー用のサンプ ル・ソース・ コード	なし

表 24. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
ADNVCRD	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD リポジトリを作成するための JCL	なし
ADNCSDWS	FEK.SFEKSAMP [FEK.#CUST.JCL]	Web サービス CRD サーバーを主 CICS 領域に対して定義するための JCL	なし
ADNCSDAR	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD サーバーを非主 CICS 領域に対して定義するための JCL	なし
ADNJSPAU	FEK.SFEKSAMP [FEK.#CUST.JCL]	CRD のデフォルトを更新するための JCL	なし
ADNMFST	FEK.SFEKSAMP [FEK.#CUST.JCL]	マニフェスト・リポジトリを作成し、定義するための JCL	なし
FEKTEP2	FEK.SFEKSAMP [FEK.#CUST.SQL]	ELAXF* で使用される SQL コマンド入力	新規、カスタマイズはオプション
FEKTIAD	FEK.SFEKSAMP [FEK.#CUST.JCL]	ELAXF* で使用される SQL コマンド入力	新規、カスタマイズはオプション
AZUZUNIT	FEK.SFEKSAMP [FEK.#CUST.PROCLIB]	zUnit プロシージャの JCL	なし
FEKRNPLI	FEK.SFEKSAMP [FEK.#CUST.CNTL]	プリプロセッサ・フレームワーク内部から PL/I コンパイラーを呼び出すための REXX	なし

表 24. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
FEKLOGS	FEK.SFEKSAMP [FEK.#CUST.JCL]	ログ・ファイルを収集するための JCL	その他の検査が追加された。古いファイルに加えたカスタマイズはすべて再実行する必要がある。
rsed.envvars	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 環境変数	古いコピーをこのコピーに置き換え、カスタマイズを再実行する必要がある。
ISPF.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	TSO/ISPF クライアント・ゲートウェイ構成ファイル	なし
CRASRV.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	CARMA 構成ファイル	デフォルト値のサポートを追加
crastart.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CRASTART を使用するための CARMA 構成ファイル	なし
crastart.endevor.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	CA Endeavor® SCM RAM 用に CRASTART を使用するための CARMA 構成ファイル	なし
include.conf	/usr/lpp/rdz/samples/ [/etc/rdz/]	C/C++ コンテンツ・アシスト用の強制インクルード	なし
ssl.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE 暗号化通信構成ファイル	なし
rsecomm.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	RSE トレース構成ファイル	なし

表 24. バージョン 9.0 のカスタマイズ (続き)

メンバー/ファイル	デフォルト・ロケーション	目的	マイグレーションの注
pushtoclient.properties	/usr/lpp/rdz/samples/ [/etc/rdz/]	クライアント 構成ファイル に情報をプッ シュする	なし

IBM Rational Developer for z Systems Host Utilities、FMID HAKG900

バージョン 8.5 には同等の機能がいないため、マイグレーションに関する注意事項はありません。

構成可能なファイル

表 25 は、バージョン 9.0 でカスタマイズされるファイルの概要を示しています。Developer for z Systems Host Utilities のサンプル・ライブラリーの AKG.SAKGSAMP および /usr/lpp/rdzutil/samples には、ここに示したものより多くのカスタマイズ可能なメンバー (サンプルのコード・レビューの後処理スクリプトなど) が含まれます。

注: サンプル・ジョブ AKGSETUP は、リストされているすべてのメンバーを別のデータ・セット (デフォルトでは AKG.#CUST.*) にコピーします。

表 25. Host Utilities バージョン 9.0 のカスタマイズ

メンバーまたはファイル	デフォルト・ロケーション	目的	マイグレーションの注
AKGSETUP	AKG.SAKGSAMP [AKG.#CUST.JCL]	データ・セットを作成し、カスタマイズ可能ファイルのデータを取り込むための JCL	なし
AKGCC	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・カバレッジの JCL	なし
AKGCR	AKG.SAKGSAMP [AKG.#CUST.PROCLIB]	コード・レビューの JCL	新規 DD BIMPORT および BEXPORT
AKGCRADD	AKG.SAKGSAMP [AKG.#CUST.JCL]	サード・パーティーのコードをコード・レビューに追加するための JCL	なし

第 11 章 ホスト構成リファレンス

このセクションでは、「*IBM Rational Developer for z Systems* ホスト構成リファレンス (SC43-2912)」に記載する情報を要約します。詳細については、この資料を参照してください。

Developer for z Systems について

Developer for z Systems ホスト・システムは、クライアントがホスト・システム・サービスとデータにアクセスできるようにするために相互に作用する、複数のコンポーネントで構成されています。これらのコンポーネントの設計を理解しておくと、構成に関して適切な判断を行うことができます。

セキュリティに関する考慮事項

Developer for z Systems は他のホスト・コンポーネントと対話するため、セキュリティについての考慮が必要になります。

TCP/IP に関する考慮事項

Developer for z Systems では、TCP/IP を使用して、非メインフレーム・ワークステーションのユーザーに、メインフレームからアクセスすることができます。また、さまざまなコンポーネントやその他の製品同士の間の通信にも TCP/IP を使用します。

