

IBM Z System Automation  
バージョン 4 リリース 2

カスタマイズとプログラミング



## 注記

本書および本書で紹介する製品をご使用になる前に、[257 ページの『付録 G 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® ライセンス・プログラム IBM Z System Automation (プログラム番号 5698-SA4) バージョン 4 リリース 2 に適用されます。また改訂版およびテクニカル・ニュースレターで特に断りのない限り、これ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も適用されます。

本書は SC43-1378-01 の改訂版です。

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：

SC34-2715-02  
IBM Z System Automation  
Version 4 Release 2  
Customizing and Programming

### 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

© Copyright International Business Machines Corporation 1996, 2019.

# 目次

図.....	xi
表.....	xv
<b>アクセシビリティ</b> .....	<b>xvii</b>
支援機能の使用.....	xvii
ユーザー・インターフェースのキーボード・ナビゲーション.....	xvii
<b>本書について</b> .....	<b>xix</b>
本書の対象読者.....	xix
前提条件.....	xix
関連資料.....	xix
Z System Automation ライブラリー.....	xix
関連製品情報.....	xx
SC43-1378-02 の変更の要約.....	xx
<b>第 1 章自動化への新規アプリケーションの追加</b> .....	<b>1</b>
アプリケーションを自動化する前の準備.....	1
アドレス・スペース・プロパティ.....	1
ステップ 1 - アプリケーションの開始.....	1
ステップ 2 - アプリケーションの停止.....	2
ステップ 3 - アプリケーション・イベント.....	3
ステップ 4 - アプリケーションのモニター.....	4
未解決の応答の処理.....	4
トポロジ.....	4
自動化へのアプリケーションの追加.....	5
Application ポリシー・オブジェクトの定義.....	5
新規自動化構成ファイルの作成.....	5
<b>第 2 章自動化プロシージャの作成</b> .....	<b>7</b>
自動化プロシージャーを呼び出す方法.....	7
自動化プロシージャーを構造化する方法.....	7
初期化処理の実行.....	8
自動化が許可されるかどうかの判別.....	9
自動化処理の実行.....	9
自動化プロシージャーを汎用化する方法.....	13
プロセッサ操作コマンド.....	14
自動化プロシージャーのメッセージの開発.....	14
AOCMSG 呼び出し例.....	14
自動化プロシージャーの例.....	15
自動化プロシージャーの例に関する注.....	16
自動化プロシージャーのインストール.....	17
自動化プロシージャーのテストおよびデバッグ.....	18
補助モード機能.....	18
自動化プロシージャーをテストするための補助モードの使用.....	18
AOCTRACE を使用した自動化プロシージャー処理のトレース.....	18
NetView テストおよびデバッグ機能.....	19
テストに関する詳細情報.....	20
ユーザー独自の情報を自動化状況ファイル (ASF) 内にコーディング.....	20

プログラミングの推奨事項.....	20
グローバル変数名.....	21
<b>第 3 章アプリケーションのモニター.....</b>	<b>23</b>
監視状況モニター.....	23
ヘルス・モニター.....	24
概要.....	24
モニター・リソース・コマンド.....	25
リカバリー・ルーチンの作成.....	26
アクティブ・ヘルス・モニター.....	26
受動のイベント・ベース・ヘルス・モニター.....	28
プログラミング手法.....	29
OMEGAMON を使用したヘルス・モニター.....	31
概要.....	31
前提事項.....	32
OMEGAMON 対話.....	32
OMEGAMON 例外に基づいたヘルス・モニター.....	36
OMEGAMON シチュエーションに基づいたヘルス・モニター.....	37
CICSplex SM を使用したヘルス・モニター.....	40
コンポーネントの概要.....	40
VOST を管理するためのアプリケーションの作成.....	40
モニター・リソースの定義.....	40
JES3 コンポーネントのモニター.....	41
AOFRJ3MN ルーチン.....	42
AOFRJ3RC ルーチン.....	44
JES2 スプールのモニター.....	44
CICS および IMS 用の外部サブシステム接続モニター.....	45
IMS コンポーネントのモニター.....	45
<b>第 4 章ジョブ・ログ・モニター.....</b>	<b>47</b>
概要.....	47
カスタマイズ.....	48
SPIN パラメーター.....	52
状況情報.....	52
<b>第 5 章アラートに基づく通知.....</b>	<b>55</b>
概要.....	55
通信のフロー.....	55
アラートの使用可能化.....	56
SA z/OS でのセットアップ.....	56
INGALERT コマンド.....	60
<b>第 6 章可用性およびリカバリー時間の報告.....</b>	<b>61</b>
概要.....	61
リソースのライフ・サイクル.....	61
SMF レコードのレイアウト.....	62
SMF レコードの使用可能化.....	64
INGPUSMF ユーティリティ.....	64
出力.....	64
INGPUSMF ユーティリティ JCL.....	65
戻りコード.....	66
DB2 への SMF レポートの書き込み.....	66
カスタマイズ.....	67
出力.....	68
<b>第 7 章プロセッサ操作制御リソースの自動化.....</b>	<b>69</b>
プロキシー定義を使用した z/OS ターゲット・システムのプロセッサ操作リソースの自動化.....	69

概念.....	69
プロキシー・リソースの自動化のカスタマイズ.....	70
メッセージの自動化の準備.....	71
Linux コンソール・メッセージの自動化.....	72
NetView への Linux コンソール接続.....	72
大文字小文字混合の文字データを使用した Linux コンソールの自動化.....	72
セキュリティー考慮事項.....	72
制約事項および制限事項.....	72
自動化にプロセッサ操作メッセージを追加する方法.....	73
プロセッサ操作ターゲット・システムによって出されたメッセージ.....	73
新規自動化定義の作成.....	77
変更済み自動化環境のロード.....	77
同期 HW コマンドへのパイプおよび ISQCCMD の使用.....	77
ISQCCMD および PIPES の使用による非同期ハードウェア・コマンドの自動化.....	78
VM 第 2 レベル・システム・サポート (VM Second Level Systems Support).....	79
ゲスト・ターゲット・システム.....	79
ターゲット・システムのカスタマイズ.....	80
GETRAW 照会の SNMP MIB 属性を見つけるためのヒント.....	82

## 第 8 章 USS リソースの自動化.....83

z/OS UNIX システム・サービスの統合.....	83
インフラストラクチャーの概要.....	83
z/OS UNIX 自動化のセットアップ.....	83
z/OS UNIX リソースのカスタマイズ.....	83
例: sshd.....	88
ヒント.....	92
UNIX syslogd メッセージのトラップ.....	92
UNIX __console () サービスで発行されるメッセージのトラップ.....	93
デバッグ.....	94

## 第 9 章 コマンド受信側.....95

コマンド受信側のセットアップ.....	95
TSO/バッチ環境のセットアップ.....	95
SA z/OS によって自動化されるサブシステムとしてのコマンド受信側の定義.....	96
コマンド受信側によって使用されるコマンド作業タスクの定義.....	96
コマンド受信側の開始および停止.....	97
複数のコマンド受信側.....	97
バッチ・ジョブからの NetView コマンドの実行依頼.....	98
サンプル・バッチ・ジョブ JCL.....	98
コマンド・ステートメントの構文.....	98
別の NetView に対するコマンドの実行.....	99
バッチ・コマンド・インターフェースの JCL.....	99
TSO REXX プログラムから NetView コマンドを実行する.....	102
TSO から SA NetView Agent へのリモート・プロシージャ呼び出し.....	102
関数 INGRPC.....	102

## 第 10 章 リレーショナル・データ・サービス (RDS) の使用可能化..... 109

永続リレーショナル・データ・テーブルの使用可能化/使用不可化.....	109
System Automation リソースのインポート.....	109
通常のスナップショット.....	110
RDS の初期化.....	110
TSO 用 RDS の使用.....	110
セキュリティー・プロファイルのセットアップ.....	111
RDS テーブル・エディター.....	111
TSO 環境のカスタマイズ.....	112
TSO における表示/編集のための RDS 作業データ・セットの定義.....	112

<b>第 11 章</b>	<b>シスプレックス自動化の有効化.....</b>	<b>113</b>
	シスプレックス機能.....	113
	結合データ・セットの管理.....	113
	システム・ロガーの管理.....	114
	カップリング・ファシリティーの管理.....	115
	リカバリー・アクション.....	116
	ハードウェアの検証.....	123
	ハードウェア関連の自動化を使用可能にする.....	124
	ステップ 1: プロセッサの定義.....	124
	ステップ 2: ポリシー項目 PROCESSOR INFO の使用.....	124
	ステップ 3: 論理区画の定義.....	125
	ステップ 4: システムの定義.....	125
	ステップ 5: システムのプロセッサへの接続.....	125
	ステップ 6: 論理シスプレックスの定義.....	125
	ステップ 7: 物理シスプレックスの定義.....	125
	結合データ・セットの連続可用性を使用可能にする.....	125
	WTO(R) バッファ不足のリカバリーを使用可能にする.....	126
	システムの除去を使用可能にする.....	127
	ステップ 1: プロセッサとシステムの定義.....	127
	ステップ 2: アプリケーション・タイプ IMAGE のアプリケーションの定義.....	127
	ステップ 3: アプリケーション・グループの定義.....	127
	ステップ 4: IXC102A と IXC402D メッセージの自動化.....	128
	ステップ 5: 自動化テーブルのエントリーの確認.....	129
	長時間エンキュー (ENQ) を使用可能にする.....	129
	ステップ 1: リソースの定義.....	129
	ステップ 2: ジョブ/ASID 定義の実施.....	129
	ステップ 3: IEADMCxx シンボルの定義.....	129
	ステップ 4: コマンドの定義.....	130
	ステップ 5: スナップショット間隔の定義.....	130
	補助記憶域不足のリカバリーを使用可能にする.....	130
	ステップ 1: ローカル・ページ・データ・セットの定義.....	130
	ステップ 2: ジョブの処理の定義.....	130
	共通自動化項目の定義.....	130
	機能を使用するシステムのカスタマイズ.....	131
	追加の自動化オペレーター ID.....	131
	シスプレックス機能のオン/オフの切り替え.....	131
<b>第 12 章</b>	<b>ネットワークの自動化.....</b>	<b>133</b>
	自動化ネットワークの定義プロセス.....	133
	SDF フォーカル・ポイント・システムの定義.....	133
	ゲートウェイ・セッションの定義.....	134
	自動開始 TAF フルスクリーン・セッションの定義.....	134
<b>第 13 章</b>	<b>SA z/OS への VTAM アプリケーションの定義.....</b>	<b>137</b>
	SA z/OS のリカバリーへの VTAM アプリケーション・サブシステムの登録.....	137
<b>第 14 章</b>	<b>z/OS システムのシャットダウン.....</b>	<b>139</b>
	純粋な z/OS 環境で z/OS システムをシャットダウン.....	140
	定義の例.....	140
	GDPS 環境での z/OS システムのシャットダウン.....	141
	定義の例.....	142
	INGREQ の考慮事項.....	143
	システム・シャットダウン時の System Recovery Boost の自動活動化.....	143
<b>第 15 章</b>	<b>WTOR 処理.....</b>	<b>145</b>
	WTOR の処理フロー.....	145

着信 WTOR に対応するアクション.....	146
SA z/OS による WTOR の保管方法のカスタマイズ.....	146
1 次 WTOR の処理.....	147
使用上の注意.....	147

## 第 16 章 SA z/OS ユーザー出口..... 149

カスタマイズ・ダイアログ 出口.....	149
BUILD 処理用のユーザー出口.....	149
COPY 処理用のユーザー出口.....	150
DELETE 処理用のユーザー出口.....	151
CONVERT 処理用のユーザー出口.....	151
RENAME および IMPORT 機能のユーザー出口.....	151
その他の機能用のユーザー出口.....	152
カスタマイズ・ダイアログ出口の呼び出し.....	152
初期化出口.....	152
AOFEXDEF.....	154
AOFEXI01.....	154
AOFEXI02.....	154
AOFEXI03.....	154
AOFEXI04.....	154
AOFEXI05.....	154
AOFEXI06.....	155
AOFEXINT.....	155
環境設定出口.....	155
静的出口.....	156
AOFEXSTA.....	156
AOFEXX02.....	157
AOFEXX04.....	157
AOFEXX05.....	157
フラグ出口.....	157
パラメーター.....	159
タスク・グローバル変数.....	159
戻りコード.....	160
コマンド出口.....	160
AOFEXC00.....	160
AOFEXC01.....	160
AOFEXC02.....	161
AOFEXC03.....	161
AOFEXC04.....	161
AOFEXC05.....	161
AOFEXC06.....	161
AOFEXC07.....	161
AOFEXC08.....	161
AOFEXC09.....	162
AOFEXC10.....	162
AOFEXC11.....	162
AOFEXC13.....	162
AOFEXC14.....	162
AOFEXC15.....	162
AOFEXC16.....	162
AOFEXC17.....	162
AOFEXC18.....	163
AOFEXC19.....	163
AOFEXC20.....	163
AOFEXC21.....	163
AOFEXC22.....	163
AOFEXC23.....	164

AOFEXC24.....	164
AOFEXC25.....	164
AOFEXC26.....	164
AOFEXC27.....	164
AOFEXC28.....	164
AOFEXC29.....	165
疑似出口.....	165
自動化制御ファイルの再ロード許可出口.....	165
自動化制御ファイルの再ロード・アクション出口.....	165
初期化コマンドにおけるサブシステム UP.....	165
出口のテスト.....	165

## **第 17 章自動化ソリューション..... 167**

LOGREC データ・セット処理.....	167
SMF データ・セット処理.....	168
SYSLOG 処理.....	168
システム・ログ障害のリカバリー.....	168
SVC ダンプ処理.....	168
処理済み WTOR の表示からの削除.....	169
AMRF バッファ不足の処理.....	169
JES2 シャットダウン前のドレーン処理.....	169
IMS トランザクション・リカバリー.....	169
INGWHY ユーザー・アクションの定義.....	170
サンプル・シナリオ.....	170
ループ・アドレス・スペース抑制.....	172
準備.....	172
サンプル・ポリシーのコピー.....	172
SOAP サーバーのカスタマイズ.....	173
APL の構成.....	173
リンケージ.....	176
ビルド.....	176
資格情報の事前準備.....	176
モニターの活動化.....	177
インストールの検証.....	177
AOFRSA01.....	177
AOFRSA02.....	178
AOFRSA03.....	179
AOFRSA08.....	181
AOFRSA0C.....	183
AOFRSA0E.....	186
AOFRSA0G.....	187
AOFRSD07.....	188
AOFRSD09.....	189
AOFRSD0F.....	190
AOFRSD0G.....	192
AOFRSD0H.....	193
HASP099.....	195
INGRMJSP.....	195
INGRCJSP (AOFRSD01).....	197
INGRTAPE.....	198
INGRX711.....	199
INGRX740.....	199

## **付録 A グローバル変数..... 203**

読み取り専用変数.....	203
読み取り/書き込み変数.....	204
コマンドのパラメーター・デフォルト.....	220



<b>付録 B 状況表示機能のカスタマイズ</b> .....	<b>229</b>
状況表示機能の概要.....	229
状況表示機能の動作方法.....	229
SDF パネルのタイプ.....	229
状況記述子.....	230
SDF ツリー構造.....	231
状況記述子が SDF に及ぼす影響.....	232
SDF による特定の問題への取り組み支援.....	234
SDF パネルの定義方法.....	235
ツリー構造メンバーおよびパネル定義メンバーの動的なロード.....	235
複数システムでの SDF の使用.....	236
SDF コンポーネント.....	237
SDF タスクの始動と停止の方法.....	238
SDF の定義プロセス.....	238
ステップ 1: SDF 階層の定義.....	239
ステップ 2: SDF パネルの定義.....	240
ステップ 3: (オプション) SDF 初期設定パラメーターのカスタマイズ.....	243
ステップ 4: (オプション) カスタマイズ・ダイアログでの SDF の定義.....	244
<b>付録 C システム操作と自動リスタート・マネージャーの調整</b> .....	<b>245</b>
ARM エlement名の定義.....	245
自動リスタート・マネージャーの MOVE グループの定義.....	246
<b>付録 D メッセージの自動化</b> .....	<b>247</b>
汎用シノニム: AOFMSGSY.....	247
SA z/OS メッセージ表示: AOFMSGSY.....	248
オペレーター・カスケード: AOFMSGSY.....	250
<b>付録 E TSO ユーザーのモニター</b> .....	<b>253</b>
<b>付録 F 選択されたハードウェア・コマンドの z13/z13 固有のタイムアウト値</b> .....	<b>255</b>
特定タイムアウト値の定義.....	255
デバッグ・メッセージ.....	255
<b>付録 G 特記事項</b> .....	<b>257</b>
商標.....	258
製品資料に関するご使用条件.....	258
<b>用語集</b> .....	<b>261</b>
<b>索引</b> .....	<b>293</b>





1. アプリケーションのライフサイクル.....	3
2. システム操作の自動化プロシージャー.....	8
3. プロセッサ操作の自動化プロシージャー.....	8
4. 自動化プロシージャーのスケルトン.....	14
5. サンプル・モニター・コマンド.....	28
6. Take Action ダイアログ.....	39
7. アプリケーションのジョブ・ログ・モニターを定義する ISPF ダイアログ.....	49
8. すべての JESMSGLG メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ (1/3).....	49
9. すべての JESMSGLG メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ (2/3).....	49
10. 特定のメッセージの自動化を定義する ISPF ダイアログ.....	50
11. 特定のメッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ.....	50
12. マルチステップ・ジョブの特定メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ...	50
13. 非メッセージ印刷行のジョブ・ログ・モニターを定義する ISPF ダイアログ (1/2).....	50
14. 非メッセージ印刷行のジョブ・ログ・モニターを定義する ISPF ダイアログ (2/2).....	51
15. メッセージ INGY1300I およびモニターに対してメッセージ ID を定義するジョブの一般的な MAT エントリー.....	51
16. すべての JESMSGLG メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ (3/3).....	51
17. メッセージ INGY1300I およびモニターに対してメッセージ ID を定義していないジョブの一般的 な MAT エントリー.....	51
18. ジョブ・ログ・モニターの状況情報.....	52
19. アラート通信のフロー.....	56
20. アプリケーションのライフ・サイクル内のイベント.....	61
21. z/OS オフローダーでの SMF 処理.....	67
22. プロセスの停止定義.....	88

23. ファイルの削除.....	88
24. SSH デーモンの listen .....	91
25. UNIX のメッセージの例.....	93
26. コマンド処理のサンプル・パネル .....	128
27. コード処理のサンプル・パネル .....	128
28. システムに対するフォーカル・ポイント転送の定義 .....	134
29. 「フルスクリーン TAF アプリケーション定義」パネル .....	135
30. 1 次 WTOR の処理例 .....	147
31. SA z/OS 初期化時の SA z/OS 出口の順序.....	153
32. シナリオ: 提案されたアクションをオーバーライドする.....	171
33. MVS コンポーネント LOGREC に対するしきい値定義.....	179
34. 項目/タイプのペア MVSESA/LOGREC の MESSAGES/USER DATA ポリシー項目.....	179
35. MVS コンポーネント SMFDUMP に対するしきい値定義.....	181
36. 項目/タイプのペア MVSESA/SMFDUMP の MESSAGES/USER DATA ポリシー項目.....	181
37. MVS コンポーネント SYSLOG に対するしきい値定義 .....	183
38. 項目/タイプのペア MVSESA/SYSLOG の MESSAGES/USER DATA ポリシー項目 .....	183
39. MVSDUMP しきい値.....	185
40. MVSESA AMRF コマンド定義.....	188
41. JES2 DRAIN の詳細パネル.....	191
42. DISPACF パネル.....	192
43. 「DISPACF JES2 INITDRAIN」パネル.....	193
44. JES2 SPOOLSHORT リカバリー定義.....	194
45. DISPACF コマンド応答パネル.....	194
46. MVS コンポーネント LOG に対するしきい値定義.....	200
47. 項目/タイプのペア MVSESA/LOG の MESSAGES/USER DATA ポリシー項目 .....	201

48. SDF パネルの例 .....	230
49. SDF ツリー構造の例 .....	231
50. 状況コンポーネントにチェーニングされた状況記述子 .....	233
51. ツリー構造の定義の例 .....	240
52. SDF パネルの例 .....	241
53. パネル定義項目の例 .....	242



---

# 表

1. Z System Automation ライブラリー.....	xix
2. アプリケーションの開始.....	1
3. アプリケーションの停止 (1).....	2
4. アプリケーションの停止 (2).....	3
5. 監視状況モニター・ルーチン.....	23
6. ヘルス状況の戻りコード .....	27
7. 通知リストのポリシー項目.....	57
8. 通知リストの通信方式.....	57
9. INGALERT メッセージ ID のコード処理例.....	60
10. SMF レコードのレイアウト.....	62
11. INGPUSMF ユーティリティー・データ・セット・レコードの形式 .....	64
12. SINGSAMP SA z/OS サンプル・ライブラリー・ルーチン.....	75
13. カスタマイズ・ダイアログにおける sshd の定義の例 .....	90
14. sshd のパスに関する説明 .....	91
15. INGUSS コマンドおよび USS パイプの要約 .....	92
16. コマンド受信側のポリシー・エントリーの名前とタイプ .....	95
17. TSO/バッチ環境のサンプル名およびロード・モジュール .....	95
18. SA と NetView の機能およびオペレーター .....	97
19. 受信側の名前および関連ポリシー .....	109
20. RDS/SA 機能および NetView オペレーター .....	110
21. WTOBUF リカバリー処理 .....	117
22. 補助記憶管理.....	122
23. SYSTEM_SHUTDOWN コマンド処理項目の例 .....	140

24. GDPS_SYSTEM_SHUTDOWN コマンド処理項目の例.....	142
25. 外部定義の共通グローバル変数 .....	203
26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) .....	204
27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数.....	220
28. SDF コンポーネント.....	237
29. パネル定義項目の説明 .....	242



# アクセシビリティ

---

アクセシビリティ機能は、運動障害または視覚障害などで身体が不自由なユーザーがソフトウェア製品を快適に使用できるように支援します。IBM Z System Automation は複数のユーザー・インターフェースをサポートしています。製品機能およびアクセシビリティ機能はインターフェースによって異なります。

本製品の主なアクセシビリティ機能により、ユーザーは以下のようなことが可能になります。

- スクリーン・リーダー・ソフトウェアやデジタル音声シンセサイザーなどの支援技術を使用して、画面に表示されている内容を聴く。本製品でのこれらのテクノロジーの使用方法について、また画面拡大鏡ソフトウェアについて詳しくは、支援テクノロジーの製品資料を参照してください。
- キーボードのみを使用して、特定の機能または画面を使用したのと同等の機能を実行
- 画面上に表示されている情報を拡大

製品資料には、アクセシビリティを補助する以下の機能が含まれています。

- 資料はすべて HTML および変換可能 PDF の両方の形式で利用でき、ユーザーはスクリーン・リーダー・ソフトウェアを最大限に活用できます。
- 視覚障害のあるユーザーが画像の内容を理解できるように、資料内の画像はすべて、代替テキストとともに提供されています。

## 支援機能の使用

---

スクリーン・リーダーなどの支援機能は、z/OS<sup>®</sup> のユーザー・インターフェースを使用して機能します。この支援機能を使用して z/OS インターフェースにアクセスする場合、その特定情報については支援機能の資料を参照してください。

## ユーザー・インターフェースのキーボード・ナビゲーション

---

ユーザーは、TSO/E または ISPF を使用して z/OS ユーザー・インターフェースにアクセスできます。TSO/E インターフェース および ISPF インターフェースへのアクセスについては、「z/OS TSO/E 入門」、 「z/OS TSO/E ユーザーズ・ガイド」、 および 「対話式システム生産性向上機能 (ISPF) ユーザーズ・ガイド第 1 巻 z/OS」を参照してください。上記の資料には、キーボード・ショートカットまたはファンクション・キー (PF キー) の使用方法を含む TSO/E および ISPF の使用方法が記載されています。それぞれの資料では、PF キーのデフォルトの設定値とそれらの機能の変更方法についても説明しています。



# 本書について

本書では、「*IBM Z System Automation* 計画とインストール」での説明に従って標準インストールが完了した *IBM Z System Automation (SA z/OS)* を、ご使用の環境に適応させる方法について説明します。本書では、自動化に新しいアプリケーションを追加する方法や、独自の自動化プロシージャを作成する方法その他について説明します。

## 本書の対象読者

本書は、主に以下の処理を担当する自動化管理者の方々を対象としています。

- システムの自動化と操作環境のカスタマイズ
- 自動化プロシージャとその他の操作機能の開発

## 前提条件

本書を通して、*Z System Automation* と以下の文書をより理解することができます。

- *IBM Z System Automation* オペレーター・コマンド
- *IBM Z System Automation* プログラマーズ・リファレンス
- *IBM Z System Automation* 自動化ポリシーの定義

## 関連資料

### Z System Automation ライブラリー

xix ページの表 1 は、*Z System Automation* ライブラリー内の情報単位を示しています。これらの資料は、*IBM Knowledge Center* からダウンロードできます。

資料名	資料番号	説明
スタート・ガイド	SC43-5201	本書は <i>SA z/OS</i> の初心者を対象としています。初期計画、製品の構成、保護方法、自動化環境のカスタマイズ、および日常的に実行する基本操作タスクに関する情報が記載されています。
計画とインストール	SC43-1373	<i>SA z/OS</i> の新機能と、 <i>SA z/OS</i> の計画、インストール、構成、およびマイグレーションの方法について説明します。
カスタマイズとプログラミング	SC43-1378	標準インストール済み環境の適応、自動化のための新規アプリケーションの追加、独自の自動化プロシージャの作成、アプリケーションのモニター、アラートの有効化などの方法について説明します。
自動化ポリシーの定義	SC43-1376	自動化ポリシーの定義方法と維持方法について説明します。
ユーザーズ・ガイド	SC43-1381	<i>SA z/OS</i> の機能と、 <i>SA z/OS</i> を使用してシステムをモニターおよび制御する方法について説明します。
メッセージおよびコード	SC43-1377	<i>SA z/OS</i> の問題判別情報 (メッセージ、戻りコード、理由コード、状況コードなど) について説明します。
オペレーター・コマンド	SC43-1371	<i>SA z/OS</i> で使用可能なオペレーター・コマンドについて、その目的、形式、使用方法の詳細などを説明します。

表 1. Z System Automation ライブラリー (続き)

資料名	資料番号	説明
プログラマーズ・リファレンス	SC43-3863	SA z/OS のプログラミング・インターフェースと 状況表示機能 (SDF) の定義について説明します。
エンドツーエンド自動化	SC43-3364	z/OS のエンドツーエンド自動化アダプターと、エンドツーエンド自動化を有効にする方法、および Service Management Unite Automation への接続方法について説明します。
Service Management Unite Automation インストールおよび構成のガイド	SC43-5202	Service Management Unite Automation の計画、インストール、セットアップ、構成、およびトラブルシューティングの方法について説明します。
製品自動化 プログラマーズ・リファレンスとオペレーターのガイド	SC43-1374	組織内のすべての CICS 領域、Db2 領域、および IMS 領域を、ローカルかリモートかにかかわらずモニターおよび制御するシンプルで一貫した方法を提供するために、SA z/OS を使用して製品自動化コンポーネント (CICS 自動化、Db2 自動化、および IMS 自動化) をカスタマイズおよび操作する方法について説明します。
TWS 自動化 プログラマーズ・リファレンスとオペレーターズ・ガイド	SC43-3363	ZWS/TWS 自動化をカスタマイズおよび操作する方法について説明します。

## 関連製品情報

IBM Z System Automation をサポートする情報については、IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome>) の z/OS ライブラリーにアクセスしてください。

## SC43-1378-02 の変更の要約

本書には、「IBM System Automation for z/OS カスタマイズとプログラミング」(SC43-1378-01) に記載済みの情報が含まれます。

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。

### 新規情報および変更情報

- (OA59461) 新しい AOF\_FDBK\_CS\_HONOR グローバル変数が導入されました。これは、INGREQ/INGMOVE/INGGROUP FDBK コマンド処理において、CSONLY フラグ付き APG の SATISFACTORY 複合状況が考慮されるかどうかを制御します。
- INGINFO コマンドの RSTAT パラメーターを指定値に設定するために、新しい INGINFO\_RSTAT グローバル変数が導入されました。
- 動的リソースの作成や削除のプロジェクトなどのプログラムされたアクション用に、新しい INGDYN 出口 AOFEXC28 が導入されました。
- 新規動的リソース作成後の再開などのプログラムされたアクション用に、新しい INGDYN 出口 AOFEXC29 が導入されました。
- IBM Z System Automation V4.2 は、INGREQ コマンドまたは GDPS によって開始されたシステム・シャットダウン・フェーズの始めに、システム・リカバリー・ブーストを自動的にトリガーできます。143 ページの『システム・シャットダウン時の System Recovery Boost の自動活動化』を参照してください。
- 139 ページの『第 14 章 z/OS システムのシャットダウン』章が改訂されて、純粋な SA z/OS 環境および GDPS 環境における z/OS シャットダウン・プロセスや、z/OS シャットダウンの違いが示されるようになりました。

## 削除された情報

- グローバル変数 AOFPAUSE は削除されました。
- 自動化システム・リソース・ディスカバリー (オートディスカバリー) 機能は Z System Automation V4.2 から除去されました。



# 第 1 章 自動化への新規アプリケーションの追加

この情報では、SA z/OS で新規アプリケーションを追加およびモニターする場合の要件の概要を示します。

## アプリケーションを自動化する前の準備

製品を自動化するには、その製品の特性 (開始と停止の動作など) およびパラメーター (ジョブ名など) を抽出しておく必要があります。

以下のステップは、これらの特性を取得するのに役立ちます。完了したら、アプリケーションを自動化ポリシーに追加する必要があります。このアクティビティーについては、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。アプリケーションの自動化の主な要件は以下のとおりです。

- アドレス・スペース・プロパティー
- アプリケーションの開始
- アプリケーションのシャットダウン
- アプリケーション・イベント
- アプリケーションのモニター
- 未解決の応答の処理
- トポロジー

### アドレス・スペース・プロパティー

新規アプリケーションを追加する場合、以下に示すそのアプリケーションの最も重要な特性を認識しておく必要があります。

- ジョブ名
- JCL プロシージャー名
- マスター・スケジューラーまたはスケジューリング・サブシステムによってスケジュールされているか。
- MVS™ アプリケーションか、OMVS アプリケーションか、または別の種類のアプリケーションか (例えば、NetView タスク)。
- アプリケーションを実行する場所 (すべてのシスプレックスのシステム、シスプレックスで 1 回、またはシスプレックス内のシステムのサブセット)

アプリケーションの汎用プロパティーは、主に APPLICATION INFO ポリシーで定義します。

### ステップ 1 - アプリケーションの開始

新規アプリケーションを導入するには、そのアプリケーションを開始する方法と、アプリケーションを操作可能にするのに必要なすべてのアクションを考慮しておく必要があります。

したがって、以下のことを認識しておくことが重要です。

必要なアクション	使用可能な機能
アプリケーション自体を開始するために完了しておくべきアクションはありますか。	これらのアクションを自動化に組み込むには、アプリケーションの PRESTART ポリシーを使用します。ここで指定したすべてのコマンドは、開始コマンドの挿入より前に発行されます。

表 2. アプリケーションの開始 (続き)	
必要なアクション	使用可能な機能
アプリケーションの開始コマンドはどのようなものですか。また、アプリケーションの開始モード (例えば、DB2® データベースの normal および light start) に応じたさまざまな開始コマンドがありますか。	開始コマンドは STARTUP ポリシーに配置する必要があります。また、このポリシーは、さまざまな開始タイプを指定することにより、さまざまな開始コマンドに対する完全な柔軟性を提供します。開始タイプを設定すると、指定したコマンドが選択されます。開始タイプは、System Automation の実行時に簡単に選択または変更できます。
アプリケーションは、System Automation によって開始されないときは誰が開始しますか。別のインスタンスによって開始されますか。	APPLICATION INFO ポリシーでは、EXTERNAL STARTUP パラメーターを指定できます。
アプリケーションの初期化後に完了すべきアクションはありますか。	POSTSTART ポリシーを使用して、アプリケーションの完全な初期化の後に追加のコマンドを発行します。

注：すべての開始ポリシーが柔軟な開始タイプをサポートします。

## ステップ 2 - アプリケーションの停止

アプリケーションが不要になったら、計画的に停止するのに必要なすべてのステップを実行する必要があります。

したがって、以下のことを認識しておくことが重要です。

表 3. アプリケーションの停止 (1)	
必要なアクション	使用可能な機能
アプリケーションのシャットダウンを準備するためのコマンドを発行する必要がありますか。	SHUTDOWN INIT ポリシーを使用して、アプリケーションの終了を開始するために発行しておくべき追加のコマンドを指定します。
終了プロセスを開始するのはどのコマンドですか。また、停止コマンドが有効でないときはどうなりますか。	System Automation には、コマンドのエスカレーションという概念があります。これにより、終了コマンドの順序を指定できます。System Automation では、最初のコマンドを発行して、その効果を検証してから、より有効な次のコマンドを発行します。シャットダウン・タイプごとに異なるシャットダウン・コマンド・シーケンスを指定できる 3 つのポリシー (SHUTDOWN NORM、IMMED、FORCE) があります。
アプリケーションは、System Automation によって停止されないときは誰が停止しますか。アプリケーションを制御する別のインスタンスがありますか。	APPLICATION INFO ポリシーでは、EXTERNAL SHUTDOWN パラメーターを指定できます。
アプリケーションの正常な終了の後に完了すべき最終的な終了アクションはありますか。	SHUTDOWN FINAL ポリシーを使用して、アプリケーションの終了後に追加のコマンドを発行します。

場合によっては、予期しないアプリケーションの終了が発生することがあります。その場合、アプリケーションを再始動するには、いくつかのクリーンアップ・アクションの完了が必要になることがあります。したがって、以下のことを認識しておくことが必要です。



必要なアクション	使用可能な機能
アプリケーションを再始動するために完了しておくべき必要なクリーンアップ・アクティビティーはありますか。	状況コマンドという概念によって、この問題に対処します。アプリケーションが特定の状況になると、定義されたコマンドが発行されます。
予期しない終了が発生した場合にアプリケーションは再始動可能ですか。	System Automation では、アプリケーションに関して複数の終了状態を認識します。状態に応じて、System Automation では、リカバリー可能エラーとリカバリー不能エラーを区別できます。結果として、アプリケーションを再始動するかどうかを System Automation で判断します。この概念が、コード・マッチング処理です。また、APPLICATION INFO ポリシーの RESTART オプションで、System Automation がアプリケーションを再始動すべき状況を定義します。
アプリケーションは、別のアプリケーションによって自動的に再始動されますか。アプリケーションは ARM (自動リスタート・マネージャー) に対応し、自動的に再始動されますか。	System Automation には、ARM メカニズムと同じ動作を行うために、MOVE グループという概念が用意されています。アプリケーションの高可用性を実現するために、MOVE グループを使用することをお勧めします。

### ステップ 3 - アプリケーション・イベント

System Automation はイベントに対応します。より具体的には、アプリケーションまたはシステム 自体によって送信されたメッセージに対応します。

アプリケーションには多くの種類があります。アプリケーションのそれぞれが、さまざまなレベルのメッセージを送信します。このメッセージを使用して、アプリケーションの状況を判別することができます。メッセージは、アプリケーションのライフサイクル中にさまざまな状態を表します。通常は、MVS リソースが、アプリケーションの状況を判別するための適切なメッセージを提供します。OMVS 内のリソースは、ほとんどの場合、メッセージを提供することはありません。

ステップ 3 では、アプリケーションのライフサイクルにおける重要なメッセージを示します。以下から分かるように、リソースは 1 回開始されます。初期化の時間の後、リソースは完全に操作可能です。リソースが不要になると、停止コマンドが呼び出され、リソースを終了します。終了処理の後、リソースはもう存在していません。

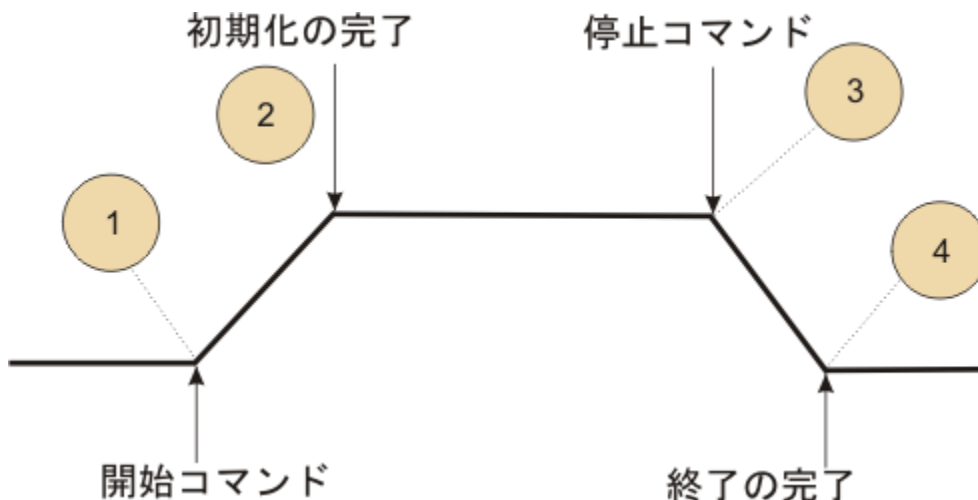


図 1. アプリケーションのライフサイクル

上記の 3 ページの図 1 に示されたアプリケーションのライフサイクルの各ポイントにおいて、重要なメッセージは何でしょうか。

例えば、アプリケーションが開始されたことをシステムが確認すると、メッセージ IEF403I が送信されます。アプリケーションが終了すると、IEF404I が発行されます。

特定のポイントにおいて、他の種類の有効なイベントはありますか。

ステップ 3 では、予期しないアプリケーションの終了の状態についても説明しています。

予期しない終了が発生した場合に、[3 ページの図 1](#) のポイント 3 および 4 において重要なメッセージはありますか。

例えば、メッセージ IEF450I は予期しない終了を示しています。

### ステップ 4 - アプリケーションのモニター

リソースの自動化は、そのリソースの開始および停止のみからなるわけではありません。コンポーネントが予期されたとおりに動作しているか、または既に終了しているかを判別する方法を認識しておくことも重要です。System Automation には、状態を判別できるようにするための適切なモニター・ルーチンが用意されています。

アプリケーションの種類と状態は、以下のものによって判別できます。

- アドレス・スペースの存在
- USS プロセスの存在
- NetView タスクの状況

対応するモニターは、フィールド MONITOR ROUTINE に指定できます。

ステップ 4 では、アプリケーションが発行するメッセージと、アプリケーションによるメッセージの発行頻度を示します。そのため、アプリケーションを積極的にモニターする必要があるか、またはアプリケーションの状態をメッセージから確実に得ることができるかを判断すると有益です。後者が当てはまる場合は、MONITOR INTERVAL を NONE に設定することができます。そうすると、NetView ログ内のメッセージと、不要なシステム・アクティビティーを減らすのに役立ちます。モニター・アクション自体は、アプリケーションの状態を検証するために、開始およびシャットダウンのサイクルで行われます。

対照的に、メッセージの発行頻度があまり高くないアプリケーションもあります。この場合は、バランスのとれたモニター間隔 (十分に明確な設定) により、定期的なモニター・サービスがアプリケーションの状態を検証するようにします。

場合によっては、リソースが稼働しているかどうかを認識するだけでは十分でないことがあります。多くの状態で、状況だけでなく、より詳しい情報も必要になります。モニター・リソースという概念では、リソースの状況を詳細に評価し、特定の状態に適切に対応するためのインフラストラクチャーを提供します。

### 未解決の応答の処理

SA z/OS は、受け取るすべての未解決のオペレーター宛メッセージ応答 (WTOR) に即時に応答しない場合は、それらの応答をトラッキングします。一部のアプリケーションは、同時に複数の未解決の WTOR を出すことがあり、すべての WTOR の重要度が同じわけではないため、WTOR は適宜、分類されます。

詳細は、[145 ページの『第 15 章 WTOR 処理』](#)を参照してください。

### トポロジー

通常は、自動化対象のアプリケーションは、JES2 または TCPIP のような基礎となるインフラストラクチャーに依存します。これは、このアプリケーションを開始するには、このインフラストラクチャーが使用可能になっていなければならないことを意味します。

逆に、このアプリケーションが、他のアプリケーションを開始する際の前提条件となることがあります。

同様に、アプリケーションを終了する前に、他のアプリケーションのうちどれを終了しなければならないかを考慮する必要があります。

上記のように、アプリケーション間には関係があります。

この時点で、絵を描いて、開始と停止の状態でのアプリケーション間の関係を視覚化しておく役立つ可能性があります。

SA z/OS では、複数の製品に対するソリューションを含むベスト・プラクティス・ポリシーが用意されています。このソリューションは、『[Add-on policies](#)』に PDF ファイル・フォーマットで示されています。

自動化しようとするソリューションについての詳細を調べるには、該当するファイルを参照してください。

## 自動化へのアプリケーションの追加

---

### Application ポリシー・オブジェクトの定義

新規アプリケーションを SA z/OS に追加するには、SA z/OS カスタマイズ・ダイアログを使用して、新しい Application ポリシー・オブジェクトを作成および定義する必要があります。

例えば、以下のようにして、カスタマイズ・ダイアログを使用して、SA z/OS で新規アプリケーションを自動化する方法を定義することもできます。

- アプリケーションに対し開始コマンドまたはシャットダウン・コマンドを指定する。
- 適切なモニター・ルーチンを指定する。
- アプリケーションを他のアプリケーションに関連させる関係を指定する。
- アプリケーションをアプリケーション・グループにリンクする。
- アプリケーションをどこに表示するかを検討する。

SA z/OS では、複数の製品に対するソリューションを含むベスト・プラクティス・ポリシーが用意されています。このソリューションは、『[Add-on policies](#)』に PDF ファイル・フォーマットで示されています。

新規アプリケーションを追加する方法と、System Automation のベスト・プラクティス・ソリューションにアクセスする方法については、「[IBM Z System Automation 自動化ポリシーの定義](#)」で詳細に説明されています。

### 新規自動化構成ファイルの作成

カスタマイズ・ダイアログでアプリケーションの定義を完了した場合は、新規の自動化構成ファイルを、更新されたポリシー・データベースから作成します。「[自動化ポリシーの定義](#)」の『[構成ファイルの作成](#)』を参照してください。

このステップを完了すると、アプリケーションは SA z/OS によって認識されるため、[5 ページの『Application ポリシー・オブジェクトの定義』](#)で定義したポリシーに従って自動化できるようになります。



## 第2章 自動化プロシージャーの作成

SA z/OS で提供されている基本自動化プロシージャーを補足するために、追加の自動化プロシージャーを作成することができます。例えば、ご使用のシステム上でのみ使用されるアプリケーションを自動化したり、サブシステムの自動化された特殊な操作を実行したりするプロシージャーを作成することができます。

SA z/OS のコマンドおよびルーチンは、メッセージのロギングや自動化フラグの検査などの基本機能を実行します。これらのルーチンは、ユーザー独自の自動化プロシージャーで使用することができます。

SA z/OS 自動化ルーチンは便利なルーチンです。このルーチンを使用すると、自動化プロシージャーは簡単かつ標準的な方法で、自動化制御ファイル、自動化状況ファイル、および NetView ログ・ファイルとのインターフェースが得られます。ユーザー独自のコードの中で使用できる場所ならどこにでも、これらのルーチンを使用することを強くお勧めします。

自動化プロシージャーを構造化する方法については、7 ページの『[自動化プロシージャーを構造化する方法](#)』を参照してください。また、自動化ルーチンの詳細と例、および自動化ルーチンで使用できるファイル・マネージャー・コマンドについては、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

### 自動化プロシージャーを呼び出す方法

自動化プロシージャーを呼び出すには、次のようないくつかの方法があります。

- SA z/OS 自動化を使用して、NetView 自動化テーブルから自動化プロシージャーを呼び出す
- NetView コマンド行に自動化プロシージャー名かそのシノニムをキー入力する。
- 別のプログラムから自動化プロシージャーを呼び出す。
- タイマーを使用して、自動化プロシージャーを始動する
- NetView EXCMD コマンドを使用して自動化プロシージャーを始動する。
- SA z/OS AOFEXCMD コマンド・ルーチンを使用して、自動化オペレーターで自動化プロシージャーを始動する。
- カスタマイズ・ダイアログで、以下の項目タイプについて、さまざまなポリシー項目用の「**コマンド・テキスト (Command text)**」フィールドまたは「**コマンド (Command)**」フィールドに自動化プロシージャー名を入力する。
  - アプリケーション
  - MVS Component
  - Timers
  - モニター・リソース

注：ルーチンを呼び出す前に、一部のルーチンはさまざまな環境設定を必要とするため、すべてのインターフェースを通じてすべてのルーチンを呼び出すことができるわけではありません。

### 自動化プロシージャーを構造化する方法

自動化プロシージャーの構造は、以下のようにより、3つの主要部分で構成することをお勧めします。

1. 初期化処理を実行する
2. 自動化が許可されているかどうかを判別する
3. 自動化処理を実行する

8 ページの図 2 は、システム操作の自動化プロシージャーの構造を示し、8 ページの図 3 はプロセッサ操作を示しています。

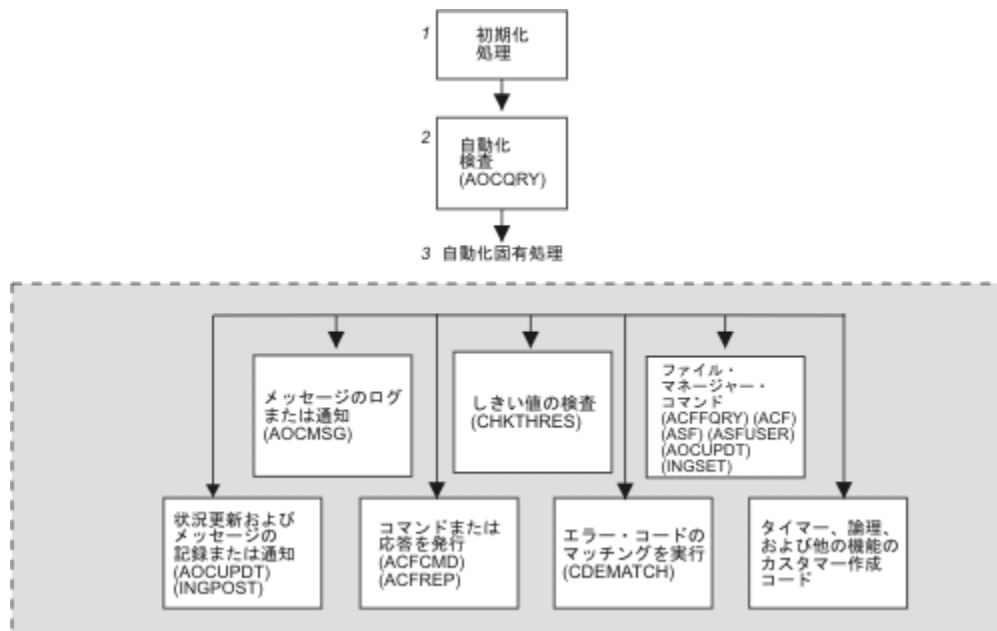


図 2. システム操作の自動化プロシーチャー

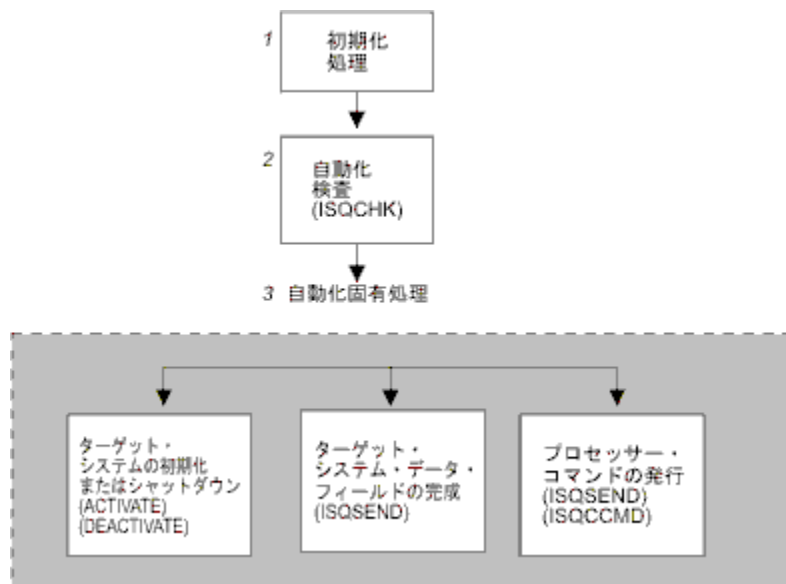


図 3. プロセッサ操作の自動化プロシーチャー

以下の項では、自動化プロシーチャーの各部分ごとについて詳しく説明します。

## 初期化処理の実行

単純な自動化プロシーチャーの場合には、初期化処理を必要としない場合があります。

初期化処理では、以下のことが行われます。

- すべてのエラー・トラップ・ルーチンの設定。
- 明示的にまたは実行時にローカル変数を設定することによる自動化プロシーチャーの識別。このステップにより、メッセージを記録し、通知を送信するルーチンのコーディングがさらに簡単になります。
- 自動化プロシーチャーのサブシステム定義値に使用する共通グローバル変数およびタスク・グローバル変数などのグローバル変数の宣言。

グローバル変数については、203 ページの『付録 A グローバル変数』を参照してください。

- デバッグがオンになっているかどうかの検査。



- デバッグ・メッセージを出す (デバッグがオンになっている場合)。
- 自動化プロシージャーの呼び出しの検証。  
これにより、オペレーターによる誤った自動化プロシージャーの呼び出しを防ぐことができます。自動化プロシージャーは、NetView または SAF 製品が提供するコマンド権限検査メソッドを使用して検証することもできます。
- NetView メッセージ・パラメーターの保管。自動化プロシージャーで NetView WAIT ステートメントを使用し、元のメッセージ・テキストや制御情報にアクセス する必要がある場合は、このステップが必要です。

自動化プロシージャー初期化セクションのコーディングについて詳しくは、[15 ページの『自動化プロシージャーの例』](#)、「[IBM Z NetView カスタマイズ・ガイド](#)」、および「[IBM Z NetView 自動化ガイド](#)」を参照してください。

## 自動化が許可されるかどうかの判別

### システム操作

アプリケーションおよび MVS コンポーネントの自動化プロシージャーは、NetView 自動化テーブルから呼び出されますが、AOCQRY 自動化ルーチン呼び出して常に自動化検査を行う必要があります。AOCQRY は、自動化フラグが自動化を許可しているか調べます。

これらの検査は、自動化すべきでないアプリケーションのメッセージや、自動化がオフになっている場合のメッセージを自動化するリスクを除去します。AOCQRY は、アプリケーションに固有の値を指定して、自動化プロシージャーで使用される大部分の共通グローバル変数およびタスク・グローバル変数の初期化も行います。

自動化検査ルーチンのコーディングに関する詳細については、「[IBM Z System Automation プログラマーズ・リファレンス](#)」を参照してください。

### プロセッサ操作

ほとんどのプロセッサ操作コマンドは、プロセッサ操作が開始されているときしか実行できません。プロセッサ操作がアクティブになっているかどうかを調べるには、自動化ルーチン内で ISQCHK コマンドを使用することができます。

プロセッサ操作が実行されていない場合は、ISQCHK は戻りコード 32 を戻し、以下のメッセージを出します。

```
ISQ0301 プロセッサ操作が開始されるまで、cmd-name コマンドは実行できません。
```

この場合、ユーザー・アプリケーションは ISQSTART コマンドを実行して、プロセッサ操作を始めることができます。

## 自動化処理の実行

自動化処理は、SA z/OS ルーチンとユーザー独自のコードとの組み合わせによって実行されます。

自動化プロシージャーのコーディングについては、次の文書で詳しく説明します。

- [9 ページの『システム操作における自動化処理』](#)
- [11 ページの『プロセッサ操作における自動化処理』](#)

### システム操作における自動化処理

この項では、システム操作の自動化処理をカスタマイズする方法について説明します。

#### 状況情報の更新

状況情報を更新するには、AOCUPDT を呼び出します。このルーチンを使用するのは、メッセージが状況の変更を示している場合です。

通常、これは自動化ルーチン ACTIVMSG、HALTMSG、および TERMMSG から行われます。ユーザーが独自に状況変更の更新を行うと、予期しない結果になることがあります。

詳しくは、「[IBM Z System Automation プログラマーズ・リファレンス](#)」を参照してください。

### メッセージのロギングと通知の送信

メッセージをログに記録し、通知を送信するには、AOCMSG を呼び出します。

AOCMSG は、次のアクションを実行します。

- メッセージを表示用またはロギング用にフォーマットする。
- メッセージを SA z/OS 通知メッセージとして通知オペレーターに出す。

詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

### コマンドおよび応答を出す

コマンドと応答を出すには、ACFCMD と ACFREP を呼び出します。

これらのルーチンを使用して、以下のことを行うことができます。

- メッセージへの応答で1つ以上のコマンドを実行する。
- メッセージへの応答で単一の応答を出す。
- ステップ・バイ・ステップ (PASS) 概念を使用して、自動化イベントへの対処または自動化イベントからのリカバリーを行う。

ACFCMD は1つ以上のコマンドを実行します。これは、単一の対処とステップ・バイ・ステップ (PASS) 概念の両方ともサポートします。詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

ACFREP は単一の応答を出します。これは、単一の対処とステップ・バイ・ステップ (PASS) 概念の両方ともサポートします。詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

多くの場合、単一処理およびパス処理もサポートする ISSUEACT 自動化プロシーチャーを使用できます。

### しきい値の検査

しきい値を検査および更新するには、CHKTHRES 自動化ルーチン呼び出します。CHKTHRES を使用して、しきい値の追跡と保守を行い、超過したしきい値レベルを基にしてリカバリー・アクションを変更します。

詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

### エラー・コードの検査

エラー・コードを検査するには、CDEMATCH を呼び出します。これはメッセージ内のエラー・コードと自動化固有のエラー・コード・セットを比較し、実行するアクションを判別します。

場合によっては、ISSUEACT および TERMMSG のコード・マッチング機能を使用できることもあります。

詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

### ファイル・マネージャー・コマンドの使用

ファイル・マネージャー・コマンドを使用して、SA z/OS アクセス制御ファイル (自動化制御ファイルおよび自動化状況ファイルなど) をアクセスすることができます。

- 自動化制御ファイルのロードまたは表示が必要な場合には、ACF コマンドを使用します。
- 自動化制御ファイルを迅速に照会するには、ACFFQRY コマンドを使用します。
- 自動化状況ファイルを表示するには、ASF コマンドを使用します。
- ユーザー固有の情報用に予約されている自動化状況ファイルのフィールドを修正するには、ASFUSER コマンドを使用します。

詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

### タイマー、ロジック、およびその他の機能に関する外部コードの使用

ユーザーの自動化プロシーチャーには、タイマーを設定したり、エンタープライズまたは自動化プロシーチャー自体に固有のロジックを実行したり、他の機能を実行したりするためのコードが必要になる場合があります。

例えば、以下のようなものがあります。



- コマンドの実行と応答のトラップ。

NetView WAIT コマンドまたは PIPE コマンドを使用して、コマンドを実行したり、応答をトラップすることができます。処理を続行する前に、システム・コンポーネントまたはアプリケーションの値または状況を調べる必要がある場合には、ユーザー・コードで、これらのコマンドを使用することが必要になることがあります。詳しくは、「*IBM Z NetView カスタマイズ・ガイド*」を参照してください。

- 処理を制御する共通グローバル変数およびタスク・グローバル変数の値の設定。

共通グローバル変数およびタスク・グローバル変数の値は、NetView コマンドを使用することによって設定することができます。あるメッセージのオカレンスから次のオカレンスまで保持する必要がある進行状況を示すフラグ、メッセージ・カウント、およびその他の標識を設定することが必要な場合には、これらの値を設定することが必要になることがあります。すべて外部化された SA z/OS グローバル変数のテーブルについては、「*IBM Z System Automation 自動化ポリシーの定義*」を参照してください。

また、「*IBM Z NetView カスタマイズ・ガイド*」にある共通グローバル変数とタスク・グローバル変数に関する説明も参照してください。

- 処理を再開するタイマー遅延の設定。

タイマー遅延は、NetView コマンド AT、AFTER、EVERY および CHRON を使用することによって設定することができます。これらのコマンドは、指定した時間が経過してから、自動化プロシーチャーが追加処理を行うために、その処理を再開するか、または別の自動化プロシーチャーを開始する必要があるときに使用できます。例えば、これらのコマンドを使用して、サブシステムのアクティブ・モニターを実行できます。詳しくは、「*IBM Z NetView Automated Operations Network ユーザーズ・ガイド*」に記載されている AT、AFTER、EVERY および CHRON コマンドに関する説明を参照してください。

## プロセッサ操作における自動化処理

この項では、プロセッサ操作の自動化処理をカスタマイズする方法について説明します。

### ターゲット・システムの初期設定

ユーザー・ルーチンがターゲット・システム (ハードウェアまたはオペレーティング・システム、あるいはその両方) を始動する必要がある場合、ISQCCMD ACTIVATE コマンドを実行します。

### ターゲット・システムのシャットダウン

ユーザー・ルーチンがターゲット・システムをシャットダウンする必要がある場合、ISQCCMD DEACTIVATE OCF コマンドを実行します。

このコマンドを実行してターゲット・システムを閉じる前に、実行中のすべてのユーザー・サブシステムをシャットダウンしてください。これにより、ターゲット・システムにおける予期しない状態が回避されます。

### その他の OCF コマンドの実行

プロセッサ操作によってサポートされるすべての OCF コマンドが、自動化ルーチンから実行できます。

これらのコマンドについて詳しくは、「*IBM Z System Automation オペレーター・コマンド*」を参照してください。

### 予約済みの SA z/OS コマンド

SA z/OS コマンド ISQISUP、ISQISTAT、ISQCMMT、ISQSTRT、ISQXIPM、ISQGPOLL、および ISQGSMSG は、ユーザーが使用するためのものではありません。これらは、ユーザーの自動化ルーチンで使用しないでください。

予期しない結果が起こることがあります。

ISQXDST、ISQXOPT、および ISQHELP コマンドは、オペレーター・コンソールからしか使用できません。そのため、ユーザーの自動化ルーチンの中で使用したり、ISQEXEC と一緒に使用したりしないでください。

次のコマンドは自動化用であり、ユーザーの自動化ルーチンの中で使用しないでください。ISQI101、ISQI212、ISQMCLR、ISQI320、ISQIUNX、ISQI347、ISQI470、ISQI886、ISQI888、ISQI889、ISQI128、ISQIVMT、ISQMVM11、ISQMVM12、ISQMWAIT、ISQMDCCF、ISQM020、および ISQIPLC。

### コマンド処理のシリアルライズ

コマンド処理をシリアルライズすると、コマンドと自動化ルーチンは、ターゲット・システム・コンソールに送信された順序で処理されるようになります。また、コマンド処理のシリアルライズは、コマンド・シーケンスに他のタスクが割り込むのを防ぐこともできます。

特定のターゲット制御タスクは、ターゲット・システムの初期化時に、特定のターゲット・システムに割り当てられます。複数のターゲット・システムは、1つのターゲット制御タスクを共用できますが、処理を実行するために、複数のターゲット制御タスクが1つのターゲット・システムに割り当てられることは決してありません。

コマンドまたは自動化ルーチンがターゲット・システムに送信されると、一部は実行タスク (ログオン・オペレーターまたは自動タスク) で処理し、一部ターゲット制御タスクで処理することができます。コマンドまたは自動化ルーチンが、ターゲット制御タスクによって処理される場合は、コマンドまたは自動化ルーチンはターゲット制御タスクに割り振られて処理されるか、あるいはキューに入れられてからターゲット制御タスクによって処理されます。これにより、コマンドと自動化ルーチンの処理がシリアルライズされます。シリアルライズすると、これらは、ターゲット・システム・コンソールに送信された順序で処理されるようになります。

NetView プログラムは、初期化時にデフォルトとして設定された優先順位をもっています。通常、NetView のもとで実行するものはすべて、優先順位が低くなっています。NetView DEFAULTS コマンドを使用して、設定内容を確認することができますが、設定値は変更しないでください。SA z/OS コマンド処理を希望通りにシリアルライズするには、SA z/OS で使用されるすべてのコマンドの優先順位設定値を「低」にする必要があります。SA z/OS で使用されるコマンドの優先順位を変更するか、そのコマンドに複数の優先順位があると、優先順位の差によってターゲット制御タスクのアーキテクチャーから生じるシリアルライゼーションが無効になることがあります。

### ターゲット制御タスクへの自動化ルーチンの送信

同じ一連の SA z/OS コマンドを定期的に行う場合、そのコマンドを NetView 自動化ルーチンにプログラミングすることができます。NetView 自動化ルーチンすべてに使用する指針に従ってください。

この場合、NetView 自動タスクまたはログオン・オペレーターは、このルーチンを実行したり、それをターゲット制御タスクに送信したりすることができます。次のコマンドは、自動化ルーチンをターゲット制御タスクに転送するのに使用します。

```
ISQEXEC target-system-name SC routine-name
```

自動化プロシーチャーを処理するために ISQEXEC コマンドを出すと、それらすべてのコマンドが、自動化プロシーチャー内に現れる順序で処理されます。これは、ISQEXEC コマンドが、コマンドを連続的に処理するターゲット制御タスクに処理を送信するためです。ISQEXEC コマンドによって同一コンソールに出される他のすべてのコマンドまたは自動化ルーチンは、ターゲット制御タスクが処理するためにキューに入れられ、直前のコマンドまたは自動化プロシーチャーが完了するまで始動されません。

また、ISQEXEC コマンドは、長時間実行されるすべてのコマンド・シーケンスからオリジナル・タスクを解放します。これにより、OST などの実行タスクを他の処理に使用することができます。

ISQEXEC コマンドは、コマンドのシリアルライゼーションを確実にするためにコンソールをロックしません。コマンド・シリアルライゼーション・プロセスは、ターゲット制御タスク割り振りスキーマの一環です。コマンドおよび自動化ルーチンは、それが発生した順序で処理されますが、他のタスクからのコマンドがコマンド・シーケンス列に割り込むことは可能です。

ISQEXEC コマンドについて詳しくは、「IBM Z System Automation オペレーター・コマンド」を参照してください。

### コンソールのロック

複数のルーチンとオペレーターが、同時に同一コンソールのアドレッシングを試みる場合があります。ISQEXEC コマンドは、ターゲット制御タスクが処理中のコマンド・シーケンスに他のタスクが割り込んでくるのを防ぎません。また、コンソールもロックしません。

コマンド・シーケンスへの割り込みを防ぐには、ISQXLOC コマンドおよび ISQXUNL コマンドを使用します。ISQXLOC コマンドは、コンソールへのアクセスをロックします。タスクが、ロックされたコンソールへのコマンドの実行を試みると、コンソールがロックされていることがそのタスクに通知され、コマンド

は失敗します。中断せずに処理する必要がある一連のコマンドが完了したら、コンソールへのアクセスをアンロックする ISQXUNL コマンドを実行してください。

自動化ルーチン内で ISQXLOC コマンドおよび ISQXUNL コマンドを使用して、自動化ルーチンが他のタスクからの干渉なしに確実に完了することができます。多くの SA z/OS コマンドを出す自動化ルーチンの場合、ISQEXEC コマンドの後、かつルーチンの先頭付近に次のコマンドを入れます。

```
ISQXLOC target-system-name SC
```

これで、次のコマンドによりロックをはずすまで、現行タスクに対するターゲット・システム・コンソールへのアクセスがロックされます。

```
ISQXUNL target-system-name SC
```

ISQXLOC を出したタスクのみが、正常に ISQXUNL を出すことができます。ISQXLOC コマンドが、ロックされた一連のコマンドから出されると、コンソールは既にロックされているために、そのコマンドは受け入れられません。

論理区画で実行中のターゲット・システムのシステム・コンソールをロックすると、そのシステム・コンソールは、そのプロセッサを使用する他のすべてのターゲット・システムに対してロックされます。ターゲット・ハードウェア定義で、他のすべてのターゲット・システム (論理区画) のシステム・コンソールに送信されるコマンドは、そのコンソールがアンロックされるまで実行されません。

ユーザー自動化ルーチンが、コンソールが解放されるのを待てない場合、ISQOVRD コマンドを使用してコンソールを制御してください。次のコマンドは、**クリティカル** 自動化ルーチン内だけで使用します。

```
ISQOVRD target-system-name SC
```

オーバーライド・コマンドを実行するルーチンが完了すると、ロックがはずされ、コンソールが使用できるようになります。

## 自動化プロシージャーを汎用化する方法

SA z/OS 自動化ルーチンを使用して、ユーザー独自の自動化プロシージャーを汎用化することができます。

汎用自動化プロシージャーは、3つの部分で構成されます。それぞれの部分ごとに、タスクを実行する上で役立つ特殊な自動化ルーチンが用意されています。

### 準備

自動化が許可されているかどうか、また、実行する必要があるかどうかを確認します。自動化ルーチン AOCQRY を使用します。

### 評価

何を実行すべきかを判断します。自動化ルーチン CDEMATCH を使用します。

### 実行

実行すべきことを実行します。自動化ルーチン ACFCMD または ACFREP を使用します。

```
*****
*****      Preparation      *****
*****

AOCQRY

- check if the resource is controlled by SA z/OS
- check if automation is allowed
- prepare/set task global variables for CDEMATCH, ACFCMD and ACFREP
...

CDEMATCH

- code matching (table search in ACF)
- find out required action
...

ACFCMD/ACFREP

- do required action:
  issue command / respond reply
```

図 4. 自動化プロシージャのスケルトン

前述の自動化ルーチンの詳細については、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。コマンド処理や応答処理の詳細については、「*IBM Z System Automation ユーザーズ・ガイド*」を参照してください。

### プロセッサ操作コマンド

ユーザーの自動化ルーチンは、可能な場合は常に、共通コマンドとも呼ばれる SA z/OS プロセッサ操作 OCF コマンドを利用します。これらのコマンドは、ターゲット・システムのプロセッサのハードウェア・タイプとは独立しています。したがって、これらのコマンドを使用すると、構成に新規プロセッサを追加する必要が生じた場合に、ユーザーの自動化ルーチンを変更する必要性が最小化されます。プロセッサ操作コマンドについて詳しくは、「*IBM Z System Automation オペレーター・コマンド*」を参照してください。

## 自動化プロシージャのメッセージの開発

追加プログラミングの有効範囲によっては、新規自動化プロシージャの作成で追加メッセージの開発も必要になることがあります。

メッセージを開発するのに使用できる SA z/OS 機能およびコマンドの中には、以下のものが含まれます。

- AOCMSG 自動化ルーチン（「*IBM Z System Automation プログラマーズ・リファレンス*」を参照）。
- AOCUPDT 自動化ルーチン（「*IBM Z System Automation プログラマーズ・リファレンス*」を参照）。

以下のステップは、メッセージ開発過程の概要です。

1. メッセージ ID を選択する。メッセージ ID が固有なことを確認してください。
2. NetView メッセージ・サービスを使用して、メッセージを NetView に定義する。  
メッセージの項目を DSIMSG データ・セットに入れます。このデータ・セットは、DSIMSG データ定義 (DD) 名で指定しておく必要があります。
3. AOCMSG 自動化ルーチンを使用してメッセージを出す（「*IBM Z System Automation プログラマーズ・リファレンス*」を参照）。
4. メッセージの項目を NetView DSIMSG データ・セットの実動コピーに追加する。

### AOCMSG 呼び出し例

この例は、メッセージ ABC123I を出す AOCMSG をコーディングする 方法を示しています。

DSIMSG メンバー DSIABC12 中のメッセージ項目は、以下のとおりです。

```
*****
120I ...
121I ...
122I ...
123I 10 40 THE EAGLE HAS &1
124I ...
*****
```

自動化プロシージャには、次の AOCMSG 呼び出しが含まれます。

```
<other 自動化プロシージャ code>
:
  AOCMSG LANDED,ABC123
:
<other 自動化プロシージャ code>
```