WLM に関する考慮事項

従来の z/OS アプリケーションとは異なり、Developer for z Systems は、ワークロード・マネージャー (WLM) で容易に識別できる一体構造のアプリケーションではありません。Developer for z Systems は、クライアントがホスト・システムのサービスとデータにアクセスできるようにするために相互に作用する、複数のコンポーネントで構成されています。これらのサービスの一部は異なるアドレス・スペースでアクティブとなるため、WLM 分類も異なることになります。

クライアントへのプッシュ機能に関する考慮事項

Developer for z Systems は z/OS Explorer のクライアントへのプッシュ機能 (またはホスト・ベースのクライアント制御機能) を拡張し、プロジェクト定義をサポートします。

CICSTS に関する考慮事項

この章には、CICS Transaction Server 管理者に有益な情報が記載されています。

AT-TLS のセットアップ

このセクションは、Application Transparent Transport Layer Security (AT-TLS) のセットアップ時、または既存のセットアップの検査時や変更時に起きる可能性があるいくつかの一般的な問題について、ユーザーを支援するためのものです。

参考文献

参考資料

本書では、以下の資料を参照しています。

表 26. 参考資料

資料名	資料番号	参照	参照 Web サイト
IBM Rational Developer for z Systems Program Directory	GI88-4172	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Program Directory for IBM Rational Developer for z Systems Host Utilities	GI88-4326	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systems ホスト構成ガイド	SC43-2913	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Rational Developer for z Systemsホスト構成リファレンス	SC43-2912	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 管理者ガイド	SC88-5664	Developer for z Systems	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
IBM Explorer for z/OS Host Configuration Guid	SC27-8437	z/OS Explorer	
IBM Explorer for z/OS Host Configuration Reference	SC27-8438	z/OS Explorer	
Communications Server IP CICS ソケット・ガイド	SC31-8807	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP 構成ガイド	SC88-8926	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Communications Server IP 構成解説書	SC88-8927	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 初期設定およびチューニング ガイド	SA88-8563	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 初期設定およびチューニング解説書	SA88-8564	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS JCL 解説書	SA88-8569	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS 計画: ワークロード管理	SA88-8574	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
MVS システム・コマンド	SA88-8593	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

表 26. 参考資料 (続き)

資料名	資料番号	参照	参照 Web サイト
Security Server RACF コマンド言語解説書	SA88-8617	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
Security Server RACF セキュリティー管理者のガイド	SA88-8613	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services コマンド解説書	SA88-8641	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX System Services 計画	GA88-8639	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
UNIX システム・サービス ユーザーズ・ガイド	SA88-8640	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/
REXX および z/OS UNIX システム・サービスの使い方	SA88-8644	z/OS 1.13	http://www-03.ibm.com/servers/eserver/zseries/zos/bkserv/

本書では、以下の Web サイトを参照しています。

表 27. 参照される Web サイト

説明	参照 Web サイト
Developer for z Systems IBM Knowledge Center	http://www-01.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html
Developer for z Systems ライブラリー	http://www-01.ibm.com/support/docview.wss?uid=swg27038517
Developer for z Systems ホーム・ページ	http://www-03.ibm.com/software/products/en/developerforsystemz/
Developer for z Systems 推奨サービス	http://www-01.ibm.com/support/docview.wss?rs=2294&context=SS2QJ2&uid=swg27006335
Developer for z Systems 機能拡張依頼	https://www.ibm.com/developerworks/support/rational/rfe/
Apache Ant のダウンロード	http://ant.apache.org/

情報資料

以下の資料は、必要なホスト・システム・コンポーネントのセットアップの問題を理解するのに役立ちます。

表 28. 情報資料

資料名	資料番号	参照	参照 Web サイト
ABCs of z/OS System Programming Volume 9 (z/OS UNIX)	SG24-6989	Redbook	http://www.redbooks.ibm.com/
System Programmer's Guide to: Workload Manager	SG24-6472	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 1: Base Functions, Connectivity, and Routing	SG24-7532	Redbook	http://www.redbooks.ibm.com/
TCPIP Implementation Volume 3: High Availability, Scalability, and Performance	SG24-7534	Redbook	http://www.redbooks.ibm.com/

表 28. 情報資料 (続き)

資料名	資料番号	参照	参照 Web サイト
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/

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

本プログラムに関する上記の情報は、適切な使用条件の下で 사용할 ことができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供され、いかなる保証条件も適用されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

プログラミング・インターフェース情報

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

製品資料に関するご使用条件

これらの資料は、以下のご使用条件に同意していただける場合に限りご使用いただけます。

適用される条件

IBM Web サイトの「ご利用条件」に加えて、以下のご使用条件が適用されます。

個人使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布（頒布、送信を含む）または表示（上映を含む）することはできません。

商業的使用

これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

権利

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入 関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

著作権使用許諾

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。サンプル・プログラムは、現存するままの状態を提供され、いかなる保証条件も適用されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

商標の帰属表示

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe および PostScript は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Cell Broadband Engine は、Sony Computer Entertainment, Inc. の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

Intel、Intel Centrino、Intel SpeedStep、Intel Xeon、Celeron、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

ITIL は英国 The Minister for the Cabinet Office の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

Linear Tape-Open、LTO、および Ultrium は、HP、IBM Corp. および Quantum の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。



Printed in Japan

SC43-2913-00



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21