AOCMSG が、自動化プロシージャで指定されたように呼び出されると、DSIMSG メンバー DSIABC12 からメッセージ ABC123I が検索されます。変数 &1 の置換が起こり、次のメッセージが生成されます。

```
ABC123I THE EAGLE HAS LANDED
```

メッセージは、メッセージ ID とメッセージの最初のワードの間の 10 および 40 によって定義されていることに注意してください。これらは、メッセージが属している SA z/OS メッセージ・クラスです。このメッセージを出すと、クラス 10 またはクラス 40 のメッセージが割り当てられているすべての通知オペレーターにコピーが送信されます。

開発中の新規メッセージに関する詳細については、「IBM Z NetView カスタマイズ・ガイド」を参照してください。

## 自動化プロシージャの例

このセクションでは、z/OS メッセージを処理するアプリケーション・プログラムの例を示します。自動化プロシージャでは、SA z/OS 自動化ルーチンのサブセットを使用します。

```
/* SA z/OS 自動化プロシージャの例 */

1 Signal on Halt Name Aof_Error; Signal on Failure Name Aof_Error
  Signal on Novalue Name Aof_Error; Signal on Syntax Name Aof_Error

2 Parse source . . ident .

3 "GLOBALV GETC AOFDEBUG AOF."||ident||".0DEBUG AOF."||ident||".0TRACE"
  If AOFDEBUG = 'Y' Then
    "AOCMSG "||ident||",700,LOG,"||time()||","||opid()||","||Arg(1)
    loc.0debug = AOF.ident.0DEBUG
    loc.0trace = AOF.ident.0TRACE
    loc.0me = ident
    If loc.0trace <> '' Then Do
      loc.0debug = ''
      Trace Value loc.0trace
    End

4 save_msg = msgid()
  save_text = msgstr()
  lrc = 0

5 /* This procedure can only be called for msg IEA099A */
  If save_msg <> 'IEA099A' Then Do
    "AOCMSG "||loc.0me||",203,"||time()||","||opid()
    Exit
  End

6 "GLOBALV GETC AOFSYSTEM"
  cmd = 'AOCQRY '||save_msg||' RECOVERY '||AOFSYSTEM
  cmd
  svretcode = rc
  If loc.0debug = 'Y' Then
    "PIPE LIT /Called AOCQRY; Return Code was "||svretcode||"/" ,
    "| LOGTO NETLOG"
```



```

/* ----- **
** Check return code from AOCQRY **
** 0 = ok 1 = global flag off **
** 2 = specific flag off 3 = resource not in ACF **
** 4 = bad parms 5 = errors/timeout **
** ----- */
Select
7   When svretcode >= 3 Then Do
    "AOCMSG "loc.0me",206,, "time()",,, "cmd",RETCODE="svretcode
    lrc = 1
    End
8   When svretcode > 0 Then Do
    "GLOBALV GETT AUTOTYPE SUBSAPPL SUBSTYPE SUBSJOB"
    "AOCMSG "loc.0me",580,, "time()", "SUBSAPPL", "SUBSTYPE", " ,
    SUBSJOB", "AUTOTYPE", "save_msg
    lrc = 1
    End
9   Otherwise Do
    Parse Var save_text With . 'JOBNAME=' save_job 'ASID=' save_asid .
10  ehkvar1 = save_job
    ehkvar2 = save_asid
    "GLOBALV PUTT EHKVAR1 EHKVAR2"
11  cmd = 'ACFCMD ENTRY='||AOFSYSTEM||',MSGTYP='||save_msg
    cmd
    svretcode = rc
    If loc.0debug = 'Y' Then
        "PIPE LIT /Called ACFCMD; Return Code was "||svretcode||"/" ,
        "| LOGTO NETLOG"

/* ----- **
** Check return code from ACFCMD **
** 0 = ok 1 = no commands found in ACF **
** 4 = bad parms 5 = errors/timeout **
** ----- */
12  If svretcode > 1 Then Do
    "AOCMSG "loc.0me",206,, "time()",,, 'cmd',RETCODE="svretcode
    lrc = 1
    End
    End
End /* End of Select svretcode */

13 Exit lrc

14 Aof_Error:
Signal Off Halt; Signal Off Failure
Signal Off Novalue; Signal Off Syntax
errtype = condition('C')
errdesc = condition('D')
Select
    When errtype = 'NOVALUE' Then rc = 'N/A'
    When errtype = 'SYNTAX' Then errdesc = errortext(rc)
    Otherwise Nop
End
"AOCMSG "errtype",760,, "loc.0me", "sig1", "rc", "errdesc
Exit -5

```

## 自動化プロシージャの例に関する注

- 1** このステップは、負の戻りコード、オペレーター停止コマンド、および REXX プログラミング・エラーにエラー・トラップを設定します。
- 2** このステップは、自動化プロシージャの ID を定義します。
- 3** このステップでは、デバッグとトレースの設定を処理します (18 ページの『[AOCTRACE を使用した自動化プロシージャ処理のトレース](#)』を参照)。
- 4** 自動化プロシージャで使用する NetView メッセージ変数を保管します。
- 5** 許可検査を実行します。このプロシージャは、特定のメッセージの場合にのみ呼び出すことができます。

**6**

この節では、自動化検査を行います。

1. システム・メッセージが自動化制御ファイル (ACF) に保管される際に使用される項目名に関する情報が入っている AOFSYSTEM 共通グローバル変数を取り出します。
2. 自動化プロシージャー は AOCQRY コマンドを呼び出します。これは自動化フラグ検査を実行し、ACFCMD などの他の自動化ルーチンで使用するいくつかのタスク・グローバル変数を事前設定します。

**7**

AOCQRY が失敗した場合は、メッセージ AOF206I を出します。

**8**

自動化フラグがオフの場合は、メッセージ AOF580I を出します。

**9**

ジョブ名と ASID をメッセージの中に入れて報告します。

**10**

ACFCMD に対して EHKVARn 変数を設定します。

**11**

ACFCMD を呼び出し、構成ファイルに指定されているコマンドを出します。メッセージ IEA099A の自動化制御ファイル項目は次のようになります。

```
MVSESA IEA099A,
CMD=(,,'MVS C &EHKVAR1,A=&EHKVAR2')
```

**12**

ACFCMD が失敗した場合は、メッセージ AOF206I を出します。

**13**

処理が成功したか失敗したかを示す戻りコードを出して終了します。

**14**

ステップ **1** でエラーがトラップされた場合は、このコードによってメッセージがログに記録されます。

## 自動化プロシージャーのインストール

新規自動化プロシージャーのインストール・プロセスは、自動化プロシージャーが書かれている言語によって異なります。

- 自動化プロシージャーがコンパイル言語 (PL/I、C、またはアセンブラーなど) を使用する場合:
  1. ユーザーのソースをオブジェクト・モジュールにコンパイルまたはアセンブルする。
  2. オブジェクト・モジュールを NetView ロード・ライブラリーにリンク・エディットする。
  3. 自動化プロシージャー用の項目を、NetView DSIPARM データ・セットの CNMCMDU メンバーに組み込む。
- 自動化プロシージャーがインタープリター言語 (NetView コマンド・リストまたは REXX など) を使用する場合。
  1. 自動化プロシージャーを NetView コマンド・リスト・ライブラリーの中にコピーする。
  2. オプションで、この自動化プロシージャー用の項目を、NetView DSIPARM データ・セットの CNMCMDU メンバーに組み込む。これによって、その検出と呼び出しがより高速で行われるようになります。

使用するコードの作成とそのインストールの詳細については、「IBM Z NetView カスタマイズ・ガイド」を参照してください。

## 自動化プロシージャのテストおよびデバッグ

この節では、自動化プロシージャのテストに使用できる次のような SA z/OS と NetView の機能について説明します。

- SA z/OS 補助モード
- SA z/OS AOCTRACE オペレーター機能
- NetView テストおよびデバッグ機能

### 補助モード機能

自動化アクションを検査できるように、SA z/OS には補助モード 機能が用意されています。

補助モードがオンになっていると、通常 SA z/OS 自動化プロシージャによって実行されるアクション (コマンドや応答の発行など) は実行されません。代わりに、発生したであろうアクションを説明するメッセージが NETLOG に書き込まれます。

補助モードは、自動化フラグ (Automation、Initstart、Start、Recovery、Terminate または Restart) と関連しています。補助モードがアクションに対して使用されるかどうかは、自動化フラグによって判別されます。そのアクションが許可されているかどうかを調べるために、自動化フラグが検査されます。

補助モードの使用が必要となるのは、例えば以下のような場合です。

- ユーザーの自動化ポリシーの開発と使用の初期段階
- ユーザーの自動化ポリシーの変更後 (アプリケーションを自動化に追加した後など)
- 新規自動化プロシージャを SA z/OS コードに追加した後

### 自動化プロシージャをテストするための補助モードの使用

補助モードは、自動化プロシージャがユーザーのプロダクション・コードに追加される前に、自動化プロシージャの問題を検出するのに役立てることができます。補助モードは、コマンドと応答が NetView によって出される前に、それらをインターセプトします。

インターセプトしたコマンドと応答 (自動化ポリシーにコーディングされている) は、NetView ログに送信されるメッセージに再フォーマットされます。

再フォーマットされたコマンドは、メッセージ AOF320I で発行され、再フォーマットされた応答は、メッセージ AOF323I で発行されます。それぞれのメッセージには、自動化ポリシーに定義されたアクションおよび実行される実際のアクションに関する詳細な情報が含まれます。

SA z/OS の実行時に、INGAUTO コマンドを使用して補助モードを使用可能に設定し、関連した自動化フラグを値 L に設定することができます。DISPFLGS コマンドを使用して、現行の自動化フラグ設定を表示することができます。自動化フラグにその他の値を設定すると、補助モードは非活動化されます。

イベントが自動化アクションをトリガーし、さらに補助モードが使用可能である場合、SA z/OS はそのアクションを NetView ログに記録します。ログを見て、自動化が期待どおりに自動化されたか確認することができます。

補助モードは、AOCQRY を呼び出して自動化フラグを検査した後に、SA z/OS 自動化ルーチンを呼び出すすべてのルーチンに対して有効になります。

### AOCTRACE を使用した自動化プロシージャ処理のトレース

AOCTRACE コマンド・ダイアログでは、グローバル実行フロー・トレースと自動化プロシージャ固有のデバッグ・フラグの両方を保守します。

グローバル・フラグを設定すると、トレースをサポートするすべてのルーチンおよびすべてのメッセージ ID は、呼び出されるたびに、ステートメントを NetView ログに記録するようになります。AOFDEBUG グローバル変数は、グローバル・フラグ情報を自動化プロシージャに渡すのに使用します。グローバル・フラグは、グローバル・トレースがオフになっているとヌルに設定され、グローバル・トレースがオンになっていると Y に設定されます。

自動化プロシージャ固有のフラグを設定して、自動化プロシージャが実行時に行う内容についての情報を取得したり、または REXX トレースを活動化したりすることができます。デバッグ・フラグは、ヌル



または Y のいずれかで、AOF.clist.ODEBUG 共通変数 (ここで、clist は実際の自動化プロシージャー名) に保管されます。

トレース・フラグは、ヌルまたは以下のような有効な REXX トレース・タイプに設定されます。

- A (すべて)
- R (結果)
- I (中間)
- C (コマンド)
- E (エラー)
- F (障害)
- L (ラベル)
- O (オフ)
- N (通常)

S (スキャン) トレース・タイプは使用できません。

トレース・フラグは、共通グローバル変数 AOF.clist.OTRACE (clist は実際の自動化プロシージャー名) に保管されます。

メッセージ・トレースは、コマンド AOCTRACE MSG/id,ON|OFF を使用して、コマンド行からのみ設定できます。ここで、id はトレースされるメッセージです。

AOCTRACE については、「IBM Z System Automation オペレーター・コマンド」を参照してください。

### REXX コーディング例

トレースおよびデバッグの設定を処理するために REXX 自動化プロシージャーの先頭および最後に置くことができるコード例については、SINGSAMP ライブラリー内の AOFEXC00 を参照してください。

デバッグ機能をサポートするコードを書き込むときには、すべてのユーザー・プロシージャーに loc. を表示させ、loc.0debug フラグと出力関連情報の値を調べるコードのフラグメントを挿入する必要があります。loc.0me 割り当てによって、自動化プロシージャー名はどこにでも使用できるようになるので、それをすべてのデバッグ・メッセージの接頭部にすることができます。これにより、メッセージの送信元が分かるようになります。例:

```
Myproc:
  Procedure expose loc.
  If loc.0debug = 'Y' Then
    'PIPE LIT /' loc.0debug ' has called procedure MYPROC/',
    '| LOGTO NETLOG'
  Return
```

## NetView テストおよびデバッグ機能

NetView は、自動化プロシージャーのテストとデバッグを援助する複数の機能を提供します。

詳細なテストを行うために、自動化プロシージャーから出されたすべてのステートメントをトレースすることができます。このタイプのテストは、NetView コマンド・リストの &CONTROL ステートメント、および REXX プロシージャーの TRACE ステートメントによって使用可能になります。

また、コマンドだけをトレースするように、TRACE ステートメント および &CONTROL ステートメントで、簡単なトレースを指定することもできます。互換機能である対話式デバッグ援助機能は、PL/I および C でコーディングされたプログラムに対して使用することができます。

NetView コマンド・リスト、REXX プロシージャー、または PL/I ルーチン全体を通して、適切なポイントで NetView MSG LOG, PIPE LOGTO NETLOG コマンドを出して、特定のトレースを実行します。

メッセージに対する適切な構文解析と対応をテストするために、NetView WTO コマンドを出す短い自動化プロシージャーを作成します。この WTO は、NetView 自動化テーブルによって処理され、該当の自動化プロシージャーを起動します。自動化プロシージャーがジョブ名を必要とする場合、ジョブ名は適切な名前前に、一時的にハードコーディングする必要があります。この場合、WTO が NetView 領域から出されたの

で、そのメッセージと関連したジョブ名は NetView 領域です。以下は、自動化プロシージャの一例です。

```
WRITEWTO CLIST  
WTO &PARMSTR  
&EXIT
```

サンプルの自動化プロシージャは、ルーチン呼び出して任意の単一行メッセージを出すことができます。例えば、プログラムの始動を指示するメッセージ ABC123I を出すには、次のコマンドを入力します。

```
WRITEWTO ABC123I My testprogram PRGTEST has started.
```

### テストに関する詳細情報

テストについての詳細は、以下の資料を参照してください。

- *IBM Z NetView* カスタマイズ・ガイド

この資料には、ユーザー・プログラムの要件 (使用するユーザー・コードの解析を含む) と、出口ルーチンとコマンド・プロセッサを作成する際の詳細情報が記載されています。

- *IBM Z NetView* 自動化ガイド

この資料には、新規自動化プロシージャを作成するためのガイドライン (推奨する開発過程を含む) が記載されています。

## ユーザー独自の情報を自動化状況ファイル (ASF) 内にコーディング

ASFUSER コマンドを使用して、自動化状況ファイルにユーザー独自の情報をコーディングすることができます。

自動化状況ファイルには、その中で定義される各リソースと関連したユーザー・データ・フィールドが 40 個あります。これらのフィールドを使用して、ユーザー・コードが後からアクセスする必要があるリソースについての持続情報を保管することができます。SA z/OS をシャットダウンしても、ASF の情報は失われません。この情報は、以下のいずれかのイベントが起こるまで持続します。

- ASF VSAM データ・セットが削除され、再定義される。
- 情報が定義されているアプリケーションを含まない自動化制御ファイルを使用して SA z/OS が始動される。

SA z/OS がダウンしている間に環境が変わることもあるため、SA z/OS を初期化した場合は、必ず自動化状況ファイルの中に保管している情報が正確かどうかを調べる必要があることに注意してください。

ユーザー・データ用に予約された各自動化状況ファイル・フィールドには、20 文字まで入れることができます。ASFUSER コマンドによって、これらのフィールドのデータの更新および表示ができます。

ASFUSER コマンドについて詳しくは、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照してください。

### プログラミングの推奨事項

この節には、ユーザー独自の自動化プロシージャを書くときに必要なコーディングの労力を軽減し、ユーザー・自動化プロシージャのパフォーマンスを向上するのに役立つヒントと手法が記載されています。

- &SUBSAPPL、&SUBSTYPE、&SUBSJOB などの変数を、パラメーター値の代わりに使用します。

自己作成コマンドからの出力を表示するには、そのコマンドに接頭部としてプロシージャ名を付けます。名前の変更に影響を受けないようにするには、3 番目のトークンとしてプロシージャ名を提供する「parse source」REXX ステートメントを使用してください。

AOCQRY 呼び出しのリソース・フィールドに NetView コマンド・リスト言語変数 JOBNAME を使用すると、自動化プロシージャは、メッセージを出せるすべてのジョブの既知のメッセージをサポートするように作成することができます。

- できる限りデフォルトを使用して、コーディングを最小化してください。
- 一般的なエラー・コードを使用してください (CDEMATCH を参照)。

- 使用可能なメッセージ解析技法を使用してください。
  - メッセージのフィールド位置に依存せずにメッセージを解析するには、NetView コマンド PARSEL2R または REXX PARSE コマンドを使用してください。
  - NetView 自動化テーブルの中でメッセージを解析して、必要なフィールドだけを自動化プロシージャに送信してください。
- CDEMATCH 呼び出しで ENTRY フィールドはコーディングしないようにしてください (デフォルトは最後の AOCQRY 呼び出しから戻される SUBSAPPL です)。
- 適切な自動化フラグを使用してください。
- 以下のようなコードの作成時に考慮する制限を含め、「IBM Z NetView カスタマイズ・ガイド」に記載のコーディング要件を検討してください。
  - TVBINXIT がオンになっているときの制約事項
  - 変数名
  - マクロの使用
  - レジスターの使用
  - プログラムの再入力
- できる限り SA z/OS 自動ルーチンを使用してください。保守のオーバーヘッドが削減されます。
- 以下の理由から、できる限り SA z/OS プロセッサ操作の共通コマンドを使用してください。
  1. ターゲット・システムのプロセッサのハードウェア・タイプから独立している。
  2. 新規プロセッサをエンタープライズに追加したときに、ユーザーの自動化ルーチンを変更する必要性が最小化される。
- 新規コードと関連したオンライン・ヘルプ・テキストを表示し、新規コードの一部となる新規コマンドのフルスクリーン・インターフェースを開発するには、NetView VIEW コマンドの使用を考慮してください。VIEW コマンドについては、「IBM Z NetView カスタマイズ・ガイド」を参照してください。

## グローバル変数名

---

ユーザー独自の自動化プロシージャを作成する場合は、作成するどのグローバル変数の名前も、SA z/OS の外部グローバル変数や内部グローバル変数の名前と競合しないことを確認する必要があります。

また、次の文字で始まる名前を使用することはできません。

- CFG
- AOF
- ING
- ISQ
- EVI
- EVE
- EVJ



## 第3章 アプリケーションのモニター

SA z/OS は、ご使用のアプリケーションをモニターするためにさまざまな方法を提供します。

- 監視状況モニター・ルーチンを使用することにより、SA z/OS は、ご使用のアプリケーションとその他のさまざまな自動化リソースがアクティブであるか、非アクティブであるか、あるいは開始中であるかを判別できます。監視状況モニター・ルーチンは常に使用可能に設定すること、また、可能な場合はプロダクト提供のルーチンを使用することをお勧めします。詳しくは、[23 ページの『監視状況モニター』](#)を参照してください。
- モニター・リソースを使用すると、オプションで、ご使用のアプリケーションのヘルスをモニターし、ヘルス状況が変更した場合にアプリケーションをリカバリーすることができます。SA z/OS では、アクティブなヘルス・モニターと受動のイベント・ベース・ヘルス・モニターは区別されます。詳しくは、[24 ページの『ヘルス・モニター』](#)を参照してください。

アクティブ・ヘルス・モニターと受動ヘルス・モニターは、以下の領域で SA z/OS によりサポートされます。

- コンソール・メッセージに基づいた、JES3 のヘルス・モニター
- IBM OMEGAMON 標準モニター の例外または IBM Tivoli OMEGAMON XE のシチュエーションに基づいた、z/OS、Db2、CICS<sup>®</sup>、IMS、およびその他のコンポーネントのヘルス・モニター
- CICSplex<sup>®</sup> SM に基づいた、CICS のヘルス・モニター
- コンソール・メッセージに基づいた、IMS のヘルス・モニター

### 監視状況モニター

SA z/OS は、ポリシー・アドミニストレーターがカスタマイズ・ダイアログで指定したルーチンを実行して、アプリケーションの監視状況を判別します。

このルーチンは個々のアプリケーションごとに指定でき（「*IBM Z System Automation* 自動化ポリシーの定義」を参照）、しかも、システム全体のすべてのアプリケーションについてデフォルトのモニター・ルーチンを指定できます（カスタマイズ・ダイアログ内の「AUTOMATION INFO ポリシー項目」を参照）。

[23 ページの表 5](#) には、アプリケーション・モニターとして指定できるルーチンを一覧表示します。

モニター・ルーチン	以下のルーチンを使用
AOFADMON	このルーチンは非推奨となっており、前のリリースとの互換性のためにのみ提供されています。このルーチンは、 <b>MVS D A,jobname</b> コマンドを実行してアプリケーションの状況を判別します。使用されるジョブ名は、カスタマイズ・ダイアログで定義されたジョブ名です。
AOFATMON	このルーチンは、NetView 環境内で作動するタスクの状況を判別するために使用されます。
AOFAPMON	このルーチンは、プログラム間インターフェース (PPI) の受信側の状況を判別します。
AOFPCSM	このルーチンは、SA z/OS プロセッサ操作のアプリケーションの状況をモニターするために使用される専用ルーチンです。
AOFUXMON	このルーチンは、アプリケーション・タイプ USS のリソースの状況を判別します。このリソースは、z/OS UNIX プロセス、UNIX ファイル・システム (zFS) 内のファイル・システム、または TCP ポートのいずれかです。AOFUXMON は、リソースの種類（プロセス、ファイル、またはポート）に応じて、使用する内部モニター・メソッドを決定します。

表 5. 監視状況モニター・ルーチン (続き)	
モニター・ルーチン	以下のルーチンを使用
INGPJMON	このルーチンは、z/OS で特定のジョブ名を持つアドレス・スペースを検索することにより、アプリケーションの状況を判別します。使用されるジョブ名は、アプリケーションのカスタマイズ・ダイアログに定義されたジョブ名です。
INGMTSYS	このルーチンを使用して、BCPII の使用について IMAGE アプリケーションをモニターできます。
INGROMON	このルーチンを使用して、OMVS をモニターできます。
INGPSMON	このルーチンは、サブシステムのサブシステム・インターフェースへの登録をモニターします。
ISQMTSYS	このルーチンを使用して、プロキシーで表されるプロセッサ操作のターゲット・システム・リソースをモニターできます。プロキシー定義の使用方法的例については、69 ページの『 <a href="#">プロキシー定義を使用した z/OS ターゲット・システムのプロセッサ操作リソースの自動化</a> 』を参照してください。アクティブなオペレーター・コンソール接続が必須です。これは、z/OS コマンド (例えば、dt) の送信および関連する応答の受信に使用されます。

SA z/OS は、すべてのモニター・ルーチン (SA z/OS で提供されたもの、またはユーザー独自のもの) から一定の戻りコードを期待しています。それらの戻りコードは、以下のいずれかです。

#### RC

意味

0

アクティブ

4

開始中

8

非アクティブ

12

エラー

注: 独自のモニター・ルーチンを作成する場合は、このルーチンも初期状況判別時に実行されることを考慮する必要があります。この処理の後に、共通グローバル変数 `AOFCOMPL` が YES に設定され、「初期化関連処理が完了しました」を示すメッセージ `AOF540I` が発行されます。

## ヘルス・モニター

### 概要

ヘルス・モニターは、モニター・リソースと呼ばれる特殊リソースを使用して実行します。モニター・リソース (リソース・タイプは MTR) は、他のリソース (通常は、アプリケーションまたはアプリケーション・グループ。あるいはより一般的に言えば、モニター可能なすべてのオブジェクト) のヘルス状況を取得するために使用されるポリシー・オブジェクトです。リソースがアクティブであるというだけでなく、リソースが実行されている状況を調べる必要がある場合に、ヘルス状況が役立ちます。

ヘルス状況は、アプリケーション固有のパフォーマンスとヘルス・モニター情報を提供するために使用されます。例えば、アプリケーションはアクティブであるが、システム管理者が定義したパフォーマンス目標を達成できていない場合があります。ヘルス状況は、情報用のみに使用することも、あるいは決定を下したり、必要であればアプリケーション用の自動化をトリガーしたりするために、自動化マネージャーで使用することもできます。

モニター・リソースは、カスタマイズ・ダイアログ内の項目タイプ MTR に定義します。モニター・リソースは、他のすべての SA z/OS リソースと同様の特性を備えたリソースです。



モニター・リソースは、アプリケーション・リソース (APL) またはアプリケーション・グループ・リソース (APG) に接続されます。モニター対象のオブジェクトのヘルス状況は APL および APG に伝搬されて、その結果、結合されたヘルス状況がそこに存在することになります。MTR の定義および接続は、カスタマイズ・ダイアログで行うことができます (「*IBM Z System Automation* 自動化ポリシーの定義」を参照)。

モニター・リソースは、以下の異なる 2 つの方法で、オブジェクトのヘルス状況を取得します。

- ポーリングによる積極的な方法—モニター・コマンドを定期的に行う
- イベントの処理による受動的な方法

アクティブ・モニターは、MTR ポリシーに定義された間隔に基づいて定期的にスケジュールされます。

パッシブ・モニターでは、モニター間隔を定義しませんが、初期ヘルス状況の判別用にモニター・コマンドを定義することができます。パッシブ・モニターが INGMON コマンドを使用してヘルス状況を設定するかどうかは、他のイベントに依存します。

モニター・リソースは、モニター対象のオブジェクト、およびオプションとしてジョブに、明示的にバインドすることができます。これにより、SA z/OS は、さまざまなモニター・イベントを一般的な方法で処理することができます。モニター対象オブジェクトとしては、例えば、OMEGAMON XE シチュエーション、あるいは CICSplex System Manager (CICSplex SM) から通知されるイベントなどがあります。[28 ページの『受動のイベント・ベース・ヘルス・モニター』](#)を参照してください。モニター対象オブジェクトは、指定されなかった場合は、モニター・リソース名から派生することに注意してください。

各ヘルス状況 (NORMAL、WARNING、MINOR、CRITICAL および FATAL) に、1 つ以上のリカバリー・コマンドを関連付けることができます。これらのコマンドは、モニター・リソースが対応するヘルス状況に切り替わると SA z/OS によって起動されます。

モニター・リソースは、DISPMTR コマンドで表示および制御されます。適切な通知リスト指定が行われている場合、モニター・リソースは、SDF だけでなく Tivoli Enterprise Portal (TEP) にも表示されます。

## モニター・リソース・コマンド

モニター・リソースを定義するときには、活動化、非活動化、およびモニター・コマンドを指定できます。NetView 環境で実行できるものであれば、どんなコマンドでも使用できます。これらのコマンドは、2 つのグループに分けられます。

- 戻りコード 0 を期待する NetView 活動化コマンドと非活動化コマンド
- ヘルス状況に戻すモニター・コマンド

これら 2 つのグループの間の主な相違は、活動化コマンドと非活動化コマンドが 1 回だけ実行され、SA z/OS がゼロの戻りコードを期待するという点にあります。

活動化コマンドがゼロ以外の戻りコードで終了した場合は、モニター・リソースは、INACTIVE 状況のままになります。非活動化コマンドがゼロ以外の戻りコードで終了した場合は、モニター・リソースは、BROKEN 状況になります。

- 活動化コマンドはオプションであり、モニター・ルーチンを実行できる環境を設定するために使用できます。このコマンドは、モニターが開始されるたびに実行されます。このコマンドは、戻りコード 0 で終了しなければなりません。
- 非活動化コマンドはオプションであり、環境を終結処理するために使用できます。このコマンドは、モニターが停止されるたびに実行されます。このコマンドは、戻りコード 0 で終了しなければなりません。
- モニター・コマンドは活動化コマンドの後に実行され、モニター間隔が指定されていれば、定期的に行われます。SA z/OS は、モニター・コマンドから有効なヘルス状況コードが戻されるのを期待します。さらに、モニター・コマンドは、その後でヘルス状況に付加されるメッセージを出すことがあります。モニター間隔が指定されていないということは、所与のモニター・リソースが受動の、あるいはイベント・ベースのヘルス・モニターであることを示します。この場合、モニター・コマンドはオプションであり、これが指定されると、最初のヘルス・モニターに対してのみ呼び出されます。そうではなく、モニター間隔が指定されている場合、所与のモニター・リソースは、アクティブ・ヘルス・モニターです。この場合、ヘルス状況に戻すためにモニター・コマンドを指定する必要があります。

活動化、非活動化およびモニター・コマンドには、NetView でサポートされる任意の言語 (REXX、アセンブラ、PL/I、C、または NetView コマンド・リスト言語 (NCLL)) で書き込まれたコマンド・プロシージャ

ーを指定できます。モニター・ルーチンの作成は、簡単な場合もあれば、複雑な場合もあります。複雑さは、モニター対象のアプリケーションによって変わります。

## リカバリー・ルーチンの作成

モニター・リソースのヘルス状況が、リカバリー・ルーチンが定義されている状況に切り替わるたびに、そのリカバリー・ルーチンが起動されます。リカバリー・ルーチンの目的は、モニター・リソース (したがって、モニター対象オブジェクト) を NORMAL のヘルス状況に戻すことです。

### リカバリー手法

MESSAGES/USER DATA ポリシー項目内のユーザー・データを使用して、他のリカバリーが既に進行中のときの追加のリカバリー処理を使用不可にすることができます。

事前定義されたキーワード DISABLETIME との組み合わせで、リカバリー停止時間を hh:mm:ss、mm:ss、:ss、または mm の形式で指定することができます。リカバリーが使用不可に設定されている間は、MESSAGES/USER DATA ポリシー項目内で指定されたメッセージおよび例外に対して、このモニター・リソースのためにコマンドが処理されることはありません。

リカバリーは、リカバリー使用不可時間の期限が切れたあと、自動的に使用可能になります。また、例えばオプション CLEARING=YES を使用して INGMON コマンドを呼び出すことによって、早めにリカバリーを使用可能にすることもできます。

```
INGMON CI2XREP MSGTYPE=XREP CLEARING=YES
```

期間全体にわたって、ますます強いリカバリー・アクションを強制する必要が生ずることがあります。これは、1 で始まり 99 まで実行する PASS カウントを使用して行うことができます。SA z/OS は、メッセージまたは例外ごとに別々に PASS カウントを維持し、そのメッセージまたは例外が処理されるたびに PASS カウントを増分します。リカバリーが正常に行われた場合、PASS カウントのリセットについては、インストール・システムに責任があります。オプション CLEARING=YES で指定された場合、INGMON はメッセージと例外のコマンド処理を使用可能にして、PASS カウントをリセットします。

### リカバリー・ルーチンのタスク・グローバル変数

以下は、リカバリー・ルーチンがアクセスできるタスク・グローバル変数です。

タスク・グローバル変数	値
&EHKVAR1	モニター名を含みます。
&EHKVAR2	現在のヘルス状況を含みます。
&EHKVAR3	以前のヘルス状況を含みます。
&EHKVAR4	ヘルス状況に関連したメッセージを含みます。
&EHKVAR5	モニターのオブジェクト名を含みます。
&EHKVAR6	ジョブ名を含みます。
&SUBSAPPL	モニター名を含みます。
&SUBSTYPE	ストリング MONITOR を含みます。

## アクティブ・ヘルス・モニター

一般に、モニター・コマンドは、データの生成、データの処理、そして戻りコードの設定を行うために 1 つ以上のコマンドを発行する必要があります。戻りコードは、リソースのヘルス状況を判別するために SA z/OS によって使用されます。

使用される戻りコードと対応するヘルス状況を、[27 ページの表 6](#)に記載します。



戻りコード	ヘルス状況	説明
1	BROKEN	モニターがリカバリー不能エラーを検出しました。SA z/OS はモニターを停止します。
2	FAILED	モニターは現在、ヘルス状況を取得することができません。問題が消滅する可能性があるため、SA z/OS はモニターをアクティブのまま保持します。
3	NORMAL	モニターが、モニター対象オブジェクトの通常操作を検出しました。
4	WARNING	モニターが、モニター対象オブジェクトの操作で、ある程度の低下を検出しました。
5	MINOR	WARNING と同じですが、それより重大です。
6	CRITICAL	MINOR と同じですが、それより重大です。
7	FATAL	CRITICAL と同じですが、それより重大です。
8	DEFER	内部で使用されます。

ヘルス状況値は、自動化マネージャー内の複合状況に影響を与えます。

ほとんどのモニター・コマンドは、UNKNOWN、NORMAL、および WARNING 状況を使用します。MINOR、CRITICAL、および FATAL の状況は、問題がさらに悪化していることを示す場合の変化度で使用されます。BROKEN および FAILED は、モニターそのものの状況を説明する状況であり、モニター・コマンドでエラーが発生した場合に使用されることがあります。FATAL のヘルス状況によって、自動化リカバリーの一環としてアプリケーションの移動がトリガーされます。

FATAL は、モニターに関連付けられたアプリケーションに対する保証された自動 ForceDown とフェイルオーバー (使用可能な場合) です。

オプションで、モニター・ルーチンは、モニターを起動した SA z/OS プロセスによってトラップされる条件を説明するメッセージを発行することができます。このメッセージは、DISPMTR パネルに表示することができます。

各モニター・コマンドには、いくつかの基本的なステップが必要です。

1. データを収集し、結果を検索するために 1 つ以上のコマンドを発行します。
2. コマンド (複数の場合もある) の結果に基づいて、戻りコードを 1 から 8 の値に設定し、オプションでその値に基づいた処理を実行します。
3. (オプション) ヘルス状況に関する、より詳しい記述情報をメッセージで提供します。このメッセージは、DISPMTR コマンドを使用して表示することができます。
4. 戻りコードを伴って終了し、SA z/OS がヘルス状況を適切に設定できるようにします。

28 ページの図 5 は、PIPE 内で NetView PING コマンドを使用してリモート・システム上の TCP/IP スタックの状況を照会する例です。入力すると IP アドレスが渡されます。このルーチンは、メッセージ BNH770I に指定された要求の平均の往復時間 (RTT) を使用してヘルスを判別します。

```

/*REXX MYMON */
Arg parm
monrcs='BROKEN FAILED NORMAL WARNING MINOR CRITICAL FATAL DEFER'
'PIPE (STAGESEP | NAME PING)',
'| NETV PING' parm,
'| LOCATE 1.8 /BNH770I /',
'| STEM out.'
if out.0 = 0 then
  lrc = wordpos('FATAL',monrcs)
else
  do
    parse var out.1 . 'averaging' ms 'ms' .
    say 'PING lasted' ms 'ms'
    select
      when ms < 10 then lrc = wordpos('NORMAL',monrcs)
      when ms < 20 then lrc = wordpos('WARNING',monrcs)
      when ms < 30 then lrc = wordpos('MINOR',monrcs)
      when ms < 40 then lrc = wordpos('CRITICAL',monrcs)
      otherwise lrc = wordpos('FATAL',monrcs)
    end
  end
end
Return lrc

```

図 5. サンプル・モニター・コマンド

## 受動のイベント・ベース・ヘルス・モニター

### 概要

受動のイベント・ベース・モニターを使用すると、イベント (例えば、メッセージ、OMEGAMON XE シチュエーション、または CICSplex SM イベント) に直接対応することができます。アクティブ・ヘルス・モニターとは異なり、SA z/OS は、モニター対象オブジェクトの状況を定期的に照会する必要はなく、上記のようなイベントが発生したときのみ通知を受け取ります。

モニター・リソース用の MONITOR INFO ポリシー項目内での定義により、モニター・リソースのバインド先となるオブジェクト、およびオプションで、モニター・リソースがイベントを受け入れる元となるジョブを定義できます。

モニター・リソース用の**モニター対象オブジェクト**の指定は、モニター・プロセスに必要とされる任意の命名規則に準拠させることができます。例えば、CICS モニターの場合は、接頭部 CPSM があり、その後に CICS 名、タイプ (接続など)、および名前が続きます。例えば、CT12 と呼ばれるリンクの場合は、モニター対象オブジェクトは次のように呼ばれます。

```
CPSM.CICSTOR1.CONNECT.CT12.
```

一方、OMEGAMON XE シチュエーションのモニターの場合は、接頭部 ITM があり、その後にシチュエーション名が続いて、例えば、ITM.MYAUSSHORTAGE\_WARN のようになります。

モニター・リソースごとに定義できるモニター対象オブジェクトは 1 つのみですが、1 つのモニター対象オブジェクトに複数のモニター・リソースをバインドすることができます。例えば、複数の IMS モニターがオブジェクトとして OLDS を指定することができます。

また、モニター・リソースがイベントを受け入れる元の**モニター対象ジョブ名**をオプションで指定することもできます。したがって、例えば IMS モニター・リソースの場合は、モニター・リソース MTR1 にジョブ名 IMS1 を、MTR2 に IMS2 をそれぞれ指定することができます。イベントが OLDS に到達し、発行元が IMS1 が場合は、MTR1 のみが影響を受けます。

### イベント・タイプ

最も単純な事例では、イベントは、ジョブが発行するプレーン・メッセージで表されます。ユーザーがモニター対象ジョブ名も指定した場合を除き、特定のメッセージに登録されるすべてのモニター・リソースは、このメッセージを受け入れます。

他の事例、例えば OMEGAMON XE シチュエーションまたは CICSplex SM が報告するイベントの場合、イベントは、ヘルス・モニターのための目的で SA z/OS が提供するトリガー・メッセージによって表されます。これで、SA z/OS は、モニター対象オブジェクト名またはジョブを含んだこのメッセージ ING150I を使用して、モニター・リソースを見つけ、さらにヘルス状況を設定したり、コマンドを発行したりすることができます。これにより、SA z/OS によるさまざまなモニター・イベントの処理が可能になります。

ヘルス・モニターを行うコマンドである INGMON は、ING150I またはモニター・リソースが登録されている他のすべてのメッセージが発行される時は常に、NetView® 自動化テーブルから呼び出されます。INGMON は、指定されたモニター対象オブジェクトまたはジョブに対するモニター・リソースを見つけた後に、コード・マッチング・テーブルで、トリガーとなるイベントが発生した場合には常に発行する必要があるヘルス状況またはコマンド (あるいは、その両方) を検索します。

### イベント・トリガー・メッセージのコード・マッチング

INGMON を使用すると、イベント・トリガー・メッセージが出された場合に発行される特定のコマンドのセットを判別するために使用されるコード (指定されている場合、最大で3つまでの) を渡すことができます。

メッセージ ING150I の場合、SA z/OS は自動化テーブル項目を作成し、**Code 1** を使用してイベントの重大度によってコマンドを選択します。その他のメッセージの場合、デフォルトの自動化テーブル項目をオーバーライドし、必要に応じて、**Code 1**、**Code 2**、および **Code 3** に適切なトークンを渡すことができます。

どちらの場合も、「**Value Returned**」フィールドには、ブランクで分離された1つまたは2つのトークンが入ります。最初のトークンは、必要なコマンドの選択機能であり、次のいずれかを指定できます。

#### selection

選択が指定されているコマンド、または選択が指定されていないコマンドを実行します。

#

パス処理を実行し、現行パスと一致するすべてのコマンドを実行します。

#### #selection

selection を別の疑似メッセージ ID として解釈します。このメッセージのパス処理を実行し、現行パスと一致するすべてのコマンドを実行します。

これは、イベント・トリガー・メッセージ (例えば、ING150I) のためのパス処理に有効です。WARNING および CRITICAL にそれぞれ1つの項目があるとします。ING150I に対するパス処理を行う場合、例えば、最初の CRITICAL イベントを受け取ったときにパス・カウンターが5になっていることがあります (既に4つの WARNING イベントを受け取っていたため)。

ただし、#selection を使用すると、例えば、返される値 #MYWARN WARNING および #MYCRIT CRITICAL を対応するレベルに対して指定することができます。INGMON は、疑似メッセージ MYWARN に対するパス処理を行い、WARNING イベントにヘルス状況 WARNING を設定します。CRITICAL イベントの場合、INGMON は、疑似メッセージ MYCRIT に対する独立したパス処理を行い、最終的に CRITICAL のヘルス状況を設定します。

疑似メッセージには IGNORE アクションを設定して、AT 項目が作成されないようにすることを覚えておいてください。

「Code Processing」パネルの「**Value Returned**」列の2番目のトークンは、設定されるヘルス状況 (オプション) を示します。これを指定する場合は、ブランクを使って選択基準と分離する必要があります。

## プログラミング手法

INGMON によって呼び出されたコマンドは、NetView SAFE、AOFMSAFE などを使用した呼び出しのトリガーとなったメッセージにアクセスすることができます。

```
/* MYCLIST, called by INGMON */
'PIPE SAFE AOFMSAFE | STEM MSG.'
If msg.0 > 0 Then
  msgtext = msg.1          /* first message line */
```

また、INGMON は、INGMON の呼び出し元となったメッセージまたは例外から派生するトークンを使用して、タスク・グローバル変数 &EHKVAR0、&EHKVAR1-9、および &EHKVART を充てんします。メッセージの場合、割り当てはメッセージ ID で開始し、例外の場合は例外 ID で開始します。

INGMON では、以下のタスク・グローバル変数も設定されます。

#### &SUBSAPPL

モニター名を含みます。

#### &SUBSTYPE

ストリング MONITOR を含みます。

**&SUBDESC**

モニター・リソースの説明を含みます。

以下の例は、メッセージ・トークンと例外トークンがこれらのタスク・グローバル変数に割り当てられる方法を示しています。

**例 1:**

```
$HASP9211 JES MAIN TASK NOT RUNNING. DURATION- hh:mm:ss.xx
```

タスク・グローバル変数	値
&EHKVAR0	\$HASP9211
&EHKVAR1	JES
&EHKVAR2	MAIN
&EHKVAR3	TASK
&EHKVAR4	NOT
&EHKVAR5	RUNNING.
&EHKVAR6	DURATION-
&EHKVAR7	hh:mm:ss.xx
&EHKVAR8 &EHKVAR9 &EHKVART	NULL

**例 2:**

```
ING080I CI2XREP/MTR/KEYA OMSY4MVS OMIIMVS XREP Number of Outstanding Replies = 4
```

タスク・グローバル変数	値
&EHKVAR0	XREP
&EHKVAR1	番号
&EHKVAR2	of
&EHKVAR3	Outstanding
&EHKVAR4	Replies
&EHKVAR5	=
&EHKVAR6	4
&EHKVAR7 &EHKVAR8 &EHKVAR9 &EHKVART	NULL

INGMON コマンドが出すコマンドを定義する場合、&EHKVARx 変数を使用して、メッセージまたは例外の対応するトークンを置換することができます。

INGMON が指定されたモニター対象オブジェクトまたはジョブ名 (あるいはその両方) に対するモニター・リソースを検索するときに、ユーザー指定の REXX 式を介してモニター・リソース処理を動的にスキップすることができます。このような REXX 式が存在しない場合、INGMON は、メッセージを発行したジョブに対して指定されたモニター対象オブジェクト名を使用してモニター・リソースを見つけ、自動化ポリシ

ーに定義されたヘルス状況設定とコマンドを実行します。自動化メッセージに対する MESSAGES/USER DATA ポリシー項目内の「User Defined Data」パネルに REXX 式を追加することにより、その REXX 式の結果に応じて、追加の処理を使用不可に設定することができます。

これを行うには、事前定義されたキーワード INGMON\_FUNCTION をキーワードとして指定し、任意の REXX 式を「ユーザー・データ処理」パネルの値として指定します。REXX 式の結果が偽（つまり、0）の場合、処理は停止され、その他の場合、INGMON 処理は継続されます。メッセージ ID MYMTR の以下の例では、モニター・リソースの処理を、共通グローバル変数 DAY\_OF\_WEEK で定義されている曜日に基づいて制御します。（処理が継続されるのは、現在の曜日が日曜日ではない場合のみです。）

事前定義キーワード	データ
INGMON_FUNCTION	cglobal('DAY_OF_WEEK') ≠ 'SUN'

モニター・リソースにモニター・コマンドが定義されているが、間隔が指定されていない場合、そのような受動モニター・リソースの初期ヘルス状況はモニター・リソースの開始時にのみ取得されます。その他のヘルス状況アップデートはすべて、モニター・リソースを登録したイベントから派生させる必要があります。

ただし、コマンド AOFRCMTR を実行することにより、任意の時点でモニター・コマンドを出すことができます。このコマンドには、パラメーターとして、モニター対象オブジェクト名（および、オプションでジョブ名）が要求されます。このコマンドは、対応するモニター・リソースを見つけ、モニター・コマンドが指定されている場合は、そのモニター・コマンドを出します。

AOFRCMTR の構文については、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照してください。

## OMEGAMON を使用したヘルス・モニター

SA z/OS を使用すると、IBM OMEGAMON 製品と対話して、アドレス・スペース、ミドルウェア、あるいはシステムのヘルス 状況を示す主要なパフォーマンス・インディケーターを収集できます。

以下のセクションでは、モニター・リソースを使用してこれらの製品と対話する方法を示します。

### 概要

SA z/OS OMEGAMON インターフェースによって、システム上の広い範囲のパフォーマンス・データを収集することができます。以下のパフォーマンス・モニター製品からデータを収集することができます。

- IBM OMEGAMON for z/OS
- IBM OMEGAMON for CICS on z/OS
- IBM OMEGAMON for IMS on z/OS
- IBM Tivoli OMEGAMON XE for Db2 Performance Expert on z/OS
- z/OS で稼働するその他の IBM Tivoli Monitoring 製品

例外分析は、システム内の事前定義しきい値をモニターする OMEGAMON 標準機能です。しきい値を超えている場合は、例外分析が起動されるたびに、例外が OMEGAMON コンソールに表示されます。次に SA z/OS を使用して、exec を実行するかまたはコマンドを発行して、これらの例外アラートに従うことができます。これにはホスト OMEGAMON に戻すコマンドの発行が含まれます。

シチュエーションは、例外と非常によく似ていますが、シチュエーションは、論理式の組み合わせ、さらには他の組み込まれたシチュエーションに基づいています。IBM Monitoring インフラストラクチャーに基づく各製品 (IBM OMEGAMON など) は、そのまま使用したり、あるいはユーザーの希望どおりに変更したりすることができる事前定義されたシチュエーションのセットを提供します。また、ユーザー自身のシチュエーションを作成し、モニターを、ユーザー固有の要求に合わせて調整することもできます。シチュエーションは、Tivoli Enterprise Portal (TEP) で編集し、表示することができます。反射的自動化と呼ばれる TEP 機能を使用して、SA z/OS に特定のシチュエーションについて通知し、それに基づいて処理することができます。

IBM Monitoring サービスを使用すると、Tivoli Enterprise Monitoring Server (TEMS) 上の標準化された SOAP サービス・インターフェースを介して、このインフラストラクチャーに基づく製品の 1 つ 1 つと対

話することもできます。SOAP サービスは、例えば IBM OMEGAMON for z/OS (以前の IBM Tivoli OMEGAMON XE on z/OS) によって収集された特定のオブジェクトからデータを取得するために存在します。他のサービスを使用して、シチュエーションと TEP ワークフロー・ポリシーを自動的に管理したり、汎用メッセージを汎用メッセージ・コンソールに送信したりすることができます。

以下のように、モニター・リソースをセットアップすることができます。

- 関心のある例外のセットを、アクティブ・モニター・リソースを使用してモニターし、それらの例外の存在に基づいて、アプリケーションのヘルス状況を設定する
- これらの例外を起こす条件に対応し解決する
- 関心のあるシチュエーションのセットを、受動モニター・リソースを使用してモニターし、アプリケーションのヘルス状況を設定して、それらのシチュエーションを発生させる条件に対応して解決する

### 前提事項

IBM OMEGAMON 標準モニター および IBM Tivoli Monitoring 製品 (OMEGAMON XE など) を使用した SA z/OS では、さまざまなトポロジーが可能です。

- 1つのシステムにつき 1つ以上のモニター製品が使用可能。
- OMEGAMON 標準モニター の場合は VTAM<sup>®</sup> および NetView Terminal Access Facility (TAF)、OMEGAMON XE の場合は TCP/IP を介した接続。
- TEMS SOAP サーバーが、ローカルで、リモート・システム上で、または分散システム上で稼働している。
- SA z/OS は以下のいずれかのフォーカル・ポイントとして振る舞うことが可能。
  - グローバルに、異なるシステム上で実行されているモニター製品からのデータをモニターする
  - ローカルに、ローカル・システム上で実行されているモニター製品からのデータをモニターする

OMEGAMON 標準モニター との対話に採用可能なトポロジーについては、以下の前提事項があります。

1. OMEGAMON 製品が、MVS と CICS、Db2、または IMS がインストールされている各システムにインストールされている。
2. OMEGAMON モニターがインストール済みで、このモニターへの複数の VTAM ベースの接続をサポートするように既に構成されている。SA z/OS とのインターオペラビリティのために、タイプ 3270 モデル 2 (24x80) の論理装置 (LU) が必要です。
3. OMEGAMON モニターが、IBM SecureWay Security Server for z/OS (前の RACF<sup>®</sup>) のような外部セキュリティ製品と対話するようにセットアップされている。
4. OMEGAMON 内で定義されたしきい値を超えた場合、OMEGAMON 例外が報告されます。このしきい値は、インストール済み環境内で一致しなければなりません。アラートが出される可能性のある最も小さな重大条件を満たす必要があるからです。

OMEGAMON XE との対話については、以下の前提事項があります。

1. 反射的自動化が、対応するシチュエーション・イベントを作成した OMEGAMON XE エージェント上で実行されている。

### OMEGAMON 対話

以下のサブセクションでは、OMEGAMON 標準モニター 対話の場合に、1つ以上の OMEGAMON セッション、および SA z/OS カスタマイズ・ダイアログを使用してネットワーク通信を取り扱うように指定された自動化機能を定義済みであることを前提としています。

OMEGAMON セッションの定義についての詳細は、「*IBM Z System Automation 自動化ポリシーの定義*」で説明されている Network (NTW) 項目タイプ内の OMEGAMON SESSIONS と AUTHENTICATION ポリシー項目、および Auto Operators (AOP) 項目タイプ内の OPERATORS ポリシーを参照してください。

SOAP サービスを使用する OMEGAMON XE 対話の場合、各 SOAP サーバーを、接続する先の自動化ポリシー内に指定する必要があります。SOAP サーバーの定義についての詳細は、「*IBM Z System Automation 自動化ポリシーの定義*」で説明されている Network (NTW) 項目タイプ内の SOAP SERVER ポリシー項目を参照してください。

### INGOMX プログラミング・インターフェースの使用

INGOMX は、オペレーター (または自動オペレーター) と OMEGAMON の間のインターフェースとして動作します。これには、CICS、Db2、IMS、および MVS 用の標準 OMEGAMON モニターだけでなく、z/OS 上で稼働する OMEGAMON XE モニターおよびその他の IBM Monitoring 製品も含まれます。

標準 OMEGAMON モニターの場合、INGOMX を使用して、OMEGAMON の Major、Minor、および Immediate コマンドを発行し、OMEGAMON 例外分析によって報告される例外リストから、関心のある 1 つ以上の例外をフィルタリングすることができます。それぞれの要求は、OMEGAMON モニターで作成されたフォーマットでコンソールに書き込まれます (ただし、NetView に公開されません)。例外フィルタリングを要求したときに、フィルター基準 (XTYPE) が一致している場合は、1 つの例外についての複数の例外行が単一行に結合されて、単一メッセージとしてコンソールに書き込まれます。INGOMX は、NetView PIPE 内で使用するのに最適です。

INGOMX SOAP インターフェースを使用すると、例えば、以下を行うために、TEMS SOAP サーバーでサポートされるすべての SOAP サービスを発行することができます。

- 特定の OMEGAMON XE オブジェクトからの属性の (例えば、OMEGAMON XE for z/OS のオブジェクト Address\_Space\_CPU\_Utilization からの Job\_name および CPU\_percent の) 取得
- シチュエーションおよび TEP ワークフロー・ポリシーの開始と停止
- 汎用メッセージの発行
- イベントの IBM Monitoring プラットフォームへの送信

SOAP サービスの全セットおよび XML 構文の記述は、「IBM Monitoring 管理者ガイド」に記載されています。

以下の例では、INGOMX の使用を示しています。これらは、OMSY4MVS という名前を持つ OMEGAMON for MVS セッションを基にしています。同じ技法は、その他の OMEGAMON モニターにも適用されます。詳しくは、「IBM Z System Automation プログラマーズ・リファレンス」を参照してください。

#### 例 1. CSAA コマンドを使用した共通ストレージ使用率の戻り情報

```
INGOMX EXECUTE,NAME=OMSY4MVS,CMD=CSAA
| IPXNG      CSAA
SUMMARY
| IPXNG
+
| IPXNG      +
System
| IPXNG      +          Maximum  Pre-CSAA  Orphan
Usage
| IPXNG      +          -----  -----  -----
| IPXNG      +          0    2    4    6    8    100
| IPXNG      +  CSA    3312K  1247K          0    1247K    37.6%|----->
| IPXNG      +  ECSA  307740K  78797K          0    78797K    25.6%|---->
| IPXNG      +   SQA    1620K    660K          0    660K    40.8%|----->
| IPXNG      +  ESQA  145696K  23930K          0    23930K    16.4%|-->          |
```

#### 例 2. OMEGAMON コマンド修飾子の使用

```
INGOMX EXECUTE,NAME=OMSY4MVS,CMD=ALLJ,MOD=#
| IPXNG      #ALLJ      166
INGOMX EXECUTE,NAME=OMSY4MVS,CMD=ALLJ,MOD=<
| IPXNG      <ALLJ *MASTER*  PCAUTH      RASP      TRACE  DUMPSRV  XCFAS      GRS  SMSPDSE
+
| IPXNG      +          CONSOLE      WLM  ANTMAIN  ANTA000  OMVS  IEFSCAS  JESXCF  ALLOCAS
+
| IPXNG      ...
```



### 例 3. 未解決のオペレーター応答のトラップ

```
INGOMX TRAP,NAME=OMSY4MVS,XTYPE=(XREP)
| IPXNG      + XREP Number of Outstanding Replies = 5
```

### 例 4. OMEGAMON Minor コマンドの発行

```
/* REXX-Routine EXMINOR
*/
cmd.1 = "CMD=SYS"      /* Major command, issued ahead of its minors */
cmd.2 = "CMD=FCSA"    /* Minor: CSA frames below 16M */
cmd.3 = "CMD=FCOM"    /* Minor: CSA, LPA, SQA, and nucleus below 16M */
cmd.0 = 3
'PIPE STEM cmd. COLLECT',
'| NETV INGOMX EXECUTE,NAME=OMSY4MVS,CMD=*',
'| CONSOLE ONLY'
* IPXNG      EXMINOR
| IPXNG      SYS      >> WLM Goal mode OPT=00 SYSRES=(150526,8812) <<
| IPXNG      fcsa     328      1312 K
| IPXNG      fcom     849      3396 K
```

INGOMX を使用する前に、オペレーターと特定の OMEGAMON モニターの間のセッションを明示的に確立する必要はありません。このようなセッションは、最初の使用時に自動的に確立されます。

個別の OMEGAMON セッションまたはコマンド (あるいはその両方) の選択的保護が、NetView コマンド権限テーブルに基づいて可能です。詳細については、「IBM Z System Automation 計画とインストール」の付録『セキュリティと許可』を参照してください。

SOAP サービスを使用する (例えば、OMEGAMON XE オブジェクトから特定の属性を取得する) には、最初に XML 文書形式で要求のパラメーターを記述する必要があります。SOAP サーバーにより、XML 文書の妥当性検査が行われ、不正確あるいは不完全な XML 文書が検出された場合はリジェクトされます。XML は大/小文字を区別する文書記述言語であるため、「<」と「>」で囲まれた名前のスペルは重要です。また、すべての XML 文書の構造は階層構造であるため、各要素は、オープン名 (例えば、「<CT\_Get>」) とそれに対応する、名前の前にスラッシュを付けて示されるクローズ名 (例えば、「</CT\_Get>」) で囲む必要があります。

以下の例は、CPU パーセントが 1.0 より大きいすべてのジョブについて、OMEGAMON XE for z/OS オブジェクトである Address\_Space\_CPU\_Utilization から、Job\_Name、アドレス・スペース ID (ASID)、および CPU\_Percent 属性を取得する要求パラメーターを説明しています。この例では、照会されたオブジェクトは KEYAS:CMS と呼ばれる TEMS 上で収集されます。

```
<CT_Get>
  <target>KEYAS:CMS</target>
  <object>Address_Space_CPU_Utilization</object>
  <attribute>Job_Name</attribute>
  <attribute>ASID</attribute>
  <attribute>CPU_Percent</attribute>
  <afilter>CPU_Percent;GT;10</afilter>
</CT_Get>
```

この XML 文書は、INGOMX で順次データ・セットまたは区分データ・セットをポイントすることにより、あるいは、INGOMX が NetView PIPE 内で起動されるとすれば、デフォルトの SAFE に渡すことができます。

INGOMX を起動するときは、接続される SOAP サーバーを指定する必要があります。以下の例では、SA z/OS カスタマイズ・ダイアログを使用して、Network (NTW) 項目タイプの SOAP SERVER ポリシー項目内に KEYAYA と呼ばれる SOAP サーバーを定義したと想定しています。この定義には、ホストの名前または IP アドレス、SOAP サーバーのポート、および SOAP サービスのパス名が含まれます。上記の要求パラメーターは、区分データ・セット SYS1.SOAP.DATA のメンバー GETCPU 内にあります。

```
soapds = 'SYS1.SOAP.DATA(GETCPU)'
soapsrv = 'KEYAYA'
Address NETVASIS 'PIPE (END % NAME GETCPU)',
'| NETV (MOE) INGOMX SOAPREQ SERVER='soapsrv' DATA='soapds,
```



```
'| L: LOC 1.8 'd||'DW0369I '||d,
'| EDIT SKIPTO 'd||'RETURN CODE' ||d,
'|      UPTO 'd||'.' ||d,
'|      WORD 3 1',
'| VAR omx_rc',
'| %L:',
'| CON ONLY'
```

正常に INGOMX に戻ると、SOAP サーバーの出力が複数行の ING160I メッセージで戻されます。

```
ING160I RESPONSE FROM SOAP SERVER: 9.xxx.xxx.xxx:1920///cms/soap
Job_Name:ASID:CPU_Percent
IXGLOGR:20:2.1
NET:59:2.1
RMFGAT:89:6.9
SDM1IRLM:108:1.7
BBOS001S:113:22.1
YANAMSJH:117:3.9
```

このメッセージの最初の行は、応答した SOAP サーバー (この例では、KEYAYA) の IP アドレスを示します (IP アドレスは匿名化されています)。

2 番目の行は、SOAP サーバーによって戻された属性の名前を示します。各属性名は、印刷不能文字 X'FF' (: で表されます) によって互いに分離されます。

3 番目の行およびそれに続くすべての行には、要求された実際のデータが含まれます。属性値は、2 番目の行の中の対応する属性名と同じ順序で表示されます。また、属性名と同様に属性値も、印刷不能文字 X'FF' (: で表されます) によって互いに分離されます。

このメッセージの表形式の構造により、NetView PIPE 内でメッセージを容易に処理することができます。

#### INGMTRAP モニター・コマンドの使用

INGMTRAP は INGOMX へのカスタマイズされたインターフェースであり、OMEGAMON 例外分析によって報告された関心のある例外のフィルタリング機能、およびこのような例外のための自動化のトリガーを提供します。呼び出し側によって提供された XTYPE フィルターに一致するそれぞれの例外に対し、INGMTRAP はメッセージ ING080I を発行します。このメッセージは、以下のように NetView に公開されます。

例:

```
ING080I CI2XREP/MTR/KEYA OMSY4MVS OMIIMVS XREP Number of Outstanding Replies = 4
```

呼び出し側によって提供された XTYPE フィルターと一致する例外がない場合、INGMTRAP は ING081I メッセージを作成します。このメッセージは NetView には公開されませんが、モニター・リソースのログに書き込まれ、例外が検出されなかったことが記録されます。例:

```
ING081I CI2XREP/MTR/KEYA OMSY4MVS OMIIMVS NO EXCEPTION FOUND
```

INGMTRAP はモニター・コマンドとしてのみ使用できます。これは、INGMTRAP を、モニター・リソースの定義内でモニター・コマンドとして直接指定する必要があるということ、あるいはそのようなモニター・コマンドの代わりに呼び出す必要があるということを意味します。以下の例は、OMEGAMON for MVS セッション OMSY4MVS で報告される未解決のオペレーター応答をトラップするために、項目タイプ Monitor Resource (MTR) の MONITOR INFO ポリシーで指定する必要があるものを示しています。

```
INGMTRAP NAME=OMSY4MVS,XTYPE=(XREP)
```

例外のリストを指定する際は、慎重に行ってください。それぞれの例外が ING080I メッセージを発行させる原因となる可能性があります。それぞれの ING080I メッセージの出現が、モニター・リソースのヘルス状況処理のトリガーとなるので、これがモニター・リソースの最終ヘルス状況に与える影響を確実に理解しておく必要があります。

INGMTRAP についての詳細は、「IBM Z System Automation プログラマーズ・リファレンス」を参照してください。モニター・リソースの定義についての詳細は、「IBM Z System Automation 自動化ポリシーの定義」を参照してください。

## OMEGAMON 例外に基づいたヘルス・モニター

このセクションでは、カスタマイズ・ダイアログを使用して、OMEGAMON 例外に基づいたヘルス・ベース・モニター用のモニター・リソースをセットアップする方法を説明し、サンプル・シナリオを提供し、OMEGAMON をモニター・リソースと組み合わせて使用する際の推奨事項を記載します。

### モニター・リソースの定義

モニター・リソースと 32 ページの『OMEGAMON 対話』に説明されている OMEGAMON 対話方式を結合することにより、OMEGAMON で報告された出力の分析の結果として、かつ、適切なヘルス状況を設定することによって、自動化をトリガーすることができます。

OMEGAMON 例外は、モニター・リソースとモニター・コマンド INGMTRAP を使用して、定期的にモニターすることができます。このような例外を取り扱うためのさまざまな方法があります。

1. カスタマイズ・ダイアログ内で、所与のモニター・リソースの MESSAGES/USER DATA ポリシーに、INGMTRAP がモニターするようにセットアップされたそれぞれの例外のヘルス状況を指定する必要があります。メッセージとは異なり、OMEGAMON 例外は、空白と次に 4 文字の OMEGAMON 例外 ID があとに続く「+」記号で表されます。
2. ヘルス状況に加えて、特定の例外を処理するために 1 つまたは複数のコマンドの集合を指定することができます。コマンドは、アプリケーション (APL) のような、MESSAGES/USER DATA ポリシーが提供されるその他のリソースと同じ方法で処理されます。これには、PASS カウントに基づく拡大処理、またはメッセージから派生した CODE を使用して定義できる選択値に基づく処理が含まれます。
3. HEALTHSTATE ポリシーを使用して、ヘルス状況が変更するたびに、OMEGAMON 例外の代わりにリカバリー・コマンドを発行することができます。

どの方法、または方法の組み合わせが選択されても、例外を取り扱う処理のトリガーとなるのは、特定のモニター・リソースおよび例外に対する ING080I メッセージの出現です。MESSAGES/USER DATA ポリシー内の定義から作成された自動化テーブルには、INGMON コマンドを起動して、モニター・リソースのヘルス状況を設定し、さらに例外に回答してコマンドを発行するステートメントを含みます。ほとんどの場合、NetView 自動化テーブルに必要な項目は、SA z/OS によって自動的に作成されます。まれなケースとして、例えばコマンド選択が CODE を基にしなければならないとき、例外の自動化テーブル定義をオーバーライドして、INGMON の起動時に最大 3 つのコード (CODE1、CODE2、および CODE3) を指定する必要があります。

代わりに、インストール先作成のモニター・コマンドを使用して、1 つ以上の OMEGAMON モニターに対して一連の例外の INGOMX を発行することができます。このようなモニター・コマンドは、INGOMX によって作成された出力の分析を基にした適切なヘルス状況で戻ります。ヘルス状況変更時に発行されるリカバリー・コマンドは、そのモニター・リソースの HEALTHSTATE ポリシー内で指定されます。

### シナリオの例

SA z/OS と OMEGAMON が一緒にどのように動作するかを示すため、以下のようなシナリオを考えてみます。

連続的にモニターする必要のある DB2 アプリケーションがあるとします。1 次アクティブ・ログの可用性に特別な関心があります。LOGN 例外は、それぞれのしきい値によって指定されるよりも少ない 1 次アクティブ・ログが存在することを示しています。これは、最後の 1 次アクティブ・ログが 100% フルになった場合に、DB2 ハング状態の原因となるので、クリティカル・ヘルス・インディケーターと考えられます。このような状態は、1 つ以上の追加の 1 次アクティブ・ログを再度使用可能にすることでのみ解決できます。

この状態をモニターし、それに対して適宜対応するためには、自動化ポリシーを変更する必要があります。最初に、まだ存在しない場合は、OMEGAMON for DB2 モニター用のセッション属性を定義して、VTAM 接続を確立できるようにします。OMEGAMON セッションは、そのセッション名によって参照されます。次に、VTAM セッション・トラフィックを処理するために開始される、セッション・オペレーター (自動化オペレーター) の数を検討します。より高度な並列処理が必要な場合には、数を追加します。セッション・オペレーターと事前定義の NetView タスクの数が等しいことを確認する必要があります。

次に、この OMEGAMON セッションから定期的に例外情報を要求する新規のモニター・リソース (MTR) を追加します。モニター対象の DB2 サブシステムとの *HasParent* 関係を使って、MTR を追加します。これにより、DB2 サブシステムが開始されると MTR が確実に活動化され、DB2 サブシステムが停止されると

MTR が確実に非活動化されるようになります。また、DB2 サブシステムに対する *HasMonitor* 関係を使用して MTR を定義し、モニターのヘルス状況が確実にアプリケーションに伝搬されるようにします。

MTR は、それがアクティブである間、モニター・コマンド INGMTRAP を使用して、現存している OMEGAMON 例外を、OMEGAMON for DB2 のインストール用プロファイル内で定義されたしきい値を基にして収集します。INGMTRAP は OMEGAMON によって戻されたすべての例外を分析し、MTR が関心を持つこれらの例外 (この例では LOGN) にフィルターを掛けます。SA z/OS は引き続きメッセージ ING080I を発行し、例外処理を開始します。

最後に、REXX 自動化プロシージャを実行する新規ルールも (SA z/OS ポリシーを使用して) NetView 自動化テーブルに追加します。この新規ルールにより、LOGN 例外が報告され、ヘルス状況が CRITICAL (6) であるときはいつでも、1 次アクティブ・データ・セットのプールに新規のログ・データ・セットが追加されます。MTR のヘルス状況は、使用可能な 1 次アクティブ・ログの数が 1 に等しい場合、CRITICAL と見なされます。LOGN 例外が次のモニター間隔で再度報告されると、自動化テーブルの 2 番目のルールが MTR のヘルス状況を FATAL (7) に設定します。通常のリカバリー処理がこれ以上動作しないようであるため、この設定がアプリケーション移動のトリガーとなります。さらに、この状態をオペレーターに伝えるために、アラートが送信されます。LOGN 例外が報告されなくなれば、MTR のヘルス状況は NORMAL (3) に設定されます。

自動化テーブルによって MTR に割り当てられたヘルス状況は、この MTR を所有する DB2 アプリケーションに伝搬します。したがって、DB2 サブシステムに支障がないかどうかは、一目でわかります。

### 推奨事項

OMEGAMON をモニター・リソースと組み合わせて使用する場合は、以下の推奨事項を考慮する必要があります。

- INGMTRAP を使用して複数の例外をモニターすることを避ける。トリップするかもしれない例外が複数ある可能性があります。従って、複数の ING080I メッセージが生成される可能性があることに注意してください。ただし、モニター・リソースのヘルス状況は、最後の ING080I メッセージによります。
- INGMTRAP を使用して異なるモニター・リソースによってモニターされている同じ例外に、異なるヘルス状況を設定することを避ける。このような例外のメッセージ ING080I を処理するために、SA z/OS によって自動化テーブル項目が 1 つだけ生成されることに注意してください。

これらのケースでは、複数の例外から結合されたヘルス状況を判別したり、または各モニター・リソースごとの個別のヘルス状況を判別したりするために、インストール先作成のモニター・コマンドから起動される INGOMX を使用する方が、INGMTRAP の使用よりもむしろ好ましいと言えます。

## OMEGAMON シチュエーションに基づいたヘルス・モニター

このセクションでは、受動の、OMEGAMON XE シチュエーションのイベント・ベース・モニターの概要を説明し、さらにカスタマイズ・ダイアログを使用してモニター・リソースをセットアップする方法を説明します。

### 概要

SA z/OS が標準 OMEGAMON モニターに使用する例外ベースのモニターとは異なり、IBM Monitoring インフラストラクチャーは、シチュエーションが発生したときにそれに対処する手段を提供します。Tivoli Enterprise Portal (TEP) では、ユーザーは、個々のシチュエーションごとにトリガーされる必要のある自動応答の種類 (反射的自動化) を指定することができます。

SA z/OS は、INGSIT と呼ばれる単純コマンドを備えることにより、この機能を利用します。ITM 管理者は、SA z/OS のヘルス・モニターまたはヘルス・ベースの自動化を起動する必要があるシチュエーションに対して、このコマンドを「Situation Editor」ダイアログを使用して TEP に入力します。INGSIT についての詳細は、「IBM Z System Automation プログラマーズ・リファレンス」を参照してください。

Take Action コマンドは、Tivoli Enterprise Monitoring Server (TEMS) が同一システム上で実行される場合を除いて、TEMS ではなくエージェント (例えば、OMEGAMON XE for z/OS) 上で実行されます。これは、ハブ TEMS が z/OS 上に常駐していないために、コマンドが送達されない可能性があるためです。

INGSIT は、個々のモニター・リソースのヘルス状況を設定できるようにするメッセージ ING150I をトリガーします。これで、自動的にシチュエーションを修正するために、コマンド (リカバリー・コマンドや通

知コマンドなど)を発行することが可能になります。ヘルス状況の定義および発行される関連コマンドは、カスタマイズ・ダイアログ内で指定できます。

### OMEGAMON XE シチュエーション用のモニター・リソースの定義

#### 手順

OMEGAMON XE シチュエーション用にモニター・リソースをセットアップするには、以下のようになります。

1. 応答したい OMEGAMON XE シチュエーションのそれぞれに対して 1 つの MTR を定義します。
2. MONITOR INFO ポリシー項目の以下のフィールドに入力します。

#### モニター対象オブジェクト

OMEGAMON XE シチュエーションの名前を、大文字で、ITM の接頭部を付けて入力します。例えば、ITM.MYSIT のようになります。

#### モニター対象ジョブ名

このシチュエーションを特定のモニター・リソースと突き合わせる、オプションのジョブ名を入力します。

3. 発行されるコマンドを作成し、重大度を有効なヘルス状況にマップするために、メッセージ ID ING150I 用のコードを MTR の MESSAGE/USER DATA ポリシー内に定義します。

#### シナリオの例

次のシナリオを検討してみましょう。

ページ・データ・セット使用率および操作可能でないページ・データ・セットに基づいて補助記憶域不足が検出されたときに PAGEADD コマンドが発行されます。

MyAuxShortage\_Warn と呼ばれるシチュエーションがインストール済み環境で定義されています。このシチュエーションは、OS390\_Local\_PageDS\_PctFull\_Warn および OS390\_PageDSNotOperational\_Warn の両方の事前定義済みシチュエーションが真であるときに、真になります。

反射的自動化として、管理対象システム (すなわち、シチュエーションが発生したシステム) 上で以下のシステム・コマンドが発行されます。

```
F NETV,INGSIT MyAuxShortage_Warn,warn
```

ここで、NETV は、NetView アドレス・スペースのジョブ名です。

このコマンドは、Take Action ダイアログから発行されます (39 ページの図 6 を参照)。

図 6. Take Action ダイアログ

INGSIT が呼び出され、ING150I メッセージを作成します。このメッセージには、モニター対象オブジェクトにマップされるシチュエーションの名前が入っています。その他のオプションの情報には、以下があります。

- シチュエーションの重大度
- このシチュエーションを特定のモニター・リソースと突き合わせるジョブ名
- イベント関連の情報を含むその他のデータ

上記の例には、シチュエーション MyAuxShortage\_Warn、およびその重大度 warn が含まれています。

カスタマイズ・ダイアログを使用して、そのモニター対象オブジェクトとして ITM.MYAUSSHORTAGE\_WARN (大文字で) を指定するモニター・リソース (例えば、AUXSHORT) を作成します。

次に、ING150I を AUXSHORT モニター・リソースの MESSAGE/USER DATA ポリシー項目に指定します。この例では、以下のコード項目を使用して、選択 ADD を導き出し、ヘルス状況を MINOR に設定することができます。

コード値	項目
コード 1	warn
コード 2	*
コード 3	*
返される値	ADD MINOR

また、1つ以上のコマンドを、コード・マッチング処理の結果生じた選択の ING150I に対して指定することができます。上記の例では、PAGEADD コマンドを選択 ADD に対して指定することになります。



選択に対して「コマンドの処理 (Command Processing)」パネルで指定したすべてのコマンドが実行された後、コード処理でマップされたヘルス状況が設定されます (この例では、MINOR)。ヘルス状況がコード・マッチング・テーブル内に指定されていない場合は、ヘルス状況は変更されません。

このシナリオをより高度に拡張すると、TEP に表示されるシチュエーション MyAuxShortage\_Warn は、SOAP サービスを使用して自動的に認知されます。これを行うには、小さな要求パラメーター XML-document を作成して、処理のために TEMS SOAP サーバーに送信する必要があります。シチュエーションを認知するには、以下の例に示すように、CT\_Acknowledge 要求を発行する必要があります。

```
<CT_Acknowledge>
  <target>KEYAS:CMS</target>
  <name>MyAuxShortage_Warn</name>
  <source>KEYAPLEX:SYS1:MVSSYS</source>
  <data>System Automation is taking care of this</data>
</CT_Acknowledge>
```

上記の XML-document は、シチュエーションを管理する TEMS (ターゲット)、シチュエーション自体 (名前)、およびこのシチュエーションのソースとなるモニター・エージェントと呼ばれるもの (ソース) を参照します。データ要素を使用して、任意の追加テキスト情報を、TEP 上でこのシチュエーションを調べているユーザーに渡すことができます。

32 ページの『OMEGAMON 対話』で説明されているように、INGOMX は、SOAP 要求を TEMS SOAP サーバーに出すために使用します。シチュエーションが認知されると、それは、TEP のシチュエーション・イベント・コンソールまたはナビゲーターの吹き出しリストでシチュエーションとして認識されます。

## CICSplex SM を使用したヘルス・モニター

このセクションでは、イベント・ベースの CICS モニターのコンポーネントを紹介し、さらにカスタマイズ・ダイアログを使用してモニター・リソースをセットアップする方法を説明します。

### コンポーネントの概要

イベント・ベースの CICS リンクおよびヘルス・モニターは、CICSplex System Manager (CICSplex SM) オブジェクトを使用して実装されます。CICSplex SM からイベントを受け取る時は常に、メッセージ ING150I が発行されます。

INGCPSM は、CICSplex SM のイベント・リスナーです。INGCPSM は長時間実行自動化プロシージャであるため、仮想オペレーター端末タスク (VOST) で実行する必要があります。始動時に構成をスキャンして、イベントを listen します。その後 INGCPSM は、構成に変更 (すなわち、モニター・リソースの追加、削除、または変更など) がないか、あるいは、モニター・リソースが初期モニターを待機中 (すなわち、STATUS=ACTIVE および HEALTH=UNKNOWN) ではないかを定期的に検査します。

### VOST を管理するためのアプリケーションの作成

タイプ NONMVS のアプリケーションを使用して、INGCPSM を実行する VOST を管理することができます。

- APL の開始コマンドとして INGVSTRT コマンドを使用することにより、VOST を開始します。この場合、そのジョブ名が、VOST の *attach\_name* として使用されます。
- 管理 APL で INGVSTOP 停止コマンドのシーケンスを使用して、VOST を停止します。
- 管理 APL で INGVMON モニター・ルーチンを使用して VOST の状況をモニターします。

詳しくは、「IBM Z System Automation プログラマーズ・リファレンス」を参照してください。

### モニター・リソースの定義

#### 手順

モニター・リソースをセットアップするには、次のようにします。

1. 各 CPSM オブジェクト (例えば、各接続) に MTR を 1 つ定義します。
2. MONITOR INFO ポリシー項目の「**モニター対象オブジェクト (Monitored Object)**」フィールドに、命名規則に従って、例えば CPSM.CICS1TOR.CONNECT.CON1 のように入力します。
3. 「**モニター対象ジョブ名 (Monitored Jobname)**」フィールドは、ブランクのままにしておきます。

4. 以下の例のように、CPSM 重大度を有効なヘルス状況にマップするために、メッセージ ID ING150I 用のコードを MTR の MESSAGE/USER DATA ポリシー内に定義します。

コード 1	返される値
VLS	* NORMAL
LS	* WARNING
LW	* WARNING
HW	* MINOR

### タスクの結果

2 つの CICS リソース間の接続をモニターするためのサンプル定義については、\*CICS アドオン・ポリシーを参照してください。

## JES3 コンポーネントのモニター

モニター・リソースの概念は、さまざまな JES3 コンポーネントのヘルスをモニターするために使用されます。SA z/OS には、モニター部分と結果のリカバリー処理の厳密な分離に対応する 2 つのコマンドがあります。

- AOFRJ3MN: JES3 環境内のコンポーネント、例えば、スプール・スペースなどをモニターするために使用されます。
- AOFRJ3RC: モニター対象の JES3 オブジェクトに対するリカバリー・アクションを実行するために使用されます。

以下の例で、スプール・スペース・モニターを定義します。

1. 対応する JES3 との「HasParent」関係を持つモニター・リソースを定義します。JES3 がアクティブの場合にのみスプール・スペースをモニターする意味があるからです。
2. MONITOR INFO ポリシー内の活動化コマンドおよび非活動化コマンドは、スプール・モニターには不要です。
3. MONITOR INFO ポリシーでモニター・コマンドとして AOFRJ3MN コマンドを使用し、必要なモニター間隔をセットアップします。この例では、60% までのスプール使用率が NORMAL、61-70% が WARNING、71-80% が MINOR、81-90% が CRITICAL、90% を超えると FATAL になります。

```
AOFRJ3MN JES3_subys SPOOLSHORT 60,70,80,90
```

4. リカバリー・アクションを HEALTHSTATE ポリシーに定義します。例えば、以下のようになります。

状態	コマンド・テキスト
NORMAL	AOFRJ3RC JES3_subsys SPOOLSHORT RESET
CRITICAL	AOFRJ3RC JES3_subsys SPOOLSHORT 05
FATAL	AOFRJ3RC JES3_subsys SPOOLSHORT 01

毎分 1 つのリカバリー・コマンドを発行します。これらのコマンドは、JES3 サブシステムの JES3 SPOOLSHORT CMDS ポリシーから読み取られます。スプール使用率が 60% を下回ると、ヘルス状況は NORMAL になります。これにより AOFRJ3RC コマンドが呼び出されますが、この場合は RESET オプション指定で呼び出されます (RESET オプションはリカバリーを停止します)。リカバリー・コマンドの自動オペレーターとして JESOPER を使用するようにお勧めします。SPOOLSHORT 状態のリカバリー・コマンドは、JES3 サブシステムに定義する必要があることに注意してください。

5. JES3 サブシステムの場合、SPOOLSHORT に対して実行すべき必要なアクションを JES3 SPOOLSHORT CMDS ポリシーに定義します。

パス	自動化機能	コマンド
1	JESOPER	<b>MVS &amp;SUBSCMDPFXF U,Q=HOLD,AGE=30D,N=ALL,C</b>
2	JESOPER	<b>MVS &amp;SUBSCMDPFXF U,Q=HOLD,AGE=10D,N=ALL,C</b>
3	JESOPER	<b>MVS &amp;SUBSCMDPFXF U,Q=HOLD,AGE=3D,N=ALL,C</b>
10	JESOPER	<b>MVS &amp;SUBSCMDPFXF U,Q=HOLD,AGE=1D,N=ALL,C</b>

これにより、最初のパスで、30 日より前のすべてのジョブが保留キューからページされます。2 回目のパスで、10 日より前のすべてのジョブがページされます。3 回目のパスで、3 日より前のすべてのジョブがページされます。最終的に、パス間隔 (この例では 5 分) が 10 回経過した後に、その間にリカバリー・アクションがリセットされない場合は、1 日より前のすべてのジョブが削除されます。

## AOFRJ3MN ルーチン

このルーチンを使用して、JES3 環境内の各種のオブジェクトをモニターします。

以下のオブジェクトをモニターできます。

- MDS キュー (取り出しキュー、検査キュー、待機ボリウム・キュー、エラー・キュー、割り振りキュー、ブレイクダウン・キュー、使用不能キュー、再始動キュー、システム選択キュー、システム検査キュー)
- 現行セットアップの深さ
- スプール・スペース

10 種の JES3 MDS キューのそれぞれについて、4 つのヘルス状況 (Warning、Minor、Critical および Fatal) の各状況ごとのしきい値を設定することができます。これは、特定のキューで含むことができるジョブ数を示し、その値になると対応するヘルス状況に設定されます。例えば、エラー・キューの WARNING しきい値が 5 に設定されている場合に、5 以上のジョブが MDS エラー・キューに保留されると、ヘルス状況は Warning に設定されます。

スプール・スペースの場合、しきい値はスペース使用量を定義し、これを超過すると対応するヘルス状況に設定されます。

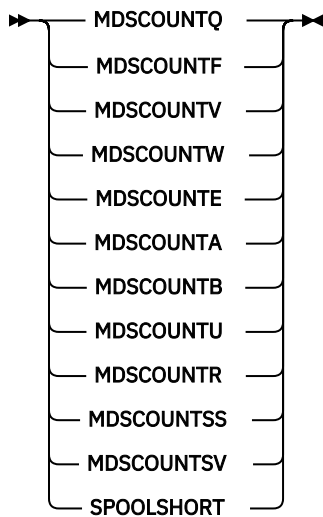
AOFRJ3MN は、呼び出されると、適切な JES3 コマンド (SPOOLSHORT の場合は \*I,Q,S で、MDS キューの場合は \*I,S) を出して、応答を解析します。メッセージ・テキストから抽出された値がしきい値と比較され、戻りコードが、対応するヘルス状況に設定されます。これは、モニター・リソース (MTR) のヘルス状況を設定するだけです。AOFRJ3MN ルーチンによってリカバリー・アクションが行われることはありません。各ヘルス状況にリカバリー・アクションを定義する必要がある場合には、モニター・リソースの HEALTHSTATE ポリシーを使用して定義します。

AOFRJ3MN ルーチンの構文は次のようになります。

▶▶ AOFRJ3MN — *jes3apl* — **object** — **threshold-list** ▶▶

**object**



**threshold-list**

➤ *warning ,minor ,critical ,fatal* ➤

**jes3apl**

このモニターが機能するカテゴリ JES3 の APL の名前を指定します。

**monitor**

モニターされる JES3 オブジェクトを指定します。

**MDSCOUNTQ**

現行セットアップの深さ

**MDSCOUNTF**

取り出しキュー

**MDSCOUNTV**

検査キュー

**MDSCOUNTW**

待機ボリューム・キュー

**MDSCOUNTE**

エラー・キュー

**MDSCOUNTA**

割り振りキュー

**MDSCOUNTB**

ブレークダウン・キュー

**MDSCOUNTU**

使用不能キュー

**MDSCOUNTR**

再始動キュー

**MDSCOUNTSS**

システム選択キュー

**MDSCOUNTSV**

システム検査キュー

**SPOOLSHORT**

スプール

**threshold-list**

4つのしきい値をコンマで区切ったリストを指定します。

**warning**

この値を超えた場合に、ヘルス状況を WARNING に設定します。



提供された JES2 スプール・モニター機能には、以下の項目が含まれます。

- 自動化ルーチン INGRMJSP、INGRCJSP (AOFRSD01)、AOFRSD09、および AOFRSD0H。195 ページの『INGRMJSP』、197 ページの『INGRCJSP (AOFRSD01)』、189 ページの『AOFRSD09』、および 193 ページの『AOFRSD0H』を参照してください。
- システム・メッセージ HASP050 および HASP355 用の自動化テーブル項目。
- JES2 アプリケーションの JES2 SPOOLSHORT および JES2 SPOOLFULL ポリシー項目内の JES2 スプール・リカバリー処理用の構成パラメーター。

## CICS および IMS 用の外部サブシステム接続モニター

SA z/OS では、CICS および IMS 用の外部サブシステムである DB2® MQ および IBM® MQ に対する接続をモニターできます。

### CICS

各モニター間隔後、定期的に CICS コマンド CEMT INQUIRE DB2CONN または CEMT INQUIRE MQCONN が発行され、CICS DB2 または CICS MQ 接続の状況を照会します。

詳しくは、「*IBM Z System Automation* 製品自動化プログラマーズ・リファレンスとオペレーターズ・ガイド」のセクション『CICS - 外部サブシステム接続のモニター』および『CICS - 外部サブシステム接続のモニター用の INGRMCDB ルーチン』を参照してください。

### IMS

各モニター間隔後、定期的に IMS コマンド DISPLAY SUBSYS が発行され、DB2® または IBM® MQ 外部サブシステムへの接続状況に関して、このコマンドへの応答が分析されます。

詳しくは、「*IBM Z System Automation* 製品自動化プログラマーズ・リファレンスとオペレーターズ・ガイド」のセクション『IMS - 外部サブシステム接続のモニター』および『IMS - 外部サブシステム接続のモニター用の INGRMIDB ルーチン』を参照してください。

## IMS コンポーネントのモニター

IMS 自動化の場合、SA z/OS は、IMS 制御領域のオンライン・ログ・データ・セット (OLDS) およびリカバリー管理データ・セット (RECON) のモニターを使用可能にして、VTAM アプリケーション制御ブロック (ACB) およびログオンの使用可能化の状況検査が行えるようにします。

このために提供されるモニター・ルーチンおよびモニター機能を使用可能にするために必要な定義は、「*IBM Z System Automation* 製品自動化プログラマーズ・リファレンスとオペレーターズ・ガイド」で説明されています。



## 第4章 ジョブ・ログ・モニター

この情報は、ジョブ・ログ・モニターのコンポーネントを紹介し、カスタマイズ・ダイアログを使用してモニター機能をセットアップする方法について説明します。

### 概要

ジョブ・ログ・モニターは、JES スプール・ファイルのみをモニターするように設計されています。現在の実装では、ジョブが実行中であるか、終了しているかに関係なく、任意の JES スプール出力ファイルがサポートされています。ただし、後者の場合は、出力が JES 出力キューにまだ保持されている必要があり、ユーザーはその出力が 1 回のみ処理されるようにしなければなりません。

ジョブ・ログ・モニターを使用するには、NetView に JES ジョブ ID が必要です。ユーザーが DSIRQJOB タスクを有効にすると、DSIRQJOB を開始することによって JES ジョブ ID を取得できます。ユーザーが DSIRQJOB タスクを無効にすると、INGTJLM タスクが、ジョブ・ログ・モニターの開始時に、必要に応じて JES ジョブ ID を自動的に獲得します。どちらの場合も JES のシャットダウンを試行する前に、ユーザーは DSIRQJOB タスクまたは INGTJLM タスクのいずれか (採用されているもの) を停止してジョブ ID を解放する必要があります (SA の推奨方法に関するベスト・プラクティス・ポリシーを参照)。あるいは、NetView の「インストール: 概説」マニュアルのトピック『[JES を開始する前に NetView プログラムを開始する](#)』を参照してください。このトピックでは、NetView MVS コマンド改訂機能を使用して、JES が異常終了したとき、またはコマンド行からユーザーが停止したときに DSIRQJOB ジョブを停止する方法について説明しています。

モニター機能は、カスタマイズ・ダイアログの定義に基づいて SA z/OS によって制御されます。ジョブが UP、ACTIVE、または RUNNING のいずれかの状況になると、モニターが自動的に開始されます。カスタマイズ・ダイアログでジョブ・ログ・モニター定義を指定したときにすでにジョブがアクティブであった場合、このジョブに対してジョブ・ログ・モニターを開始するには、ジョブを再始動する必要があります。ただし、以下の制限付きで、SA z/OS によって制御されていないジョブをモニターすることができます。

1. 自動化に渡されるデータを制限するためのフィルター基準を指定できません。これは、通常は除外される空の行を除いて、すべてのデータ行がメッセージ自動化に転送されることを意味します。
2. メッセージ自動化に転送されるメッセージは、常に自動タスク LOGOPER に対してキューに入れられます。SA z/OS によって制御されるジョブの場合、これらのメッセージは、SA z/OS を考慮してジョブを処理する自動タスクに対してキューに入れられます。
3. ユーザーは、INGJLM START コマンドを使用してモニターを手動で開始する必要があります。必須パラメーターは、ジョブ名とモニター間隔です。デフォルトのデータ・セット JESMSG LG 以外のデータ・セットをモニターする場合は、該当する DD 名も指定する必要があります。同じジョブ名を持つジョブが複数ある場合に限り、所有者またはジョブ ID の指定が必要です。例えば、ジョブが既に終了していて、出力が出力キューに保持されている場合などです。この場合、ユーザーはジョブが 1 回だけモニターされるようにする必要があります。
4. DD 名が複数回指定されているときに、マルチステップ・ジョブの最終ステップのスプール・ファイルをモニターしない場合は、DD 名をステップ名で修飾する必要があります。ステップ名がストリーム内プロシージャで固有ではない場合は、DD 名をプロシージャ・ステップ名で修飾することも必要です。

特定のジョブのモニター機能を INGTJLM STOP コマンドを使用していつでも停止することができます。これは SA z/OS によって制御されるジョブの場合も同様です。通常、モニターはジョブが終了したときに自動的に停止します。SA z/OS によって制御されるジョブの場合、この停止は、ジョブが AUTOTERM や ENDING などの終了状況に達したときに行われます。SA z/OS によって制御されないジョブの場合は、ジョブが終了し、モニター間隔が 2 回満了したときにモニターが自動的に停止します。

モニター・タスクは、INGJLM SUSPEND コマンドを使用して無期限に中断することができます。すべてのモニター対象ジョブの累積された出力は、タスクが再開されるまで処理されません。その間に終了したジョブの出力は、出力が JES 出力キューにまだ保持されている場合を除いて失われます。タスクが中断された後に開始されたジョブは、タスクが再開された後はモニターされません。タスクを再開するには、NetView コマンド START TASK=INGTJLM を使用します。

モニター・タスクには、NetView のリサイクル後にモニターを続行するよう指示を出すことができます。これは、タスクがアクティブであるときにコマンド `INGJLM RECYCLE RESUME` を発行することによって行います。このコマンドを発行する必要があるのは、次の IPL まで、以降のすべての NetView リサイクルについて 1 回のみです。この動作はタスクを中断して再開したときに似ていますが、タスクが中断されない点、およびタスクの終了と再開の間に NetView リサイクルが行われる点が異なります。コマンド `INGJLM RECYCLE RESET` を発行することによって、状態をリセットできます。これにより、NetView の再始動後に内部構造がクリアされます。これがデフォルトです。内部構造は、マスター・アドレス・スペースに固定されているデータ・スペース内に保持されます。この構造の初期サイズは、モニター用の PDB で定義されているジョブ数によって異なります。最小で 16K になります。モニター用に現行割り振りだけでなくスペースが必要になるたびに、データ・スペースは、インストール先制限まで 1 ブロックずつ拡張されます。ただし、DD 名のモニターに必要なスペースは、モニター対象 DD 名のジョブの終了後に再利用されます。

データ・スペースの名前は、DSIPARM メンバー `INGXINIT` で定義されている `GRPID` パラメーターの値が接尾部として付けられた定数 `INGJLM` によって与えられます。この値を変更すると、System Automation が次回に初期化されたときに、新しいデータ・スペースが作成されます。

拡張自動化オプションを使用して、動作を自動的に設定できます。System Automation は、(再)初期化されるたびに、ジョブ・ログ・モニター・タスクの状況を検査します。タスクがアクティブである場合、または開始を必要としている場合は、共通グローバル `AOF_AAO_JLM_RECYCLE` の評価も行い、グローバルの値に従ってコマンド `INGJLM RECYCLE` を発行します(詳しくは、[204 ページの『読み取り/書き込み変数』](#)を参照してください)。

異常終了の状態では、次の場合を除き、モニター・タスクによって内部中断コマンドが実行されます。エラー状態の原因となったジョブは「エラー状態」とマークされ、タスクが再開したときにモニター対象から除外されます。

注: モニター・タスクは、JES をシャットダウンする前に終了する必要があります。このため、デフォルト・ポリシーが更新され、次の `STOP` コマンドが JES の `SHUTINIT` フェーズに追加されています。

```
PIPE NETV INGJLM STATUS
| LOC /INGTJLM: ACTIVE/
| EDIT /STOP TASK=INGTJLM/ 1
| NETV
```

## 制限

1. 実行時間が 2 秒より短い SA z/OS 制御のジョブの実行はモニターされない可能性があります。1 つの原因としては、自動化ではこれ以上アクティブなジョブは検出されないためです。あるいは、モニター・タスクがリソースを割り振る前に JES がそのリソースの割り振りを解除しているためです。後者は、ジョブが終了した後に出力が JES 出力キューに保持されていない場合の SA z/OS によって制御されないジョブにも当てはまります。いずれの場合も、出力を JES 出力キューに保持するジョブの開始時にメッセージ・クラスを指定して、モニターを手動でトリガーします。
2. モニター機能は、1 次サブシステム JES2 および JES3 に限定されます。
3. 動的割り振りスプール出力データ・セットはサポートされません。
4. NetView タスク `DSIRQJOB` が定義されているときは、ジョブ・ログ・モニターは `DSIRQJOB` が JES ジョブ ID を受け取るのを待ちます。ジョブ ID がないと、スプール・データ・セットにアクセスできません。`DSIRQJOB` が定義されていない場合、ジョブ ID はジョブ・ログ・モニターが取得してモニター・タスクの終了時に JES に返します。

`DSIRQJOB` が JES2 環境で終了すると、ジョブ ID が JES2 に返されるためモニター・タスクは自動的に中断されます。`DSIRQJOB` の再開後、モニター・タスクは自動的に再開されます。

JES3 環境では、`DSIRQJOB` は終了時にジョブ ID を返しません。このため、モニター・タスクは終了しません。しかし、ジョブ ID が何らかの理由で JES3 に返されると、モニター・タスクは自動的に中断します。ただし、新しいジョブ ID の受け取り時にモニター・タスクを手動で再開する必要があります。

## カスタマイズ

SA z/OS によって制御されるジョブの場合は、カスタマイズ・ダイアログで各ジョブに対してモニター間隔、メッセージ、およびモニター対象のデータ・セットを定義する必要があります。ジョブのモニターを

実際に開始する主要な定義はモニター間隔の定義です。この間隔は「APPLICATION INFO」ポリシーで指定します。

注：間隔を定義しないと、「MESSAGES/USER DATA」ポリシーで指定される内容に関係なくモニターは実行されません。

```

                                Application Information                                Line 00000001
Command ==> ----- Scroll ==> CSR__
Entry Type : Application                PolicyDB Name : SAMPLE_PDB
Entry Name : JLMS05                    Enterprise Name : KEYPLEX
:
JCL Procedure Name . . . . AAZJLMS
Job Log Monitor Interval . 00:17      (mm:ss NONE)
Captured Messages Limit . ___         (0 to 999)
:

```

図 7. アプリケーションのジョブ・ログ・モニターを定義する ISPF ダイアログ

自動化する専用メッセージを定義するには、メッセージ ID のオフセットを 1 行以内に指定する必要があります。オプションでメッセージを固有にする 1 つまたは複数のトークンを指定します。オフセット値 0 は特定の ID を持たないメッセージを示します。この場合は、少なくとも 1 つのトークン・ペアを定義してメッセージを識別する必要があります。すべてのメッセージが関連し、自動化される必要がある場合に備えて、共通のメッセージ ID 値 JOBLOGALL が指定されます。この ID には、これ以上のユーザー・データの指定は必要ありません。ただし、メッセージ ID に「S」、「C」、「R」、「K」、または「U」の定義が付いている場合を除いて、このダイアログでは ACF フラグメントは生成されません。このため、ユーザー・データのキーワードと値のペア JLM\_OFFSET=NO が必要になります。

```

                                Message Processing                                Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
Entry Type : Application                PolicyDB Name : SAMPLE_PDB
Entry Name : JLMS05                    Enterprise Name : KEYPLEX
:
Cmd Message ID      Description                Cmd Rep Cod Usr A M
u__ JOBLOGALL      Automate all JESMSGLG messages_____  * *
:

```

図 8. すべての JESMSGLG メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ (1/3)

```

                                User Data Processing : JOBLOGALL                Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
:
Cmd Keyword      Data
__ JLM_OFFSET    NO_____
:

```

図 9. すべての JESMSGLG メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ (2/3)

特定のメッセージをモニターする場合は、キーワード JLM\_OFFSET で、対応する SYSOUT データ・セット内のメッセージ ID の実際のオフセットを定義する必要があります。JESMSGLG データ・セットの場合、すべてのテキストのオフセットの開始位置は特定の桁になります。

#### JES2 環境

テキストの開始位置は桁 20 です。複数行メッセージであっても、メッセージ・テキストが折り返されていても関係あり複数行メッセージであるか、メッセージ・テキストが 126 文字 (19 バイトの接頭部を含む) よりも長いために折り返されているかにかかわらず、テキストの開始位置は桁 20 です。複数行メッセージの後続行は、桁 4 に複数行メッセージのコンソール ID が表示されます。折り返されたメッセージ・テキストは、最初の 19 桁にブランクが表示されます。このため、このようなメッセージのトークンのトラップでは、各トークンの正確な位置を知っている必要があります。各メッセージの先頭行の最初の 2 つのトークンは時刻とジョブ ID です。

#### JES3 環境

テキストの開始位置は桁 12 です。複数行メッセージであっても、メッセージ・テキストが折り返されていても関係あり複数行メッセージであるか、メッセージ・テキストが 126 文字 (11 バイトの接頭部を



含む) よりも長いために折り返されているかにかかわらず、テキストの開始位置は桁 12 です。各メッセージの最初のトークンは常に時刻です。

その他の SYSOUT データ・セットのレイアウトは異なります。

以下のサンプルにある最初のトークンの定義は「\*」文字で終わっています。これはメッセージ・トークンについて、星印までのすべての文字が検査されることを示します。星印と直後のすべての非空白文字は無視されます。示されているメッセージが自動化に渡されるのは、その ID が桁 21 にあり、ID の後の次のトークンの値が「3」であり、7 番目のトークンの先頭が KEY= である場合です。

```

User Data Processing : cccnnnna          Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
:
Cmd Keyword          Data
--- JLM_OFFSET ----- 21
--- JLM_TOKEN ----- (7, 'KEY=*)
--- JLM_TOKEN ----- (4, '3')
:

```

図 10. 特定のメッセージの自動化を定義する ISPF ダイアログ

JESMSGLG データ・セット以外のデータ・セットをモニターする場合は、メッセージ定義を修飾する必要があります。次の例では、AAAZOUT と SYSPRINT の DD 名で参照される SYSOUT データ・セットがすべてのメッセージを自動化に渡します。

```

User Data Processing : JOBLGALL          Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
:
Cmd Keyword          Data
--- JLM_OFFSET ----- NO
--- JLM_DDNAME ----- SYSPRINT
--- JLM_DDNAME ----- AAAZOUT
:

```

図 11. 特定のメッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ

次の例は、マルチステップ・ジョブのモニター定義を示しています。DD 名 SYSPRINT に書き込まれた、ジョブ・ステップ STEP1 および STEP3 のすべてのメッセージが自動化に渡されます。ただし、SYSPRINT に書き込まれた STEP2 のメッセージは無視されます。

```

User Data Processing : JOBLGALL          Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
:
Cmd Keyword          Data
--- JLM_OFFSET ----- NO
--- JLM_DDNAME ----- STEP1.SYSPRINT
--- JLM_DDNAME ----- STEP3.SYSPRINT
:

```

図 12. マルチステップ・ジョブの特定メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ

次の例は、メッセージに関連付けられていないフィルター基準を定義する方法を示しています。「ダミー」メッセージ ID をアンカーとして定義することは必要です。最初のトークンの値が「Name」である、DD 名 AAAZOUT で定義された SYSOUT データ・セットのすべての行が自動化に渡されます。

```

Message Processing          Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
:
Entry Type : Application          PolicyDB Name : SAMPLE_PDB
Entry Name : JLMS05              Enterprise Name : KEYPLEX
:
Cmd Message ID      Description          Cmd Rep Cod Usr A M
u__ DUMMY1          Automate specific text line_____ * *:
```

図 13. 非メッセージ印刷行のジョブ・ログ・モニターを定義する ISPF ダイアログ (1/2)



```

User Data Processing : DUMMY1                Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
:
Cmd Keyword      Data
--- JLM_OFFSET----- 0
--- JLM_DDNAME---- AAZOUT
--- JLM_TOKEN----- (1, 'Name')
:

```

図 14. 非メッセージ印刷行のジョブ・ログ・モニターを定義する ISPF ダイアログ (2/2)

ビルド・プロセスによって、メッセージ INGY1300I を自動化するために次の一般的な MAT エントリーが自動的に生成されます。ただし、SA z/OS によって制御されるジョブの場合に限ります。

```

* INGY1300I for jobs defining a message ID
IF MSGID = 'INGY1300I' & TOKEN(6) ^= 'N/A'
& TOKEN(3) = SVJOB & TOKEN(6) = MSGTYPE THEN
  DISPLAY(N) NETLOG(Y) SYSLOG(N)
  EXEC(CMD('ISSUEACT JOBNAME=SVJOB' MSGTYPE='MSGTYPE')
  ROUTE(ONE %AOFOPGSSOPER%)
);

```

図 15. メッセージ INGY1300I およびモニターに対してメッセージ ID を定義するジョブの一般的な MAT エントリー

これは単にメッセージ ID の代わりに実行されるコマンドの定義を必要とします。例えば、次のとおりです。

```

Message Processing                Line 00000001 Col 001 075
Command ==> ----- Scroll ==> CSR__
:
Entry Type : Application          PolicyDB Name   : SAMPLE_PDB
Entry Name : JLMS05              Enterprise Name : KEYPLEX
:
Cmd Message ID  Description      Cmd Rep Cod Usr  A M
c__ JOBLOGALL   Automate all JESMSGLG messages  1 * *
:

```

図 16. すべての JESMSGLG メッセージのジョブ・ログ・モニターを定義する ISPF ダイアログ (3/3)

メッセージ INGY1300I は複数行メッセージであることに注意してください。最初の行はラベル行です。この行は OWNER、JOBNAME、JOBID、DDNAME、MSGID の各トークンをこの順序で示します。最後のトークンは、「Message Processing」ダイアログで指定したメッセージ ID です。後続の行はそれぞれ、オリジナル・メッセージの行を表します。

ポリシーに定義されていないジョブ、またはポリシーに定義されているがメッセージ ID がモニター用に定義されていないジョブをモニターする場合は、既存の事前定義 AT 項目を以下のように手動で調節 (AT オーバーライドを作成) する必要があります。

```

* INGY1300I for jobs having no message ID defined
IF MSGID = 'INGY1300I' & TOKEN(6) = 'N/A' THEN
  DISPLAY(N) NETLOG(Y) SYSLOG(N)
  EXEC(CMD('xxxxxxx ...'))
  ROUTE(ONE %AOFOPGSSOPER%)
);

```

図 17. メッセージ INGY1300I およびモニターに対してメッセージ ID を定義していないジョブの一般的な MAT エントリー

いずれの場合も、手動でモニターを開始する必要があります。また、フィルター基準が定義されていないため、すべてのメッセージが自動化に渡されます。

注: メッセージを取得するために NetView PIPE ステージ SAFE を使用すると、メッセージがデフォルト SAFE に示されます。

## SPIN パラメーター

スピンオフされ、ジョブ・ログ・モニターでモニターされるように定義されている、以下のスプール・データ・セットの場合、

```
JESMSG LG
JESYSMSG
SYSOUT DD 名 (JES2 のみ)
```

ジョブ・ログ・モニターがデータ・セットを処理するまで、出力を保持しておくことを強くお勧めします。そうでないと、自動化したい情報を失うおそれがあります。

JESMSG LG データ・セットの場合は、出力が抑止されていないことを確認する必要があります。JES2 環境では、コマンド

```
$D JOBCLASS(STC),LOG
```

がパラメーター LOG について値 YES を表示しなければなりません。JES3 環境では、コマンド

```
*I,STD
```

はパラメーター JESMSG について値 NOSTC を表示しなければなりません。また、コマンド

```
*I,C=x
```

がパラメーター JESMSG について値 NOLOG を表示してはなりません。「x」は開始タスクに割り当てられているクラスです。

ジョブ・ログ・モニターは、特定のデータ・セットが JES によってスピンオフされたことを検出すると、データ・セットの処理を終了後にメッセージ INGY1333I を出します。このメッセージは、特定のスプール・データが解放可能になったことを通知するものです。例えば次のように、ジョブ名またはジョブ ID とともにメッセージの SYSOUT ID を使用してデータ・セットをページできます。

```
JES2: $P 0 JQ(jnm|jid),OUTGRP=soid
JES3: *F,U,J=jnm|jid,DSN=...soid,CANCEL
```

「jid」はジョブ ID の数値部分のみであることを注意してください。

SYSOUT ID がメッセージにない場合、タスクは値を判別できません。

メッセージ INGY1306I は、特定の DD 名に関してモニターが停止したこと、およびすべての対応するスプール・データ・セットを解放可能であることを示します。

## 状況情報

NetView PIPE でも実行できるコマンド INGJLM STATUS は、モニター・タスクの現在の状況とそのモニター対象データ・セットを戻します。

```
Status of task INGJLM: {ACTIVE|INACTIVE|SUSPENDED}
Monitoring on recycle : {RESUME|RESET|N/A}
Owner   Jobname Jobid   DDname  Status   Freq. Last           Read  Passed
        Procstep Step
-----
xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx mm:ss hh:mm:ss      n      n
                                     n      n
xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx mm:ss hh:mm:ss      n      n
        xxxxxxxx xxxxxxxx                                     n      n
xxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx mm:ss hh:mm:ss      n      n
        xxxxxxxx                                     n      n
*** Status complete ***
```

図 18. ジョブ・ログ・モニターの状況情報

データ・セットの状況には、以下の情報が表示されます。

**E.....**

データ・セットは「エラー状態」とマークされ、それ以降はモニターされません。

**.A.....**

データ・セットは割り振られています。

**..O.....**

データ・セットは正常に開きました。

**...U.....**

ユーザーにデータ・セットの読み取り許可がありません。

**.....S.....**

データ・セットのモニターは ACF ベースです。

**.....M....**

すべてのメッセージが自動化に渡されます。

**.....J..**

対応するジョブがアクティブ・アドレス・スペースのチェーンにありません。

**.....I.**

スプール・データ・セットが初期化されていません。次のモニター間隔でデータ・セットの割り振りが再試行されます。これはデータ・セットが初期化されるまで繰り返されます。

**.....N**

ddname で SPIN 属性が定義されています。これは、対応するスプール・データ・セットが処理中にスピノフ可能であることを意味します。

頻度 (Freq.) には、2 回の累積メッセージの検査の間隔 (分と秒) が表示されます。次の列には、最後の検査が行われた時刻が表示されます。最後の 2 つの数値には、読み取られたメッセージの合計数と自動化テーブルに渡されたメッセージの数が表示されます。各データ・セットの 2 番目の状況表示行には、最後の検査以降に読み取られて、自動化に渡されたメッセージの数が表示されます。



## 第5章 アラートに基づく通知

SA z/OS では、主題のエキスパートにアラートを出すことができる、アラートに基づく通知サービスが用意されています。アラート、イベント、またはトラブル・チケットを各種の通知ターゲットに送信することで、手操作による介入を必要とする自動化の問題をエスカレートすることができます。

SA z/OS では、通知ターゲットへのアラートの送信を可能にする以下のような複数の通信方式をサポートします。

- System Automation for Integrated Operations Management (SA IOM)
- Tivoli Enterprise Console (TEC)
- Tivoli® NETCOOL/OMNIbus
- IBM Tivoli Service Request Manager®
- ユーザー定義のアラート・ハンドラー

### 概要

SA z/OS のアラートに基づく通知サービスを使用して、事前定義済み状態に対するアラートをオペレーターまたはシステム・プログラマーに送信できます。

また、必要に応じて、カスタマイズ・ダイアログおよび **INGALERT** ユーティリティを使用して、アラートの発行時点をカスタマイズすることもできます。アプリケーション (APL)、モニター・リソース (MTR)、アプリケーション・グループ (APG)、および MVS コンポーネントに関してのみアラートを発行することができます。

アラートは、SA z/OS 自動化エージェントが収集して、通知処理のためにターゲットに送信する情報のセットです。送信される情報はテキストから成り、アラートを受ける個人またはグループに転送されます。この情報は追加オプションで補足され、それにより、異なる種類の通知ターゲットでの処理が詳細に決定されます。

SA z/OS には、コマンドが問題状態 (リソースの劣化、あるいは所定の時間間隔内に up 状態にならないなど) を検出した場合にアラートをトリガーする、いくつかの事前定義されたアラート・ポイントがあります。

アラートは、さまざまなレベルで使用可能または使用不可に設定できます。

- グローバルに **INGCNTL** コマンドを使用する
- リソースの通知リストをリソース固有に使用する
- メッセージ ID **INGALERT** 用のコード定義をアラート固有の方法で使用する

### 通信のフロー

56 ページの図 19 は、自動化マネージャーと自動化エージェント間の通信の概要を示しています。

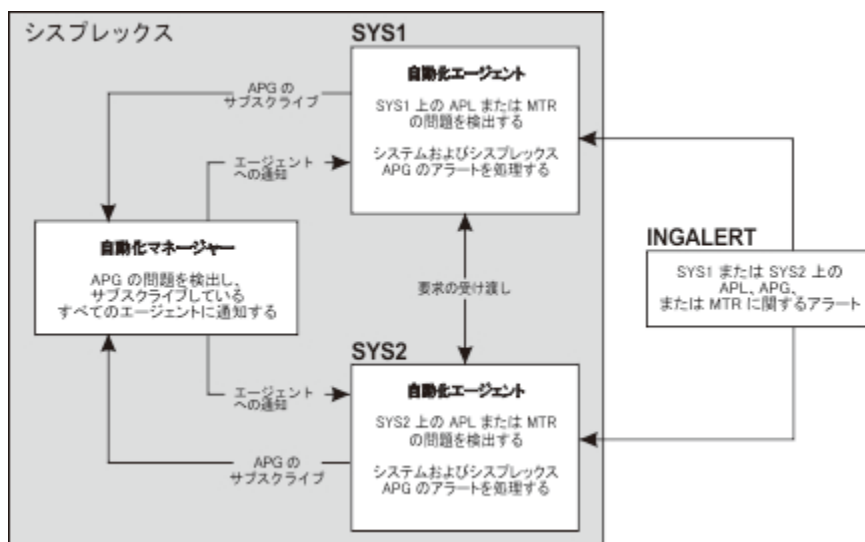


図 19. アラート通信のフロー

シスプレックス内のシステム上の自動化エージェントは、システムまたはシスプレックス・アプリケーション・グループ (APG) の問題に関連するアラートを受け取るために自動化マネージャーにサブスクライブします。これは、自動化マネージャー自体は、通知ターゲットにアラートを送信できないからです。自動化マネージャーは、APG での問題を検出した場合は常に、サブスクライブしている自動化エージェント (システム APG の場合は 1 つの自動化エージェント、シスプレックス APG の場合はすべての自動化エージェント) にアラートを送信します。シスプレックス APG に対するすべてのアラートは、シスプレックス内の 1 つの自動化エージェントのみによって処理されます。

自動化エージェントは、アプリケーション、アプリケーション・グループ、またはモニター・リソースに関連するアラートを INGALEERT コマンドを介して受信することもできます。影響を受けるリソースが別の自動化エージェントで管理されている場合、その要求は受け渡されます。リソースを管理する自動化エージェントは、そのアラートを通知ターゲットに送信します。何らかの理由でアラートを送信できない場合、この自動化エージェントは、要求をシスプレックス内の次の自動化エージェントに受け渡します。アラートが正常に送信されるまで、あるいは、その他には使用可能な自動化エージェントがなくなるまで、この受け渡しが数回にわたって発生することがあります。

それぞれのアラートごとに、自動化エージェントは通知ターゲットに接続し、アラートを送信した後に切断します。自動化エージェントは、SA IOM サーバーへの永続的な接続は維持しません。

## アラートの使用可能化

デフォルトでは、アラートは使用可能ではありません。アラートを活動化するには、SA z/OS と通知ターゲットの両方でセットアップ・アクションを実行する必要があります。

### SA z/OS でのセットアップ

SA z/OS の 3 つの異なるレベルで、アラートをオンまたはオフにすることができます。

- システム・レベル (INGCNTL コマンドを介して)。アラートをオフにするということは、システムが検出する、または受け入れるアラートはないことを意味します。アラートは、少なくとも 1 つの通知ターゲットについてグローバルに、または選択的に、明示的にオンにする必要があります。
- リソース・レベル (「通知リスト (Inform List)」ポリシー・フィールドを介して)。アラートをオフにするということは、そのリソースに関して検出される、または受け入れられるアラートはないことを意味します。通知ターゲットについてアラートを活動化するには、通知ターゲットを明示的に指定 (または、デフォルトから継承) する必要があります。
- アラート ID レベル (リソースまたは MVS コンポーネント項目の INGALEERT メッセージ ID 用のコードを介して)。

## INGCNTL コマンド

デフォルトでは、アラートは使用可能ではありません。INGCNTL コマンドを実行して、アラートを使用可能に設定し、接続プロパティを通知ターゲット用に設定する必要があります。

これを実行するには、次のようにします。

- NetView スタイル・シートで以下のように補助コマンドを使用します。

```
*****
* Auxiliary commands
*****
* Enable Alerting and set connection properties
auxInitCmd.A = INGCNTL SET ALERTMODE=IOM ALERTHOST=saiom:1040:SAALERT
```

- SA z/OS で提供されている AOFEXDEF 出口から次のように入力します。

```
'INGCNTL SET ALERTMODE=IOM ALERTHOST=saiom:1040:SAALERT'
```

INGCNTL コマンドについて詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

## 通知リスト

57 ページの表 7 に示すように、通知ターゲットに対する適切な通信方式を、適切なポリシー項目の「通知リスト (Inform List)」フィールドに指定して、特定のリソースまたはリソースのクラスに関するアラートを明示的に使用可能にする必要があります。

ポリシー・オブジェクト	ポリシー項目
アプリケーション・グループ (APG)	APPLGROUP INFO *
アプリケーション (APL)	APPLICATION INFO *
モニター・リソース (MTR)	MONITOR INFO *
MVSCOMP デフォルト (MDF)	MVSESA INFO *
システム・デフォルト (SDF)	AUTOMATION OPTIONS
シスプレックス・デフォルト (XDF)	RESOURCE INFO
疑似メッセージ CAPMSG (APG、APL、MTR、MVC)	メッセージ/ユーザー・データ

\* 「通知リスト (Inform List)」フィールドをブランクのままにすると、ポリシー・オブジェクトは、システム・デフォルト定義またはシスプレックス・デフォルト定義に指定された値を継承することができます。

57 ページの表 8 に示すように、希望する通知ターゲットに対して適切な通信方式を指定する必要もあります。

「通知リスト (Inform List)」の値	通信方式	SA z/OS でサポートされる通知ターゲット
IOM	IBM Tivoli System Automation for Integrated Operations Management (SA IOM) の対等プロトコル	SA IOM
EIF	Tivoli Event Integration Facility (EIF)	TEC または OMNIbus
TTT	XML	TDI 経由の TSRM

表 8. 通知リストの通信方式 (続き)		
「通知リスト (Inform List)」の値	通信方式	SA z/OS でサポートされる通知ターゲット
USR	コマンド呼び出し	ユーザー定義のアラート・ハンドラー

ブランクで分離された値のリストを指定して、複数の通知ターゲットへのアラート送信を可能にできます。

### 「コード処理 (Code Processing)」

INGALERT メッセージ ID を使用したコード処理により、通知ターゲットに渡されるイベントに追加の特性を定義することができ、また、特定のアラートに対してイベントを作成しないようにすることができます。このような定義は、アラート ID に依存し、通知ターゲットのジョブおよびタイプを発行して行うことができます。

メッセージ ID INGALERT のコード定義は、タイプが APL、APG、MTR のリソース、および MVS コンポーネントに対して使用できます。APL、APG または MTR リソースで一致するものが見つからなかった場合、INGALERT コード定義は、リソースが常駐するシステム上で対応する MVS コンポーネント項目に対して検査されます。

INGALERT メッセージ ID のコード処理パネルで、以下のように入力します。

#### コード 1

アラート・タイプを示すアラート ID です。SA z/OS は、以下の組み込みアラート・ポイントのセットを提供します。

アラート ID	説明	対象のリソース・タイプ
CMD_FAILED	戻りコード検査がオンであり、コマンドが RC≠0 で終了しました	APL、MTR
COMM_FAILED	他のシステムとの通信中にエラーが検出されました	APL
CRITICAL_WTOR	CRITICAL と定義された WTOR (REPLY の定義なし)。	APL
ISQHWMSG	重要ハードウェア・メッセージをターゲット・ハードウェアで確認できます。	MVSESA
ISQHWST	ターゲット・ハードウェアが DEGRADED 状況または SERVICE REQUIRED 状況です。	MVSESA
MSG/Message ID	重大度がクリティカルなメッセージ。このメッセージはワイルドカードを使用して短縮可能です (例: MSG/DFS54*)。  このアラート ID は、AOFCPMSG コマンドによって取り込まれたメッセージにのみ適用されます。	APL、APG、MTR
OS_DEGRADED	リソースの監視状況が低下しています	APL
OS_PROBLEM	自動化状況が ZOMBIE または BROKEN であるか、SA z/OS 外部でのシャットダウン、および再始動が許可されていません	APL
REC_FAILED	マイナー・リソースのクリティカルしきい値を超えたため、自動化が一時停止されました	APL
REP_FAILED	SA z/OS によって保管されている未処理の WTOR に対して、これ以上の応答は不要です	APL
START_FAILED	開始コマンドが失敗しました	APL



アラート ID	説明	対象のリソース・タイプ
START_PENDING	タイムアウト・インターバル内で UP メッセージが受信されませんでした	APL
STOP_PENDING	停止コマンドを使い尽くしました	APL
CS_PROBLEM	複合状況 PROBLEM が設定されています	APG
OPCERR	メッセージ EVJ120I で報告されたとおりに TWS リソースへの状況変更を示します	APL

注：アラート・ポイント ISQHWMSG および ISQHWST は、SNMP プロトコル経由で接続されたターゲット・ハードウェアのプロセッサ操作によって提供されます。

また、ユーザー定義の任意のアラート ID を使用することもできます。対応するコード項目にそのアラート ID を指定し、この ID を使用して INGALERT を呼び出します。ワイルドカードがサポートされています。

### コード 2

APL の場合、これは、アラート実行対象のジョブ名です。MVC の場合は、MVSESA が含まれます。APG および MTR の場合、コード 2 は無視されます。ワイルドカードがサポートされています。これにより、APL クラスを使用して、複数の APL に対するアラートを同時に設定することができます。

### コード 3

アラートを通知ターゲットに送信するために使用する通信方式です。有効な値は IOM、EIF、TTT、または USR です。ワイルドカードがサポートされています。

### 返される値

この値は、イベントの作成を行わない場合は IGNORE か、または、渡されるイベントと一緒に通知ターゲットに送信するパラメーターである場合があります。これらのパラメーターの意味は、通信方式のタイプによって以下ようになります。

#### IOM

「戻された値 (Value Returned)」の最初の 2 つのトークンは、以下とみなされます。

- アラートの優先順位 (0 から 999)。
- アラートの処理方法を決定するルールを定義するために SA IOM で使用されるエスカレーション ID。この値の長さは 20 文字に制限されます。

最初の 2 つのトークンに無効値が入っている場合、「戻された値 (Value Returned)」は IGNORE とみなされます。

「戻された値 (Value Returned)」フィールドに 2 つを超えるトークンを指定すると、余分なトークンは無視されます。

#### EIF

「戻された値 (Value Returned)」は、イベントの重大度とみなされます。有効な値は HARMLESS、WARNING、MINOR、CRITICAL、または FATAL か、それに対応する 1 から 5 までの数値です (1 は HARMLESS に対応します)。重大度を指定するいずれの代替方法も、TEC または NETCOOL/OMNIBus へのイベントに対して使用できます。重大度を数値で指定する場合、コード定義を使用してアラートを SA IOM に送信することもできます。

有効な重大度を指定しないと、「戻された値 (Value Returned)」は IGNORE とみなされます。

「戻された値 (Value Returned)」フィールド内の余分なトークンは無視されます。

#### TTT

TSRM が通知ターゲットの場合、「戻された値 (Value Returned)」は以下のように使用されます。

- 最初に報告されたときのトラブル・チケットの優先順位 (1 から 5)
- トラブル・チケットをどの程度早急に解決する必要があるかを示す緊急度 (1 から 5)
- トラブル・チケットが業務に及ぼす影響または重大度 (1 から 5)

ほかのターゲットでは、これ以外の値が期待される場合があるため、これらの値は検証されません。

「戻された値 (Value Returned)」フィールドに 3 つを超えるトークンを指定すると、余分なトークンは無視されます。3 つより少ないトークンを指定すると、それらのトークンはその位置に従って使用され、欠落したトークンはデフォルトで N/A になります。

#### USR

「戻された値 (Value Returned)」フィールドの内容は、呼び出されるユーザー定義のアラート・ハンドラーに渡されます。

#### コード定義例

60 ページの表 9 の例を検討します。

コード 1	コード 2	コード 3	返される値
START_FAILED	IMS*	IOM	500 IMS_start
START_FAILED	DB2*	EIF	CRITICAL
*	*	*	IGNORE

この例のコード定義は、以下のような動作を行うこととなります。

- 名前の接頭部が IMS のジョブに対して、アラート ID START\_FAILED を持つアラートが、優先順位 500 およびエスカレーション ID IMS\_start で、IOM へ送信されます。
- 名前の接頭部が DB2 のジョブに対して、アラート ID START\_FAILED を持つアラートが、イベントの重大度 CRITICAL で、TEC または NETCOOL/OMNibus へ送信されます。
- すべての通知ターゲットに対するほかのすべてのアラートは無視されます。

## INGALERT コマンド

INGALERT コマンドを使用して、アラートをシステムに注入することができます。これは、NetView 自動化テーブル、自動化プロシージャー、またはコマンド行のいずれかから行うことができます。

以下のパラメーターを指定できます。

- リソース名、テキスト MVSESA、あるいはジョブまたはサブシステム名。
- アラート ID (例えば CS\_PROBLEM、CMD\_FAILED など)。
- 通知ターゲットに渡されるメッセージ・テキストまたはテキスト・ストリングを識別するメッセージ ID。

例えば、以下をコマンド行または自動化プロシージャーから使用できます。

```
INGALERT MYGRP/APG/SYS1 ID=MYALERT TEXT=(MYGRP HAS A PROBLEM)
```

この例では、INGALERT はアラート ID MYALERT を使用して、メッセージ ID INGALERT の一致するコード定義を介して追加パラメーターを取得し、アラート・テキストとして TEXT パラメーター値を使用します。

次の例は、メッセージ ABC123I が発行されると必ずアラートが送信されるようにする場合に、NetView 自動化テーブルから使用できます。

```
IF MSGID='ABC123I'
THEN
EXEC(CMD('INGALERT'));
```

INGALERT は ABC123I をアラート ID として使用し、アラート・テキストとして ABC123I の完全なメッセージ・テキストを使用します。INGALERT のリソース・パラメーターは、デフォルトで、そのメッセージを発行したサブシステムのジョブ名になります。

INGALERT ユーティリティについて詳しくは、「IBM Z System Automation プログラマーズ・リファレンス」を参照してください。

## 第 6 章 可用性およびリカバリー時間の報告

SA z/OS には、利用者への請求、あるいはユーザーの重大なアプリケーションまたはそのアプリケーションが依存するソフトウェアの信頼性の報告に役立つサポートが導入されました。例えば、アプリケーションの実行所要時間に基づいて正確に課金したい場合があります。このことは、非 MVS リソース (USS アプリケーションなど)、または NetView アドレス・スペースで実行される可能性があるモニター・リソースの場合に重要です。

### 概要

SA z/OS は、ジョブ関連の情報を収集して記録し、システム管理機能 (SMF) レコードをリソース存続期間中の特定イベントに書き込みます。

このリソースには、以下のものがあります。

- サブシステム (APL)
- ローカル・システムがホスティングするアプリケーション・グループ (APG) およびシスプレックス・アプリケーション・グループ
- モニター・リソース (MTR)

INGPUSMF バッチ・ユーティリティーは、スプレッドシートにインポートできるレポート・ファイルを作成します。IBM Tivoli System Automation Application Manager で提供および利用される DB2 テーブルに、レポートを変換し、書き込むこともできます。詳しくは、66 ページの『DB2 への SMF レポートの書き込み』を参照してください。

リソースの情報ポリシー項目の「Inform List」フィールドに値 SMF を入力することにより、リソースに関してレコードを書き込むかどうかを制御できます。

### リソースのライフ・サイクル

61 ページの図 20 は、SA z/OS が情報を記録するときの、アプリケーションの存続期間内のイベントを示します。

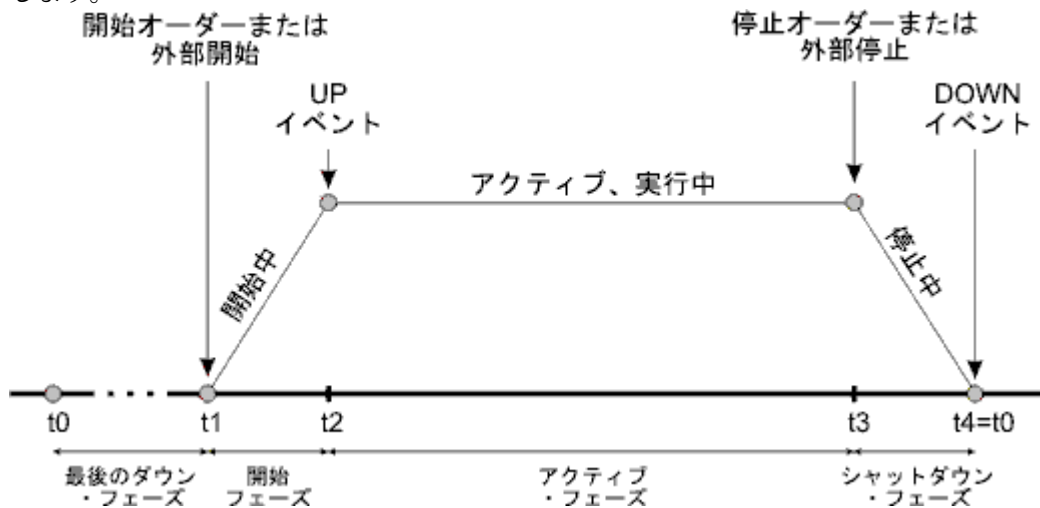


図 20. アプリケーションのライフ・サイクル内のイベント

イベントは以下のとおりです。

- 自動化マネージャーからの開始オーダーの受信 ( $t_1$ )
- UP シグナルの受信 ( $t_2$ )
- 自動化マネージャーからの停止オーダーの受信 ( $t_3$ )
- DOWN シグナルの受信 ( $t_4=t_0$ )

これらのレコードを調べることにより、特定の時間枠に関して以下の情報を確立することができます。

- アプリケーションの実行可能時間とダウン時間
- アプリケーションの始動時間とシャットダウン時間
- スケジュールされた停止の回数とスケジュールされたダウン時間の概数
- スケジュールされていない停止の回数とスケジュールされていないダウン時間の概数

SA z/OS SMF レコードの使用を簡単にするために、自動的に、以下の期間が計算されて SMF レコードに保管されます (秒単位)。

- 起動時間 ( $t_2-t_1$ )
- シャットダウン時間 ( $t_4-t_3$ )
- アプリケーションがアクティブであった時間 ( $t_3-t_2$ )
- 最後のダウン時間 ( $t_1-t_4$ )

これにより、アプリケーションのライフ・サイクルを的確に把握することができます。

## SMF レコードのレイアウト

62 ページの表 10 は、SMF レコードに保管されるデータの詳細を示しています。

オフセット	長さ	形式	説明
00	2	2 進数	レコード長。このフィールドとその次のフィールド (合わせて 4 バイト) がレコード記述ワード (RDW) になる。
02	2	2 進数	セグメント記述子。これはゼロです。
04	1	2 進数	システム・インディケータ・ビット: <b>0</b> 予約済み <b>1</b> 使用するサブタイプ
05	1	2 進数	SMF レコード・タイプ。これは 114 です。
06	4	2 進数	レコードが SMF バッファーに移された午前 0 時からの経過時間 (1/100 秒単位)。
10	4	パック	レコードが SMF バッファーに移された日付を 0ccyydddF 形式で示す。
14	4	EBCDIC	システム 識別 (SID パラメーターから)。
18	2	2 進数	レコード・サブタイプ: <b>1</b> 自動化トラッキング・レコード
20	2	2 進数	レコード・バージョン。
22	2	—	予約済み。
24	4	2 進数	レコードの開始からプロダクト・セクションへのオフセット (レコード記述ワード (RDW) を含む)。
28	2	2 進数	プロダクト・セクションの長さ。
30	2	2 進数	プロダクト・セクションの数。これは、常に 1 です。

表 10. SMF レコードのレイアウト (続き)			
オフセット	長さ	形式	説明
32	4	2 進数	レコードの開始からリソース・セクションへのオフセット (レコード記述ワード (RDW) を含む)。
36	2	2 進数	リソース・イベント・セクションの長さ。
38	2	2 進数	リソース・イベント・セクションの数。これは、常に 1 です。
40	8	—	予約済み。
プロダクト・セクション			
00	16	EBCDIC	プロダクト名 (例えば、SA z/OS V4R2M0)。
16	8	EBCDIC	シスプレックスの名前。
24	8	EBCDIC	ドメイン ID。
32	8	EBCDIC	MVS システム名。
40	8	EBCDIC	XCF グループ名。
自動化セクション			
00	24	EBCDIC	リソース名 (自動化マネージャーの表記による)。
24	8	EBCDIC	ジョブ名 (オプション)。
32	2	2 進数	イベント・タイプ: <b>X'0001'</b> 開始中 <b>X'0002'</b> アクティブ <b>X'0003'</b> 停止中 <b>X'0004'</b> 非アクティブ <b>X'0005'</b> 悪化
34	2	—	予約済み。
36	12	EBCDIC	自動化エージェントの状況 (オプション)。
48	12	EBCDIC	開始タイプ。
60	12	EBCDIC	停止タイプ。
72	5	EBCDIC	終了タイプ (異常終了コード)。オプション。
77	3	—	予約済み。
80	4	2 進数	合計起動時間 (秒単位)。
84	4	2 進数	リソースがアクティブであった経過時間 (秒単位)。
88	4	2 進数	合計シャットダウン時間 (秒単位)。
92	4	2 進数	リソースの最後のダウン時間 (秒単位)。

## SMF レコードの使用可能化

### 手順

SMF レコードをリソースに対して使用可能にするには、以下のようになります。

1. SYS(TYPE ステートメントにタイプ 114 を追加して、SYS1.PARMLIB 内の SMFPRMxx メンバーが確実にタイプ 114 の SMF レコードを収集するようセットアップします。

```
SYS(TYPE(30,...,114))
```

2. APPLICATION INFO ポリシー項目の「通知リスト (Inform List)」で、リソースに SMF を指定します。

## INGPUSMF ユーティリティー

INGPUSMF ユーティリティーを使用して、SMF レコードを分析し、スプレッドシート・プログラムにインポート可能なデータ・セットを作成できます。このデータ・セットには、容易にインポート可能な形式で SA z/OS が作成する、タイプ 114 レコードが含まれます。デフォルトでは、フィールドはセミコロン区切り文字で区切られています。

### 出力

データ・セット内の最初のレコードは、各列を表すタイトル・レコードです。残りのレコードは、データ・レコードです。タイプ 114 SMF レコードのそれぞれに対して、1つのデータ・レコードが書き込まれます。

64 ページの表 11 は、各レコードの形式を示します。

列	説明
1	SMF システム ID
2	SMF レコードが書き込まれた日付 (YYYYMMDD 形式)
3	SMF レコードが書き込まれた時刻 (hhmmss 形式)
4	SA z/OS プロダクト名 (リリース・レベルを含む)
5	シスプレックスの名前
6	システム名
7	NetView ドメイン ID
8	XCF グループ名
9	リソース名 (自動化マネージャーの表記による)
10	ジョブ名 (存在する場合)
11	Event
12	自動化エージェント status
13	起動時間 (秒単位)
14	アクティブ時間 (秒単位)
15	シャットダウン時間 (秒単位)
16	ダウン時間 (秒単位)
17	始動タイプ
18	停止タイプ

表 11. INGPUSMF ユーティリティー・データ・セット・レコードの形式 (続き)	
列	説明
19	終了 (異常終了) コード APG の終了コードには、APG リソースの計画外の終了を意味するストリング \$UNPL を含めることができます。

## INGPUSMF ユーティリティー JCL

INGPUSMF ユーティリティーはバッチ・ジョブとして実行されます。サンプルについては、INGEUSMF を参照してください。DD ステートメントの意味は、以下のとおりです。

### STEPLIB

INGPUSMF ユーティリティーを含むロード・ライブラリー。このユーティリティーは SINGLOAD ライブラリーにあります。

### REPORT

セミコロン区切り文字で区切られた形式のスプレッドシート・インポート・データ・セットを含む出力データ・セット。レコード・サイズは 255 バイトです。

### SYSPRINT

ユーティリティーが書き込む情報を含みます。

### HSATRACE

デバッグ目的のみに使用されます。これがある場合、INGPUSMF ユーティリティーは、プロセス・フローを記録するためにトレース項目を書き込みます。

### SMFDATA

SMF レコードを含みます。レコード・フォーマットは、可変、ブロック、スパンです。

### USRPARMS

フィルター基準または特定の分離文字などのユーザー・オプションを含みます。

### ユーザー・オプション

ユーティリティーの処理を制御する USRPARMS データ・セットに、さまざまなオプションを指定することができます。各オプションは、別個のレコードに指定する必要があります。オプションは、keyword=value のペアで定義します。オプションが複数回指定された場合は、最後のオカレンスが使用されます。キーワードは、レコードの第 1 列で開始する必要があります。等号記号(=)の前または後のブランクは、許可されません。アスタリスク(\*)は、コメントと見なされます。

以下のオプションがサポートされます。

#### SEPCHAR=char

各列を分離するために使用する分離文字を定義します。これを省略した場合のデフォルトは、セミコロン (;) です。

#### SYSID

フィルターとして使用する SMF システム ID を定義します。そのシステムで生成された SMF レコードのみが取り込まれます。可能な値は 1-4 文字です。

#### FROM=date

フィルターとして使用する開始日。形式は YYYYMMDD です。指定された日付以降に書き込まれたすべての SMF レコードが取り込まれます。

#### TO=date

フィルターとして使用する終了日。形式は YYYYMMDD です。指定された日付までに書き込まれたすべての SMF レコードが取り込まれます。

#### RESOURCE=

フィルターとして使用するリソースを自動化マネージャーの表記で定義します。最大 10 個のリソース名を指定できます。この名前には、\*abc、abc\* または \*abc\* などのワイルドカードを使用できます。

## 戻りコード

ユーティリティにより、以下の戻りコードが設定されます。

**0**

正常終了。

**8**

USRPARMs データ・セット内で無効なオプションが検出されました。

**12**

REPORT データ・セットにアクセスできません。

**16**

重大エラー (例えば、SMFDATA データ・セットのオープン・エラー、あるいは REPORT ファイルへのレコードの書き込みなど) が発生しました。

## DB2 への SMF レポートの書き込み

---

IBM Tivoli System Automation Application Manager で提供および利用される DB2 テーブルに、INGPUSMF で作成した SMF レポートを変換し、書き込むことができます。

以下のレポートが用意されます。

### • 開始およびシャットダウン・レポート

- リソースについて、その依存関係を含む開始およびシャットダウンの累積時間を報告します。
- 選択されたドメイン内で、最も長い開始時間およびシャットダウン時間を要したリソースを報告します。

### • 可用性およびリカバリー・レポート

- リソースの実行時間とダウン時間、予期しない停止、および対応するリカバリー時間を報告します。
- 選択されたドメイン内で、予期しない停止の回数が最も多かったリソースを報告します。

z/OS オフローダーと呼ばれる変換ユーティリティは、これらのレポートを実行する際に必要な z/OS ドメイン・データを送信します。

z/OS オフローダー・コンポーネントはバッチ・ジョブとして実行され (67 ページの図 21 を参照)、エンドツーエンド自動化アダプター zFS ディレクトリー (通常は /usr/lpp/ing/adapters) にインストールされている既存のプログラムおよび新規プログラムを使用します。



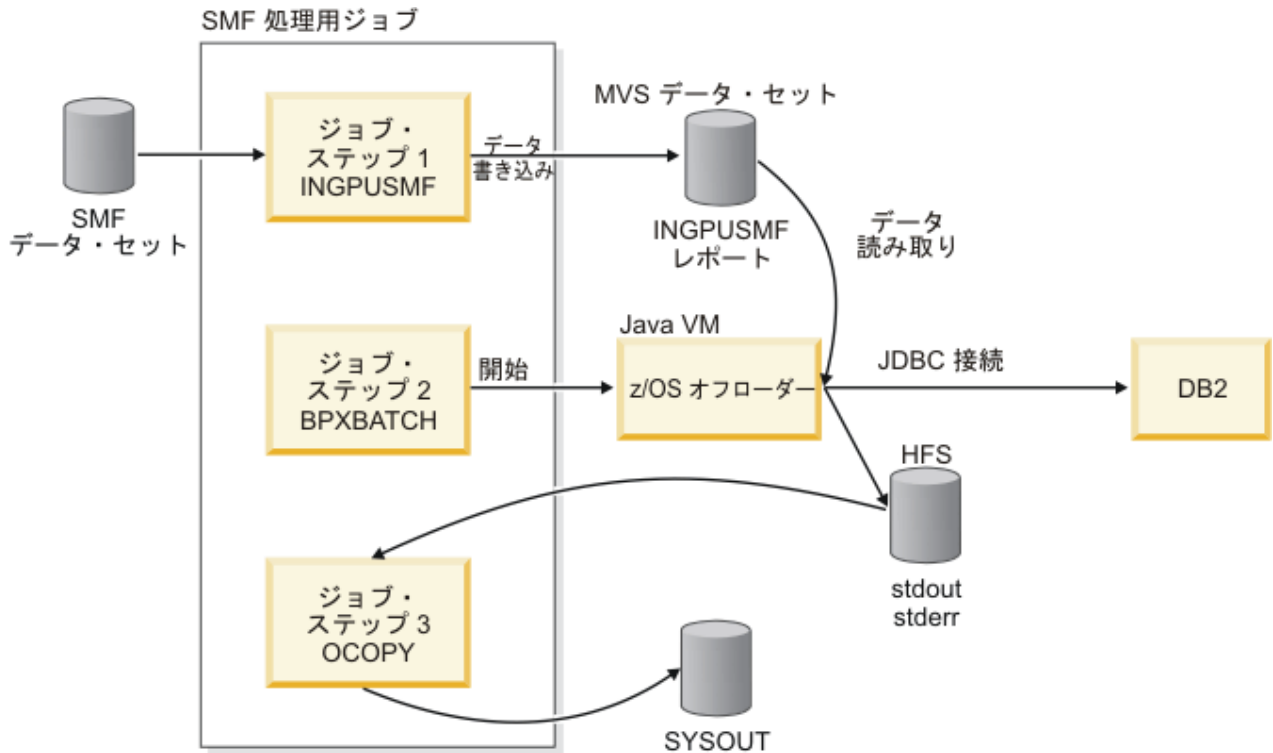


図 21. z/OS オフローダーでの SMF 処理

サンプル・ジョブ INGXRPT を使用して、z/OS オフローダーを実行できます。

## カスタマイズ

### 手順

z/OS オフローダーのインストール後に、次のカスタマイズのステップを実行する必要があります。

1. スクリプト /usr/lpp/ing/adapter/ingreport.sh をカスタマイズします。次のインストール・パスを適応させます。

```
INSTALL_DIR=/usr/lpp/ing/adapter
```

2. サンプル・ジョブ INGXRPT をコピーして、そこに説明されているステップに従います。  
 入力パラメーターには、正しく設定する必要があり、そうしないと変換ユーティリティが DB2 テーブルにアクセスできないものがいくつかあります。

パラメーター	詳細
INGDSN=HLQ.SMF.REPORT	INGPUSMF ユーティリティで作成される SMF レポートのデータ・セット。
INGSEPCHAR=;	これは、INGPUSMF ユーティリティで使われる分離文字でなければなりません。
INGDOMAIN=MyDomain	E2E アダプター構成ファイル ing.adapter.plugin.properties に指定された E2E ドメインの名前。省略された場合は、シスプレックス名および XCF グループ名から成るデフォルトが使用されます。
INGDB2_USER=db2inst1	リモート・ログオン用の DB2 ユーザー名。
INGDB2_PSW=db2admin	リモート・ログオン用の DB2 パスワード。
INGDB2_PORT=50000	リモート DB2 に接続する TCP/IP ポート。

パラメーター	詳細
INGDB2_SERVER=db2-host-name	リモート DB2 に接続する TCP/IP ホスト名。
INGDB2_NAME=EAUTODB	DB2 の名前、または、DB2 が z/OS に常駐する場合は DB2 の場所。
INGDB2_SCHEMA=EAUTOUSR	テーブルの DB2 スキーマ。

3. (オプション) データベースが z/OS システム上に配置されている場合は、DB2 ライセンス・ファイルが必要です。z/OS プラットフォーム用の適切なライセンス・ファイル db2jcc\_licencse\_\*.jar が、アプリケーション・クラスパスにインストールされている必要があります。下の表で定義されているように、z/OS データベースへの接続はライセンス・ファイルによって可能になります。

DB2 データベースの更新元 ==> 更新先	必要なライセンス・ファイル
分散システム ==> z/OS DB2	db2jcc_license_cisuz.jar
z/OS システム ==> z/OS DB2	db2jcc_license_cisuz.jar
z/OS システム ==> 分散 DB2	db2jcc_license_cu.jar

- a) 適切なライセンス・ファイルを、例えば DB2\_INSTALL\_PATH/db2/db2v8/jcc/classes/db2jcc\_license\_cisuz.jar から、ディレクトリー /usr/lpp/ing/adapter/lib にコピーします。
  - b) スクリプト /usr/lpp/ing/adapter/ingreport.sh 内のクラスパスを変更して、例えば以下のようにライセンス・ファイルを追加します。  
DB2\_LICENSE+\$INSTALL\_LIB/db2jcc\_license\_cu.jar
4. SMF レポートを DB2 にコピーする INGXRPT ジョブを実行します。

## 出力

ingreport.sh シェル・スクリプトの出力に、z/OS オフローダーの進行状況が示されます。発生したすべてのエラーがこの出力に報告されます。これらのメッセージ (INGX9850E、INGX9855E、および INGX9856E) について詳しくは、「IBM Z System Automation メッセージおよびコード」を参照してください。

## 第7章 プロセッサ操作制御リソースの自動化

この情報は、プロセッサ操作によって制御されるターゲット・システムからのメッセージの自動化が、ご使用の SA z/OS インストール済み環境で使用可能になるようにカスタマイズする方法について説明します。

これらのターゲット・システムまたはリソースは、以下の説明では **プロセッサ操作リソース** として参照されます。

プロセッサ操作 (フォーカル・ポイント・タイプ機能) によって、カップリング・ファシリティのイメージを含めて、プロセッサ・ハードウェアを、単一 NetView のプロセッサ操作フォーカル・ポイントからモニターして制御することができます。

注：

1. VM ゲスト・システムは、ProcOps によって制御される他のターゲット・システムと同様に扱われます (詳しくは、「*IBM Z System Automation* オペレーター・コマンド」を参照してください)。
2. PSM は「仮想」ハードウェアであるので、必ずしもすべてのターゲット・ハードウェア・コマンドが適用されるわけではありません (詳しくは、「*IBM Z System Automation* オペレーター・コマンド」を参照してください)。

この章で説明されている方法で、SA z/OS システム操作を使用してこれらのメッセージに対応することができます。この情報は 69 ページの『プロキシ定義を使用した z/OS ターゲット・システムのプロセッサ操作リソースの自動化』にあり、そのようなメッセージの自動化を実現するための一般的なプロセスを紹介しています。

注：

このセクションで説明する機能は、System Automation のシステム操作パネル (INGLIST や INGREQ など) からハードウェア・リソースを管理するための忠実性の低い方法です。ただし、この方法での以下のような制約のため、この機能の使用は推奨されません。

- すべてのターゲット・システムは、既存のプロセッサ操作パネルを今日使用する場合に得られるハードウェア状況の細分性が欠けたアプリケーション・リソースとして示されます。
- この機能が提供するメッセージ自動化機能は、IPL の過程での典型的なメッセージの自動化に重点を置いています。System Automation 製品コードがまだ扱っていないメッセージの場合、カスタム・メッセージ規則 (ユーザー定義の自動化テーブル、またはカスタマイズ・ダイアログからの自動化テーブル・オーバーライドの使用) の実装が必要です。

### プロキシ定義を使用した z/OS ターゲット・システムのプロセッサ操作リソースの自動化

SA z/OS プロセッサ操作を使用して、ターゲット・システム上では自動化できないメッセージを自動化できます。一般に、これらのメッセージには IPL 時に表示されるメッセージが含まれます。

シスプレックス環境では、シスプレックスに結合する場合は IPL 時に、システムがシスプレックスを離れる場合はシャットダウン時に、追加メッセージ (XCF WTOR) が表示されます。SA z/OS のシステム操作はその時点ではアクティブではないために、これらの WTOR はまだ自動化することができません。

この章で説明する XCF メッセージの自動化フレームワークにより、ユーザー独自の XCF メッセージの自動化を活用する方法をとることができます。

注：シスプレックス障害管理機能 (SFM) による自動化が可能な XCF WTOR メッセージがあります。これらの場合には、矛盾する自動化を避けるために、SA z/OS によってこれらのメッセージを自動化することはお勧めしません。

#### 概念

SA z/OS 標準インターフェースおよびルーチンを使用すると、システムの内部生成メッセージとほとんど同じ方法で、システムの外部メッセージを処理することができます。これは、自動化の際、メッセージの

自動化の制御手段と同様、カスタマイズ・ダイアログにおけるメッセージの自動化の定義手段にも適用されます。

メッセージ自動化にシステム操作メカニズムを活用するには、プロセッサ操作リソースを表すプロキシー・リソースを、項目タイプ・アプリケーション (APL) としてカスタマイズ・ダイアログで生成する必要があります。

プロキシーとプロセッサ操作リソース (ターゲット・システム) との間には、1対1の関係があります。この関係をカスタマイズ・ダイアログにインプリメントする方法について、以下の各項で説明します。

アクティブな SA z/OS が存在しない、または SA z/OS がまだアクティブになっていない外部システムで生成されたメッセージも自動化できます。これらのメッセージを生成するリソースをプロセッサ操作リソースと呼びます。これらは、項目タイプシステム (SYS) としてカスタマイズ・ダイアログに定義されます。

## プロキシー・リソースの自動化のカスタマイズ

既にカスタマイズ・ダイアログを使用してプロセッサ操作プロセッサ (PRO) を定義していることが前提となります。さらに、各プロセッサ項目内で、LPARS AND SYSTEMS ポリシー内で自動化するターゲット・システムを指定したことを前提としています。ポリシー (「*IBM Z System Automation* 自動化ポリシーの定義」の LPARS AND SYSTEMS ポリシー項目も参照) を作成し、プロセッサ操作フォーカル・ポイントに構成をロードすることにより、ハードウェアとそのターゲット・システムにプロセッサ操作がアクセスできるようになります。

ここで、プロセッサ操作フォーカル・ポイントで自動化するプロセッサ操作ターゲット・システムごとに、カスタマイズ・ダイアログで、プロセッサ操作リソースを項目タイプ System (SYS) として、および対応するプロキシー・リソースを項目タイプ Application (APL) として定義する必要があります。

多数のプロキシー・リソースを定義したい場合は、「*IBM Z System Automation* 自動化ポリシーの定義」に説明されているように、アプリケーション・クラスの使用することができます。

プロキシー・リソースをアプリケーション (APL) として定義することには、もう1つ利点があります。システムが INGLIST パネルに表示されるようになるため、システムをアプリケーション・リソースのように管理しモニターできるようになります。SA z/OS ユーザーは、メッセージの自動化をターゲット・システムのメッセージに使用できるだけでなく、始動コマンドや停止コマンドを実行してシステムの IPL やシャットダウンを行うこともできます。これらのコマンドは、アプリケーションの始動コマンドおよび停止コマンドと同様に定義することができます。アプリケーション・リソースとは異なり、ターゲット・システムは、プロセッサ操作コマンド (例えば、ISQCCMD *target\_system\_name* ACTIVATE FORCE(NO) または ISQSEND *target\_system\_name* OC vary xcf,*target\_system\_name*,off,retain=yes) によって管理されます。プロセッサ操作コマンドを使用すると、MVS コマンドをターゲット・システムに送信でき、またハードウェア・コマンドをプロセッサ (サポート・エレメントまたはハードウェア管理コンソール) に送信できます。

プロキシー・リソースを定義するときに従う必要のある規則は、以下のとおりです。

1. プロセッサ操作ターゲット・システムを項目タイプ System (SYS) として定義します。これらのシステムには、以下の規則が適用されます。

SYSTEM INFO ポリシーで、「**イメージ/システム名 (Image/System name)**」にプロセッサ操作ターゲット・システムの名前を使用します。「**イメージ/システム名 (Image/System name)**」はポリシー・データベース内で固有でなければなりません。

2. プロキシー・リソースを項目タイプ Application (APL) として定義します。これらのアプリケーションには、APPLICATION INFO ポリシー内で、以下の規則が適用されます。

- a. 「**ジョブ名 (Job Name)**」にプロセッサ操作ターゲット・システムの名前を使用します。「**ジョブ名 (Job Name)**」はポリシー・データベース内で固有でなければなりません。
- b. 「**ジョブ・タイプ (Job Type)**」に NONMVS を使用します。プロキシー・リソースのジョブ・タイプは常に、NONMVS でなければなりません。
- c. 「**システム名 (Sysname)**」にプロセッサ操作ターゲット・システムの名前を使用します。これは、上記の関連プロセッサ操作リソース SYS-entry の「**ジョブ名 (Job Name)**」および「**イメージ/システム名 (Image/System name)**」に指定した値と同じです。

- d. 「**モニター・ルーチン (Monitor Routine)**」に ISQMTSYS を使用します。ISQMTSYS は、ゲートウェイ通信、XCF 通信 (同一シスプレックス内)、またはハードウェア・コンソールを介した通信のいずれか (その順序で最初に成功した方式) を使用して、プロセッサ操作ターゲット・システムからシステム応答を取得しようとします。
  - e. プロキシ・リソースおよびプロセッサ操作ターゲット・システムを IPL 時に自動的に開始したくない場合は、「**IPL 後に再始動 (Restart after IPL)**」オプションを NO に設定します。
3. アプリケーションの自動化は、アプリケーション・グループ (APG) を介してそのアプリケーションをシステムにリンクすることによってのみ行うことができるため、プロキシ・アプリケーション用にアプリケーション・グループを定義する必要があります。「**タイプ (Type)**」SYSTEM および「**性質 (Nature)**」BASIC の APG を選択し、「**動作 (Behavior)**」を PASSIVE に設定し、「**自動化名 (Automation Name)**」を空のままにして、このグループのリソースが作成されないようにします。プロキシ・アプリケーションをこのアプリケーション・グループ内の他のアプリケーションにマージしないでください。マージされたアプリケーション・グループに適用される破壊的な要求が、そのグループに含まれるプロキシ・リソースにも影響を与えるためです。
  4. プロキシ・アプリケーションの「**メッセージ処理 (Message Processing)**」パネルで、自動化するメッセージを「**メッセージ ID**」列に定義します。メッセージ ID ISQ900I は指定しないでください。このメッセージは元のターゲット・システム・メッセージのキャリアとして使用されます。  
「**アクション (Action)**」列に cmd と入力して、定義したメッセージが発生した場合に処理されるコマンドを指定します。
  5. 自動化されるメッセージが WTOR の場合は、変数 &EHKVAR1 に応答 ID が入ります。この変数が次に ISQSEND コマンドへのパラメーターとして使用されます。

```
ISQSEND &SUBSJOB OC R &EHKVAR1,COUPLE=00
```

### 始動およびシャットダウンについての考慮事項

プロセッサ操作リソースを始動または停止するには、プロセッサ操作コマンドを使用する必要があります。例えば、次のように入力します。

- 始動の例:

```
ISQCCMD &SUBSJOB LOAD FORCE(NO)
```

- 停止の例:

```
Pass 1 ISQSEND &SUBSJOB OC Z EOD
```

```
-----  
Pass 2 ISQSEND &SUBSJOB OC VARY XCF,&SUBSAPPL,OFF,RETAIN=YES
```

注: パス 1 とパス 2 のコマンドの送信の間の遅延時間が適切でない場合は、「Application Automation Definition」パネルでリソース固有の「終了遅延」を定義することもできます。

プロセッサ操作コマンドについて詳しくは、「IBM Z System Automation オペレーター・コマンド」を参照してください。

### メッセージの自動化の準備

ターゲット・システムとの対話は、SA z/OS プロセッサ操作コンポーネントを基に行われます。したがって、このコンポーネントのインストールおよびカスタマイズは、この時点で完了する必要があります。

フォーカル・ポイントで受信を行うプロセッサ操作ターゲット・システムからのオペレーティング・システム・メッセージは、ISQ900I メッセージに転送されます。

ISQ901I は関係ありません。これはターゲット・システムのメッセージに関心のあるオペレーターに通知する際に使用されます。自動化目的では使用されません。

**CONSOLxx** メンバーの **MSCOPE()** パラメーター: MSCOPE により、シスプレックス内の次のようなシステムを指定することができます。すなわち、そのシステムからのメッセージは明示的にはこのコンソールにルーティングされていないのですが、この指定でコンソールはそのメッセージを受け取ることとなります。

アスタリスク (\*) は、この CONSOLE ステートメントが定義されるシステムを示します。デフォルトは \*ALL (シスプレックス内のすべてのシステムからの非送信請求メッセージを、このコンソールで受信することを示す) であるため、SA z/OS プロセッサ操作で正しく自動化するには、このパラメーターを「\*」に設定する必要があります。

## Linux コンソール・メッセージの自動化

### NetView への Linux コンソール接続

Linux® ターゲット・システムが IPL を行うとき、そのブート・メッセージが System z® または 390-CMOS プロセッサ・サポート・エレメント (SE) のコンソール統合機能 (CI) に表示されます。SA z/OS プロセッサ操作では、CI は、Linux オペレーティング・システムとの通信を行うためにサポートされる唯一のインターフェースです。

プロセッサ操作フォーカル・ポイントと CI との間の通信は、NetView RUNCMD、サポート・エレメントのオペレーター・コマンド機能 (OCF)、および SNA アプリケーションに基づいています。SA z/OS プロセッサ操作では、この接続パスは NetView 接続 (NVC) と呼ばれます。

### 大文字小文字混合の文字データを使用した Linux コンソールの自動化

コンソール・コマンドからの入力を大文字に変換するオペレーティング・システムとは異なり、Linux では大文字小文字が区別されます。NetView の自動化テーブル構文では、IF ステートメントの compare 引数内で大文字小文字の混合文字を使用できます。

このような比較の結果として自動化コマンドがスケジュールされると、NetView は渡されるメッセージ・トークンの引数を大文字に変換しません。ご使用の自動化ルーチンが、渡されるパラメーターの大文字変換を行わないことを確認してください。例えば、REXX では、ステートメント ARG P1 P2 ではなく PARSE ARG P1 P2 を使用します。これで、暗黙的に大文字への変換が行われます。Linux メッセージが自動化コードを呼び出し、メッセージ情報が NetView の GETMLINE 機能を使用して検索される場合、大文字変換は行われません。大/小文字混合コマンド・データを Linux コンソールに送信するには、以下の REXX ステートメントを使用してください。

```
Address Netvaxis 'ISQsend MYlinux 0c whoami'
```

アドレッシングされた REXX コマンド環境 Netvaxis は、大文字変換を行わずにコマンド・ストリングを渡します。ISQSEND コマンドは、内部的にその宛先パラメーターを MYLINUX および OC に変換しますが、コマンド whoami はそのまま残します。

### セキュリティー考慮事項

Linux システムの初期化後、通常は、LOGIN プロンプト・メッセージが表示され、定義されているユーザーがシステムにログインすることを許可します。ISQSEND コマンド・インターフェースは、パスワード・データの表示を抑止しません。

パスワード情報が NetView ログに表示されないようにするために、NetView LOG 抑止文字を使用できます。サポート・エレメント・ログ・ファイルでは、このようなパスワード・データをテキスト形式で表示できます。

### 制約事項および制限事項

以下の Linux システムがサポートされます。

- System z または 390-CMOS プロセッサ・ハードウェアの LPAR で実行されている Linux システム
- 基本モードに構成された System z または 390-CMOS プロセッサ・ハードウェアで実行されている Linux システム
- z/VM® バージョン 4.3 以上のもので VM ゲスト・マシンとして実行されている Linux システム

自身が VM ゲストとして実行される VM のもつて実行されている Linux システムはサポートされません。

Linux コンソールのコマンド・シェル環境では、Control-C のように、機能を実行するためにキーボード制御キーの組み合わせを押す代わりに、文字ストリングとして制御キーを渡すことができます。SA z/OS プ



ロセッサ操作の現在の Linux サポートについては、この Linux 機能を使用したテストはまだ済んでおりません。制御キーによるユーザー対話を必要とする Linux プログラムやコマンド・スクリプトは、SA z/OS プロセッサ操作 ISQSEND インターフェースを使用して呼び出さないでください。

## 自動化にプロセッサ操作メッセージを追加する方法

コンソールの自動化をインプリメントするには、NetView 自動化テーブル (AT) と SA z/OS コマンド・セットを使用します。特定のメッセージが生成されたときに、オペレーターが実行するルーチンの機能を自動化することができます。詳しくは、「*IBM Z System Automation* 自動化ポリシーの定義 (SD88-6151)」を参照してください。

### プロセッサ操作ターゲット・システムによって出されたメッセージ

ターゲット・システムがメッセージを出すと、そのメッセージは、プロセッサ操作フォーカル・ポイント・システムに転送されます。フォーカル・ポイント・システムは、そのメッセージを SA z/OS ISQ900I メッセージまたは ISQ901I メッセージ、あるいはその両方にパッケージし直し、次のように、該当タスクあてにそのメッセージのパスを指定します。

- ISQ900I メッセージは、SA z/OS プロセッサ操作自動タスクにルーティングされる。自分で作成した自動化で ISQ900I メッセージを受信したい場合、ISQEXEC コマンドを使用して、ターゲット制御タスクでその自動化を実行します。ISQEXEC コマンドの使用法については、11 ページの『その他の OCF コマンドの実行』の『自動化ルーチンのターゲット制御タスクへの送信』の節を参照してください。SA z/OS 用のユーザーの NetView 自動化テーブル項目では、プロセッサ操作フォーカル・ポイント・システムに転送されるすべてのターゲット・システム・メッセージに関して、ISQ900I ID の確認を通知する必要があります。ISQ900I 自動化テーブル項目をターゲット・システムに固有になるように指定できますが、これはお勧めできません。
- ISQ901I メッセージは、ISQXMON コマンドで関係オペレーターとして識別されている、あるいは、カスタマイズ・ダイアログ内で関係オペレーターとしてマークされているすべてのログオン・オペレーターにルーティングされます。

ISQEXEC コマンドおよび ISQXMON コマンドについては、「*IBM Z System Automation* オペレーター・コマンド」を参照してください。

SNMP 接続から転送されるメッセージは、以下のもので構成されます。

- ISQ900I または ISQ901I メッセージ ID
- メッセージの起点となったターゲット・システムの名前
- メッセージの起点を説明しているコンソール指定機能形式
- ターゲット・システムからの元のメッセージのメッセージ ID および テキスト

例えば、NetView 接続でメッセージ「IEA101A SPECIFY SYSTEM PARAMETERS」をオペレーティング・システムからフォーカル・ポイント・システムに転送すると、SA z/OS は、次の SA z/OS メッセージのいずれか、あるいは両方を作成します。

```
ISQ900I target-system-name OC IEA101A SPECIFY SYSTEM PARAMETERS
ISQ901I target-system-name OC IEA101A SPECIFY SYSTEM PARAMETERS
```

このメッセージ・フォーマットは、すべてのプロセッサ操作ターゲット・システム・メッセージに適用されます。これは、元のメッセージを生成したターゲット・システム・リソースとは独立しています。

プロセッサ操作ターゲット・システム・メッセージは、プロセッサ・サポート・エレメント (SE) またはハードウェア管理コンソールで表示されるのと同じフォーマットで送信されます。

**VM 第 2 レベル・システムの詳細:** ゲスト・マシンのオペレーティング・システムからのメッセージは、以下のフォーマットで表示されます。

```
ISQ900I psm-name.guest-name OC IEA101A SPECIFY SYSTEM PARAMETERS
```

仮想マシンの CP からのメッセージは、以下のフォーマットで表示されます。

```
ISQ900I psm-name.guest.name OC HCPGSP2627I The virtual machine is
placed in CP mode due to a SIGP initial CPU reset from CPU 00.
```

PSM 自体からのメッセージは、以下のフォーマットで表示されます。

```
ISQ700I psm-name SC ISQCS0314E Message Handler has failed.
```

注：コンソールに表示されるメッセージが、ユーザーが予期し、その内容に応じて NetView 自動化テーブル項目に書き込んだフォーマットであることを確認してください。

### サンプル NetView 自動化テーブル・ステートメント

以下のメッセージ応答例は、メッセージ ID スtringに「IEA101A」が含まれているときのシステム・パラメーターについての要求を表しています。

```
IF      TEXT = . 'IEA101A SPECIFY SYSTEM PARAMETERS'  
& MSGID = 'ISQ900I' .  
THEN   EXEC( CMD('ISQI101 ' ) ROUTE ( ONE * ) )  
        DISPLAY(N) NETLOG(Y);
```

この NetView 自動化テーブル・ステートメントは、メッセージ条件が真であると、ISQI101 ルーチンを開始します。

注：メッセージ内のテキストは、大/小文字混合でかまいません。必ず、テキストは大/小文字混合になっているものとしてコーディングしてください。

### メッセージ ISQ211I

一部の SA z/OS コマンドは、ポートのロックとアンロックを試みます。オペレーターがポートのロックを所有していると、SA z/OS アンロック・コマンド ISQXUNL は、メッセージ ISQ211I「Unable to unlock target name console」に関連付けられている RC=12 を戻します。

このような場合は、ISQOVRD コマンドを使用してアンロックを強制するか、あるいは、自動化を終了してメッセージを出すかのいずれかを選択することができます。その後、NetView ログを表示して、ポートのロックの理由を突き止めることができます。

自動化では、このメッセージ ISQ211I が頻繁に表示されることがあります。ロックされたポートのアンロックを試みることはエラー条件ではありません。しかし、呼び出しコマンドが正常に終了しなかったことを示している可能性があります。自動化は、ISQ211I メッセージからではなく、コマンドが実行されなかったことを明確に示すメッセージからスケジュールしてください。

### プロセッサ操作コマンド・メッセージ

一部の SA z/OS コマンドはターゲット・システム上で実行されます。これらのコマンドから戻されるメッセージは、サポート・エレメントが操作をスケジュールするように指示したことしか示しません。したがって、ターゲット・システムにおける操作は、SA z/OS メッセージで正常に完了したことを示している場合でも、完了していないことがあります。

SA z/OS は、コマンドが、サポート・エレメントに正常に転送されたことだけを確認します。ターゲット・システムで操作が失敗すると、サポート・エレメントがフォーカル・ポイント・システムに転送する非送信請求メッセージが ISQ900I メッセージの中に生成されます。自動化は、ターゲット・システム操作が完了したか、完了しなかったかを明確に示すメッセージからスケジュールしてください。

SINGSAMP SA z/OS サンプル・ライブラリーには、選択されたメッセージへの応答を出す複数の自動化ルーチンの PL/I ソース・コードが収められています。エンタープライズに最も適した応答を選択することができます。また、このサンプルをモデルとして使用して、ユーザー独自の自動化ルーチンを作成することもできます。75 ページの表 12 のリストは、これらのルーチン、そのルーチンが応答するメッセージ、および最初に出す応答を要約したものです。



表 12. SINGSAMP SA z/OS サンプル・ライブラリー・ルーチン		
SINGSAMP メンバー	ルーチン	説明
INGEI120	ISQI120	以下のメッセージに応答します。 <b>IEA120A</b> Device ddd volid read, reply cont or wait. <b>IOS120A</b> Device ddd shared (PR volid not read.) the recovery task, reply cont or wait. ターゲットに次の応答を出します: CONT
INGEI357	ISQI357	以下のメッセージに応答します。 <b>IEE357A</b> Reply with SMF values or U. ターゲットに次の応答を出します: U
INGEI426	ISQI426	以下のメッセージに応答します。 <b>\$HASP426</b> Specify options - subsystem_id. ターゲットに次の応答を出します: WARM,NOREQ
INGEI502	ISQI502	以下のメッセージに応答します。 <b>ICH502A</b> Specify name for primary/backup RACF data set sequence nnn or none. ターゲットに次の応答を出します: NONE
INGEI877	ISQI877	以下のメッセージに応答します。 <b>IEA877A</b> Specify full DASD SYS1.DUMP data sets to be emptied, tape units to be used as SYS1.DUMP data sets or GO. ターゲットに次の応答を出します: GO
INGEI956	ISQI956	以下のメッセージに応答します。 <b>IEE956A</b> Reply - ftime = hh.mm.ss, name = operator,reason = (ipl,reason) or u. ターゲットに次の応答を出します: U

SINGPARM データ・セットの ISQMSG0 メンバー内の SA z/OS 自動化テーブル項目には、これらの自動化ルーチンを呼び出す非アクティブ項目が組み込まれています。これらのルーチンをユーザーの自動化に組み込むには、以下のことを行います。

1. 使用したい自動化ルーチンを開始するメッセージについての対応する自動化テーブル項目からコメントを除去します。これらのステップを初期 SA z/OS インストールの一部として実行する場合、SA z/OS 項目を組み込む前に、これらの変更を行います。初期 SA z/OS インストール後にこれを行うには、NetView 自動化テーブルを変更します。
2. 必要な応答を出すのに使用するルーチンをコーディングします。
3. 使用したいルーチンの PL/I ソース・コードをコンパイルし、結果としてできるオブジェクト・コードを PL/I ライブラリーとリンクします。
4. NetView プログラムをリサイクルして、新規項目を活動化します。

自動化処理を実行するには、それぞれのフォーカル・ポイント・システムとターゲット・システムの NetView 自動化テーブルの各メッセージを、システムの NetView プログラムで使用できるようにしておく必要があります。z/OS の場合、MPF が NetView プログラムに対するメッセージの可用性を制御します。自動化に必要なメッセージがマークされているかを確認するには、SYS1.PARMLIB データ・セットの中の MPF リスト・メンバーを調べてください。他のオペレーティング・システムを使用しているターゲット・システムの場合、システム上で使用しているメッセージ抑制機能を調べてください。

## メッセージのテスト

SA z/OS は、ご使用の SA z/OS 構成用の NetView 自動化テーブル項目の 集合を提供します。NetView 自動化テーブル項目は、SA z/OS SINGPARM インストール・データ・セットの AOFPCMD メンバーに入っています。これらの項目を NetView 自動化テーブルに移動すると、追加編集が必要なことがあります。

例えば、実動 NetView 自動化テーブルで、特定メッセージを既にテストしている可能性があります。その同一メッセージをテストする項目を追加すると、自動化テーブルは予想どおりには実行されないこととなります。テスト基準との一致が見つかり、自動化テーブルの検索は中止されます。2 番目の NetView 自動化テーブル・ステートメントは見つかりません。したがって、メッセージはユーザーが必要とするアクションすべてを駆動するわけではありません。

これを避けるには、項目を単一テスト条件に結合します。これにより、必要なアクションはすべて、全メッセージ用にスケジュールされます。以下のメッセージの場合、

```
IEA320A RESPECIFY PARAMETERS OR CANCEL
```

NetView 自動化テーブルには、既に次の項目が作成されている場合があります。❶

```
IF      MSGID = 'IEA320A'
THEN    EXEC (CMD('USERJOB') ROUTE( ONE * ) ) CONINUE(Y);
```

SA z/OS をインストールすることによって、System z または 390-CMOS プロセッサ・ハードウェアから転送されたときに次のメッセージが表示されます。

```
ISQ900I SYS1 OC IEA320A RESPECIFY PARAMETERS OR CANCEL
```

SA z/OS 項目を追加した後で、NetView 自動化テーブルには、以下の項目が含まれます。

```
IF      TEXT = . 'IEA320A RESPECIFY PARAMETERS' .
        & MSGID = 'ISQ900I' .
THEN    EXEC (CMD('ISQI320 ' ) ROUTE( ONE * ) )
        DISPLAY(N) NETLOG(Y);
```

この場合、最初の項目が IF テストを満たし、コマンド USERJOB が実行されます❶。メッセージがテーブル項目と一致すると、自動タスクが検索を停止するので、2 番目のコマンド ISQI320 は実行をスケジュールされません。これら 2 つの項目を、以下のように単一項目に結合します。

```
IF      TEXT = . 'IEA320A RESPECIFY PARAMETERS' .
        & MSGID = 'ISQ900I' .
THEN    EXEC(CMD('ISQI320 ' ) ROUTE( ONE * ) )
        EXEC(CMD('USERJOB ' ) ROUTE( ONE * ) )
        DISPLAY(N) NETLOG(Y);
```

2 番目の例を使用すると、コマンドは両方ともスケジュールされます。

ご使用の NetView 自動化テーブルが SA z/OS メッセージのテキストをテストする場合、このメッセージ形式は、テストする文字ケースと一致している必要があります。これは、すべてのサイトでメッセージに同一形式を使用するように要求したり、大文字形式と大/小文字混合形式の AT 項目を重複させたりすることによって実現できます。

## 新規自動化定義の作成

カスタマイズ・ダイアログを使用して、メッセージ応答と自動化オペレーター情報を自動化ポリシーに追加し終わったら、システム操作制御ファイルを作成する必要があります。

これらのファイルの作成および配布の方法に関する詳細については、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。

SA z/OS 作成機能は、新規自動化定義を「Build Parameters (ACF and AMC Build Parameters)」パネルで定義したデータ・セットの中に入れます。

新規自動化定義は、NetView 始動プロシージャの SA z/OS NetView DSIPARM 連結の中にコピーするか、あるいは、NetView DSIPARM データ・セットに連結します。

## 変更済み自動化環境のロード

AMC ファイルを再ロードするために、自動化制御ファイルおよび AT は、以下のアクションを実行します。

MPF リストを再ロードするには、次のようにコマンドを入力します。

- z/OS コンソールから入力する場合:

```
SET MPF=xx
```

- NetView コンソールから MVS 接頭部を使用して入力する場合:

```
MVS SET MPF=xx
```

この xx は、ロードする SYS1.PARMLIB データ・セットの MPF メンバーの接尾部です。

自動化マネージャー構成ファイルを再ロードするために、すべての更新された自動化制御ファイルおよび自動化テーブルは、次のコマンドを出します。

```
INGAMS REFRESH
```

データ・セット名を指定するか、(現行のデータ・セットを再ロードするという意味の) アスタリスク (\*) を指定します。

## 同期 HW コマンドへのパイプおよび ISQCCMD の使用

プロセッサ操作 SNMP 接続、および制限付きで BCP 内部インターフェース 接続に使用可能な Z System Automation ハードウェア・インターフェース・コマンドである ISQCCMD を使用すると、プロセッサと論理区画、およびハードウェア活動化プロファイルを管理し、制御することができます。

自動化プロシージャ内で使用する場合、ISQCCMD は、プロセッサ操作の管理および構成タスクを自動化するための使いやすいインターフェースを提供します。

以下の HW コマンドは、コマンドが完了すると直ちに、すべての応答情報を NetView に返すので、同期コマンドと呼ばれます。

- CCNTL
- CONDATA
- CPCDATA
- GETCLUSTER
- GETISTAT
- GETIINFO
- GETSDGA
- GETSINFO
- GETSSTAT
- ICNTL
- PROFILE

## ISQCCMD および PIPES の使用による非同期ハードウェア・コマンドの自動化

- STPDATA
- TCDATA

SNMP および BCPii 接続の場合、ISQCCMD は NetView PIPES をサポートします。ISQCCMD コマンドが完了すると、以下の例に示すように、ISQ.SNMP という名前の PIPE KEEP に、実行された HW コマンドの即時コマンド応答が含まれます。

```
* ISQCCMD G14 GETSINFO
| ISQ417I GETSINFO STATUS(SUCCESS)
| ISQ900I G14.KEY3 SC AOFA0017 GETSINFO G14 STATUS(OPERATING) PDATA(TYPE(2084)
,MODEL(B16),S/N(000020016F7A)) MODE(LPAR) APROF() CPCSNAM(IBM390PS.G14) NAME(G14)
TSTIME(070825131936)
| ISQ419I ISQCCMD GETSINFO processing on G14 is complete.
* IPSFO PIPE KEEP ISQ.SNMP | CONS
| IPSFO AOFA0017 GETSINFO G14 STATUS(OPERATING)
PDATA(TYPE(2084),MODEL(B16),S/N(000020016F7A)) MODE(LPAR) APROF()
CPCSNAM(IBM390PS.G14) NAME(G14) TSTIME(070825131936)
```

上記の例では、HW 共通コマンド GETSINFO が NetView コンソールで実行されました。「ISQ」メッセージに組み込まれた、ハードウェアからの応答がコンソールに表示されます (レポート ID AOFA0017 から開始)。

上記の例に示すように、ISQCCMD コマンドが完了して ISQ.SNMP の内容がコンソールに表示されると、ISQ.SNMP という名前の PIPE KEEP を参照した場合に同じ情報を得ることができます。

自動化プロシージャ内で、次の例に示すように、上記をコーディングすることができます。

```
/*ReXX*/
/* Display CPC information using the ISQ.SNMP KEEP */
Arg cpcname
'ISQCCMD 'cpcname' GETSINFO'
If RC = 0 Then Do
  'PIPE KEEP ISQ.SNMP ' ,
  ' | LOC /AOFA0017/ ' ,
  ' | LOC /'cpcname'/ ' ,
  ' | CONS ONLY'
End
```

別の方法として、PIPE NETVIEW ステージの後に EXPOSE TOTRAP ステージを続けて使用する場合は、即時 ISQCCMD HW 応答を直接 PIPE 入力ストリームに入れることができます。この場合、すべての ISQ メッセージと AOFA0017 レポート・データを PIPE 処理に使用することができます。

```
/*ReXX*/
/* Display CPC information in a PIPE */
Arg cpcname
'PIPE NETV ISQCCMD 'cpcname' GETSINFO' ,
' | EXPOSE TOTRAP ' ,
' | LOC /ISQ90/ ' , /* takes ISQ901I or ISQ900I */
' | LOC /AOFA0017/ ' ,
' | LOC /'cpcname'/ ' ,
' | CONS ONLY'
```

## ISQCCMD および PIPES の使用による非同期ハードウェア・コマンドの自動化

以下の ISQCCMD ハードウェア・コマンドは、NetView に 2 つのメッセージを戻します。最初のメッセージは、HW コマンドの実行が受け入れられたか、またはリジェクトされたかのメッセージです。受け入れメッセージが発行された場合、次に、実際のコマンドの正常実行または失敗の情報がいった完了イベント・メッセージが非同期に送信されます。

- ACTIVATE
- CBU
- CTRLCONS
- DEACTIVATE
- EXTINT
- LOAD
- OOCOD

- RESERVE
- RESTART
- START
- STOP
- STP
- SYSRESET
- TCM

ISQCCMD インターフェースを使用する自動化スクリプトでは、非同期 HW コマンドの受け入れまたはリジェクトの応答と、実際のコマンド完了情報 (正常実行または失敗のいずれかが示されます) は区別する必要があります。ハードウェアからの非同期コマンド完了イベントは、メッセージ自動化および ProcOps の TRAP AND WAIT 処理で使用可能になりました。ISQCCMD インターフェースを使用するアプリケーション・スクリプトでは、ACCEPTED または REJECTED の応答を ISQCCMD 終了時に直接入手することができます。ACCEPTED 応答は、その後のコマンド完了イベント・メッセージの待機に使用されることがあります。

SINGSAMP ライブラリーのメンバー INGEI004 は、TRAP AND WAIT とともに ISQCCMD および NetView PIPES を使用して非同期ハードウェア・コマンドを自動化する方法を示す REXX サンプルを提供します。

## VM 第 2 レベル・システム・サポート (VM Second Level Systems Support)

この機能では、VM 下で稼働中のゲスト・マシンを制御およびモニターするための ProcOps サポートが提供されます。

ProcOps によって、オペレーティング・システムをプロセッサ中の他の機能間に IPL することができます。このようなオペレーティング・システムの 1 つに VM があります。VM 内では、他のオペレーティング・システムをゲスト・マシンとして IPL することができます。特に対象マシンであるのは LINUX ゲスト・マシンですが、MVS、VSE、さらに VM ゲスト・マシンも可能です。(これより低いレベルのゲスト・マシンは対象外です)。以前は、こうしたゲスト・ターゲット・システムとの間でコマンドを入力し、またメッセージを受信することによって、そのシステムが正しく IPL されているか、あるいは正しく動作しているかを検査するのに、効果的な方法がありませんでした。

第 2 レベル・ゲスト・マシン・サポートを使用すると、以下のことを行うことができます。

- ゲスト・マシン自体から出されたメッセージを取り込んで、それを表示、または自動化処理、あるいはその両方を行うために、ProcOps プロセスに送り返す。
- コマンドをオペレーター要求または自動化アクションのいずれかとして ProcOps からゲスト・マシンに送信する。

### ゲスト・ターゲット・システム

ターゲット・システムとして最も使用される可能性が高いゲスト・マシンは、LINUX システムです。LINUX マシンに 2 次ユーザーがいる場合には、その 2 次ユーザーは、以下の目的で CP SEND コマンドを使用できます。

- ゲスト・マシンに CP コマンドを出す。
- ユーザーとして LINUX にログオンする。
- LINUX コマンドを入力する (ログオン後)。

(LINUX システムを、LINUX にログオンせずに、VM コンソールに LINUX コマンドを入力できるような方法でセットアップすることもできます。)

2 次ユーザーは、以下のものを受信します。

- すべての「ブート・アップ・メッセージ」
- ゲスト・マシンで実行された CP コマンドに対する応答
- ログオンおよび LINUX コマンドに対する応答

MVS マシンはより複雑です。MVS マシンの稼働中、元の VM ユーザーはまず NIP コンソールとなつてから、次に MCS コンソールとなります。こうしたコンソール・モードでは、コンソールとの間でのすべての入出力を MVS が担当し、そのコンソールに送られた MVS メッセージは、どの CP 機能も代行受信することはできません。このため、SCIF SEND コマンドを使用してコマンドを MVS に送信することも、このコンソールへの MVS メッセージを代行受信することもできません。

しかしゲスト・マシンの「仮想 SCLP コンソール」は使用できます。初期化の NIP フェーズの間、ゲスト仮想マシンを使用可能な 3270 コンソールがないように構成して、このコンソールを強制的に使用させることができます。これにより、NIP は、ゲスト・マシンへのそのメッセージを行モード・コマンドとして送ります。これは、論理区画内で稼働中の MVS システムにより、HMC 上のオペレーティング・システム・メッセージ (OSM) ウィンドウに送られるメッセージ・ストリームと似ています。

どの NIP メッセージに対する応答も、CP VINPUT コマンドを使用して入力されます。これは、ターゲット・システムのオペレーター・コンソール (OC) に ISQSEND コマンドが出されると、内部的に行われます。こうした VINPUT コマンドが正しく処理されるように、ゲスト・マシンは、この時点で RUN ON 状態で作動している必要があります。

RUN ON 状態が確実に設定されるように、このゲスト・マシンが PSM によって始動された時点で、CP SET RUN ON コマンドがすべての MVS ゲスト・マシンに送られます。

MCS 操作が確立されると、オペレーターのアクションを必要とする重要なメッセージがゲスト・マシンに送られます。この場合もまた、HMC の OSM ウィンドウに送られるメッセージ・ストリームと似ています。最初、MVS に対してコマンドを入力することはできません。コマンドを入力するには、「問題判別」モードに入る必要があります。このモードに入るには、VARY CONSOLE(\*),ACTIVATE コマンドを入力しなければなりません。このコマンドが実行されると、以下のようになります。

- 表示されるすべての MVS メッセージがゲスト・マシンに経路指定されます。
- CP VINPUT コマンドを使用して、コマンドを入力できます。

問題判別は、一般的にはお勧めしません。

LINUX コマンドを入力するには、通常 LINUX にログオンする必要があります。これにはユーザー ID とパスワードが必要です。そこで LINUX コマンドに備えるために、ProcOps に対するユーザー ID とパスワードの指定が必要になり、さらにセキュリティの分野であらゆる難しさを伴います。現時点では、指定のメッセージが表示されたら、LINUX システムは IPL 完了とみなされます。ユーザーのログオンは必要ありません。

VM マシンもゲスト・マシンとすることができます。第 3 レベル・ゲスト・マシンはサポートされていません。

VSE マシンもゲスト・マシンとすることができます。

## ターゲット・システムのカスタマイズ

### LINUX

LINUX ターゲット・システムには、その VM ディレクトリー項目内に CONSOLE ステートメントが必要です。このステートメントは、その PSM をデフォルトの 2 次ユーザーとして設定します。例えば、仮想マシン LNXA01 が仮想マシン ISQPSM1 で稼働中の PSM によって制御されている場合には、その CONSOLE ステートメントは以下のようになります。

```
CONSOLE 009 3215 T ISQPSM1
```

LINUX ターゲット・システムを非活動化する場合には、FORCE コマンドが使用され、遮断されます。デフォルトのゲスト・シグナルのタイムアウト間隔値 (SET SIGNAL コマンドで設定) とゲスト・マシンに定義された値によって、LINUX システムを通常どおりに遮断できるようにするときを使用される間隔が決定されます。ゲストにこの機能が必要な場合には、この機能が的確に設定されるようにしなければなりません。

このようなアクションには、LINUX システム自体における etc/inittab 項目の更新、および VM ホスト上での SHUTTRAP モジュールのセットアップが含まれる場合があります。

### MVS

PSM を 2 次ユーザーとして定義する CONSOLE ステートメントが VM ディレクトリー項目に必要です。

```
CONSOLE 01F 3270 T ISQPSM1
```

さらに、初期アクションとして CMS システムを IPL する必要があります。この CMS システムが IPL されると、そのシステムは、以下のようなステートメントが含まれる PROFILE EXEC を実行するはずで

```
SET RUN ON
DETACH 01F
IPL 7700
```

応答が NIP コンソールに送られるときに、使用される VINPUT コマンドが有効となるように、SET RUN ON が必要です。

MVS システムの IPL の際には、そのシステムが使用可能な定義済みのどの 3270 コンソールも検出しないようにするために、DETACH を使用します。(どのユーザーも NIP または MCS コンソールとして定義されたアドレスに、VM DIAL を出さないようにする必要があります。)

MVS システムを IPL するには、IPL コマンドが使用されます。

IPL 後にコマンドを MVS に入力できるように、MVS システムそのものに、SYSCONS 用の CONSOLE ステートメントをアクティブ CONSOLxx 定義に組み込んでおく必要があります。以下はその例です。

```
CONSOLE DEVNUM(SYSCONS)
         ROUTCODE(ALL)
         AUTH(MASTER)
         MSCOPE(*)
         CMDSYS(*)
         UD(Y)
```

## VM

PSM を 2 次ユーザーとして定義する CONSOLE ステートメントが VM ディレクトリー項目に必要です。

```
CONSOLE 01F 3270 T ISQPSM1
```

さらに、初期アクションとして CMS システムを IPL する必要があります。この CMS システムが IPL されると、そのシステムは、以下のようなステートメントが含まれる PROFILE EXEC を実行するはずで

```
SET RUN ON
DETACH 01F
IPL 7700
```

応答がコンソールに送られるときに、使用される VINPUT コマンドが有効となるように、SET RUN ON が必要です。

VM システムが IPL する際、そのシステムが使用可能な定義済みのどの 3270 コンソールも見つけれられないようにするために、DETACH が使用されます。(どのユーザーもオペレーター・コンソールとして定義されたアドレスに、VM DIAL を出さないようにする必要があります。)

VM システムを IPL するには、IPL コマンドが使用されます。

VM システム自体には、SYSTEM CONFIG ファイル(「parm disk」にあります)内の OPERATOR\_CONSOLES ステートメント内に、エミュレーションされたシステム・コンソールの指定が含まれている必要があります。以下はその例です。

```
OPERATOR _CONSOLES 01F 020 System_Console
```

これにより、VM が IPL し、使用可能な正規のコンソールを見つけられない場合に、エミュレーションされたシステム・コンソールが使用されるようになります。このコンソールが代わりに、メッセージを行モード・メッセージのストリームとして 2 次ユーザーに送ります。

## VSE

PSM を 2 次ユーザーとして定義する CONSOLE ステートメントが VM ディレクトリー項目に必要です。

```
CONSOLE 01F 3270 T ISQPSM1
```

さらに、初期アクションとして CMS システムを IPL する必要があります。この CMS システムが IPL されると、そのシステムは、以下のようなステートメントが含まれる PROFILE EXEC を実行するはずで

```
TERM CONMODE 3215
IPL 7700
```

TERM CONMODE 3215 コマンドは、コンソールを行モードに設定します。

## GETRAW 照会の SNMP MIB 属性を見つけるためのヒント

ISQCCMD 共通コマンド GETRAW は、SE コンソールまたは HMC コンソールから CPC および CPC イメージ (LPAR) 情報を取得するための高速低水準プログラミング・インターフェースです。アクティブ ProcOps SNMP 接続を使用し、データを 16 進値のシーケンスとして返します。ISQCCMD、GETRAW 共通コマンド、およびサポートされるパラメーターについては、「オペレーター・コマンド」または SA z/OS オンライン・ヘルプを参照してください。

以下の情報を使用して、照会をセットアップするためのオブジェクト属性情報を検索します。SNMP MIB 変数およびオブジェクト ID を基本的に理解しておくことをお勧めします。

SA z/OS サンプル・ライブラリーのメンバー INGEI007 は、このプログラミング・インターフェースを REXX で使用する方法をいくつかの ISQCCMD GETRAW 呼び出しで示しています。サンプルでは、さまざまな属性を取得しています。

IBM Z CPC および CPC イメージ MIB 変数とそのオブジェクト ID およびサフィックスについては、[IBM Z SNMP Application Programming Interfaces \(SB10-7171-xx\) 資料](#) (IBM Resource Link から入手可能) にすべて記載されています。

トピック	<i>IBM Z SNMP Application Programming Interfaces</i> マニュアル内の位置
Attributes with specific mainframe/console levels	About this publication > Summary of SNMP API support updates Appendix E. Object Attribute Availability
List of supported suffixes for attributes in ascending order	Chapter 3. Console application APIs > Data exchange APIs and commands API structures and definitions > Constant definitions.
Detailed object descriptions for the GETRAW supported CPC and CPC image (LPAR) objects, including the data types returned	Chapter 4. Console application managed objects > Defined CPC Chapter 4. Console application managed objects > CPC image



## 第 8 章 USS リソースの自動化

この情報では、z/OS UNIX システム・サービスを SA z/OS に統合する方法および z/OS UNIX 自動化のセットアップ方法について説明し、z/OS UNIX 自動化を使用する際のヒントを記述します。

### z/OS UNIX システム・サービスの統合

z/OS UNIX アプリケーションでは、次の機能が SA z/OS によりサポートされています。

- アプリケーションの始動と停止
- 次に対するモニター:
  - 処理 (コマンドまたはパスとユーザー ID で表現される)
  - TCP ポート
  - ファイルとファイル・システム
  - 一般ユーザーのモニター (ユーザーが z/OS UNIX モニター・ルーチンまたはスクリプトを提供する)
- z/OS UNIX コマンド (INGUSS コマンド) を実行する API の使用

### インフラストラクチャーの概要

自動化を行う必要がある z/OS UNIX リソースは、SA z/OS で既に自動化されている z/OS システムの z/OS UNIX で実行しなければなりません。自動化マネージャーの全体像からみると、このシステムの NetView エージェントが z/OS UNIX リソースに対して責任を負います。

INGUSS を介したコマンドの実行またはユーザー定義モニターでは、z/OS UNIX プログラム (SA z/OS により提供される) は SA z/OS によって直接起動されます。このプログラム (ingccmd) は UNIX コマンドを実行し、ジョブ名 INGCUNIX の SA z/OS により開始された時に実行します。ingccmd プログラムは、z/OS UNIX に対する NetView ベースのエージェントの拡張子です。標準 z/OS UNIX リソース (プロセス、ポート、またはファイル) をモニターするために、SA z/OS 内部ルーチンが開始されます。

USS リソースのプロセス初期化および終了状況の更新は、プログラム間インターフェース INGUXPPI によってシステム出口から SA z/OS 環境に直接報告されます。同じ名前を持つ NetView タスクによって UP または DOWN 状況が即時にポストされます。自動化エージェントが認識し、そのリソースの正しい自動化状況を設定します。

この機能のために、NetView サブシステム・インターフェース (SSI) が必要になります。SSI の正しいカスタマイズについては、「IBM Z System Automation 計画とインストール」の『ホスト・システムへの SA z/OS のインストール』の章のステップ 5 を参照してください。

USS のモニタリングで、それがダウンしていることが示されると、USS の状況が AUTODOWN に更新されます。しかし、USS プロセスが終了するまでに時間がかかることがあるため (獲得されたリソースをクリーンアップするため)、クリーンアップ遅延後にモニタリングが繰り返されます。独自の USS プロセスを定義する場合は、APPLICATION INFO ポリシー項目を使用して、適切なクリーンアップ遅延を指定してください。複数のプロセスを定義する必要がある場合は、アプリケーション・クラスの使用を検討してください。

### z/OS UNIX 自動化のセットアップ

#### z/OS UNIX リソースのカスタマイズ

z/OS UNIX リソースは、SA z/OS カスタマイズ・ダイアログでそれらのリソースを定義することで SA z/OS に導入されます。

カスタマイズ・ダイアログは、アプリケーション・タイプ USS をサポートしています。USS を選択すると、UNIX ユーザー ID、コマンドまたはパス、ファイル名、またはモニターされるポートなどの z/OS UNIX 固有のデータを入力できます。これらのフィールドの 1 つを選択して、データを入力します。

MVS と z/OS UNIX コマンド間で始動と停止の定義を切り替えることができます。例えば、アプリケーションを停止するには、最初に UNIX 強制終了コマンドを出し、(これが成功しなかった場合は) その後で MVS のキャンセルを実行できます。

### 自動化セットアップの定義

SA z/OS とともに提供されるプログラムを配置する USS パスは、「システム情報 (System Info)」ポリシーの「USS 自動化パス (USS Automation Path)」フィールドで定義する必要があります。

### z/OS UNIX リソースの定義

新規のアプリケーション項目 (APL、クラス、またはインスタンス) を定義するには、「新規項目の定義 (Define New Entry)」パネルにアプリケーション・タイプの USS を指定します。アプリケーション・タイプ USS を選択すると、オプションの「USS 制御 (USS Control)」が「ポリシーの選択 (Policy Selection)」パネルに表示されます。

「ポリシーの選択 (Policy Selection)」パネルで「USS 制御 (USS Control)」を選択して、新規の z/OS UNIX リソースのデータを入力します。1つのクラスに対して、ユーザー ID と z/OS UNIX モニター・ルーチンだけをこのパネルで指定できます。その他の定義 (from/to、依存関係など) はすべて通常どおりに入力できます。このパネルの詳細については、「IBM Z System Automation 自動化ポリシーの定義」を参照してください。

USS アプリケーションは、JES に対して HASPARENT 関係として定義する必要があります。

タイプが INSTANCE のオブジェクトについて、「USS 制御の指定 (USS Control Specification)」ポリシー項目を使用して、リソースを以下のいずれかとして定義します。

#### プロセス

実行中のコマンドのパス (UNIX コマンド `ps -e` で表示される) を「プロセス・コマンド/パス (Process Command/Path)」フィールドに入力します。

#### フィルター

USS プロセスを一意的に識別する追加のフィルター基準を入力します。このフィールドはオプションであり、「コマンド/パス (Command/Path)」が指定されている場合にのみ指定できます。

#### TCP ポート

リソースがローカル・ホスト上で listen する TCP ポート番号を、「ポート番号 (Port Number)」フィールドに入力します。

#### ファイル

zFS 内のファイルのパスを「ファイル名 (File Name)」フィールドに入力します。

#### IP スタック

TCP ポートのモニターに対して、複数の IP スタックが構成されている場合にオプションで IP スタックの名前を入力できます。

Java™ プロセスでは特に、コマンド/パスの指定が固有でないことがよくあります。これが不可能な場合には、「フィルター (Filter)」フィールドで USS プロセスを固有に識別することができます。

AOFUXMON 使用時の USS アプリケーションのモニタリングには、以下の 2つの方式があります。

- 標準的な方式は、APPLICATION INFO ポリシー項目でモニター・ルーチン AOFUXMON を指定することです。AOFUXMON は、UNIX システム・サービスのリソース用に SA z/OS によって呼び出されます。
- zFS 内で独自のスクリプトまたはプログラムを使用することを選択する場合、これは AOFUXMON によって呼び出されます。次に、そのスクリプトまたはプログラムを USS CONTROL ポリシー項目の「モニター・コマンド (Monitoring Command)」フィールドに指定する必要があり、また、APPLICATION INFO ポリシー項目に AOFUXMON を指定する必要もあります。

このプログラムがスラッシュ (/) で始まっていない場合、これは、SA z/OS 提供の z/OS UNIX `ingccmd` ルーチンと同じディレクトリーに入っていないなければなりません。そうでないと、指定されたその名前は絶対パス ID であるとみなされます。

UNIX モニター・ルーチンには、終了値がなければなりません。この値は以下のいずれかです。

#### 0

リソースは使用可能

4

リソースは開始中

8

リソースは使用不可

12

エラーが発生

ユーザー指定モニター・ルーチンがループすると、AOFUSSWAIT 時間 (NetView スタイルシートに定義される) の後で SIGKILL を受信します。

**ヒント:** アクションを起動するか、あるいは「NetView 自動化テーブル (NetView Automation Table)」(AT) で状況変更を行うために、この UNIX モニター・ルーチンから MVS システム・ログにメッセージを書き込むことができます。

モニター・ルーチン AOFUXMON を指定する必要があります。指定しない場合は、デフォルトのモニター・ルーチン (通常 INGPJMON) が呼び出されますが、これは z/OS UNIX リソースには不十分です。

「**ジョブ・タイプ (Job Type)**」フィールドは、MVS または NONMVS のどちらかにできます。

### MVS

固有のジョブ名をもつプロセスを示すリソースでのみ使用されます。これらのリソースでは、SA z/OS は状況変更で次のメッセージを受け入れます。

- IEF403I ジョブが開始しました
- IEF404I ジョブが終了しました
- IEF450I ジョブが異常終了しました

始動コマンドを指定しないと、デフォルトの MVS 始動メソッド (s <JOBNAME>) が使用されます。

### NONMVS

SA z/OS は状況変更について上にリストされたメッセージを無視します。これは、ジョブ名が固有でない場合に必要です。

z/OS UNIX リソースでは、SA z/OS がアプリケーションの始動コマンドを出すと、「始動遅延 (Start Delay)」間隔が開始します。SA z/OS は、モニターされるリソースが開始したことを z/OS から通知されます。この結果、USS リソースは ACTIVE 状況に設定されます。最初の始動遅延間隔と正常モニターの後で、そのエージェント状況を UP に設定する ACTIVMSG コマンドが起動されます。「Start Timeout」のデフォルト値は 2 分です。

APPLICATION INFO ポリシー項目の「**ACTIVE 状況をスキップ (Skip ACTIVE status)**」フィールドを YES に設定すると、プロセスが実行中であることを SA z/OS が z/OS から通知されたとき、リソースが即座に UP に設定されます。

アプリケーションのシャットダウンでは、SA z/OS は、プロセスが終了するとすぐ z/OS から通知を受けます。この時点で、SA z/OS は即時にリソースを AUTODOWN 状況に設定します。

この動作の結果、アプリケーションとして、またはクラス・レベルで以下のパラメーターを APPLICATION INFO ポリシー項目にどのように設定するかを注意深く検討する必要があります。

- 始動遅延 (Start Delay)
- 始動サイクル (Start Cycles)
- ACTIVE 状況をスキップ (Skip ACTIVE status)
- シャットダウン・パス間隔 (Shutdown Pass Interval)
- クリーンアップ遅延 (Cleanup Delay)

詳細については、\*USS ベスト・プラクティス・ポリシーを参照してください。

### 自動化リソース

## プロセス・モニター

UNIX プロセス ID (PID) をモニターすることはできません。モニター・ルーチンには、始動コマンドと、そのプロセスが属しているユーザー ID が必要です。UNIX コマンドの `ps` を実行すると、この情報が得られます。

次の例では、ユーザー `USER` に属しているプロセスがすべて表示されます。

```
USER:/u/user/ingcmd>ps -e
      PID COMMAND
33554481 /bin/sh
50331698 /usr/sbin/rlogind2
33554486 /usr/lpp/netview/bin/cnmeunix
67108927 /bin/sh
83886176 /bin/ps
33554821 /usr/sbin/inetd
83886472 FTPD
67109276 /bin/sh
16777629 /usr/sbin/rlogind2
33554924 HSAPYTCP
```

これは、`/usr/sbin/rlogind2` によって開始された 2 つのプロセス間で、自動化が区別できなかったことを意味します。同一のコマンドで開始されるプロセスには、別々のユーザー ID がなければなりません。

### 代替方法 1

複数のインスタンスを実行するプロセスを自動化する必要がある場合は、異なるプロセスを区別するために、ユーザーはソフト・リンクを使用できます。例えば、次のプロセス

```
/u/user/usstest/testme
```

は、複数回開始されるはずですが、この場合、次のように、いくつかのソフト・リンクを作成します。

```
USER:/u/user/usstest> ln -s testme test1
USER:/u/user/usstest> ln -s testme test2
```

その結果は次のようになります。

```
USER:/u/user/usstest>ls -al
total 216
drwxrwxr-x  2 USER    DE#03243    8192 Jan 24 16:24 .
drwxr-xr-x 19 USER    DE#03243    8192 Jan 24 16:23 ..
lrwxrwxrwx  1 USER    DE#03243     6 Jan 24 16:24 test1 -> testme
lrwxrwxrwx  1 USER    DE#03243     6 Jan 24 16:24 test2 -> testme
-rwxrwxr-x  1 USER    DE#03243   94208 Jan 24 16:23 testme
```

これらの 3 つのプログラム (「実際の」同じプログラム) は、異なる 3 つの始動コマンド `test1`、`test2`、と `testme` で自動化できます。これらのリンクは `prestart` コマンドとして作成し、`shutfinal` コマンドとして削除できます。

**注:** プログラムの始動に使用されたパラメーターではなく、コマンドのみが使用されます。これは、SA z/OS が別の始動パラメーターでプログラムを始動する場合があるため、自動化マネージャーが自動化エージェントに対して通知した実行内容によって異なります。この場合は、定数値のみがコマンドであり、パラメーターではありません。

### 代替方法 2

同じプログラムが、異なる始動パラメーターを使用して、複数回並列実行される場合があります (Java プログラムのように)。この場合には、これらのプロセスを上述のように自動化することは非効率です。Java プログラムは Java 環境で実行され、Java プロセスとして表示されます。例えば、以下のとおりです。

```
# ps -e
      PID TTY          TIME CMD
50331734 ?        5h24  .../V6R1/AP/AppServer/java/bin/java
83886173 ?        1:44  .../V6R1/AP/AppServer/java/bin/java
60341724 ?        2h36  .../V6R1/AP/AppServer/java/bin/java
73392173 ?        1:02  .../V6R1/AP/AppServer/java/bin/java
```

このケースでは、モニターすべきプロセスを区別し評価することは不可能です。

コマンド `ps -ef` は同じプロセスを表示します (例えば、Java 環境で実行しているプログラム) が、完全修飾された Java パスを使用せずに、始動時に使用されるパラメーター・チェーンを使用します。

```
#ps -ef
  UID      PID PPID  C   STIME TTY   TIME CMD
EEZDMN  50331734  1 -   Jun 27 ?   5h25 java -Djava.util.logging.configureByServer=true
EEZDMN  83886173  1 -   Jun 27 ?   1:44 java -Dcom.ibm.eez.adapter.debug=true
EEZDMN  60341724  1 -   Jun 27 ?   2h36 java -Djava.util.logging.manager=connect
EEZDMN  73392173  1 -   Jun 27 ?   1:02 java -Djava.security.auth.login.config=/etc/security.conf
```

一致する PID を使用して両方のコマンドの出力をマッピングすることにより、固有のプロセスを評価しモニターすることができます。区別されるプロセスは以下のようになります。

```
/SYSTEM/local/WebSphere/V6R1/AP/AppServer/java/bin/java
-Djava.util.logging.configureByServer=true
```

ここで、「UNIX 制御の指定 (UNIX Control Specification)」パネルで指定されるデータは、「プロセス・コマンド/パス (Process Command/Path)」フィールドでは、`/SYSTEM/local/WebSphere/V6R1/AP/AppServer/java/bin/java` であり、「フィルターあり (with Filter)」フィールドでは、フィルター - `Djava.util.logging.configureByServer=true` です。

USS プログラムにスティッキー・ビットが設定されている場合、MVS のロードはシンボリック・リンク名を使用して行われます。例えば、`syslogd` の 2 つのインスタンスを実行するには、`/tmp/syslogd` などのシンボリック・リンクを使用する必要があります。同じ名前 (`syslogd`) を作成できるようにするために、個別の `/tmp` ディレクトリーを使用しなければなりません。

### TCP ポート・モニター

1 つのリソースにつき、入力できる TCP ポート番号は 1 つだけです。SA z/OS は、関数 `gethostid()` によって戻される時にローカル・ホストをモニターします。

このポートが「listening」状態の時、SA z/OS には、このリソースは「使用可能」とみなされます。その他の状態のポートはすべて「使用不可」にマップされます。

定義の際にユーザー ID は不要です。

システムで複数の IP スタックが構成されている場合は、定義されたポートの対応する IP スタックの名前を「IP スタック (IP Stack)」フィールドに指定できます。

### ファイルまたはファイル・システムのモニター

ファイル (特定のユーザーに属する) の存在が検査されます。多くのアプリケーションは始動時にファイルを作成し、正常終了時にこれらのファイルを削除します。複数のファイルをモニターする必要がある場合は、自動化マネージャー中のアプリケーション・グループ (APG) としてモデル化できます。

このモニターを使用して、特定のファイル・システムがマウントされているかどうかを判別できます。このリソースの始動コマンドは UNIX の「`mount`」コマンドであり、停止コマンドは UNIX の「`umount`」です。

### 始動と停止の定義 (INGUSS コマンド)

従来の MVS コマンドによってリソースを制御する場合は、他のすべての MVS アプリケーションの場合と同じ方法で行われます。

z/OS UNIX 環境でのコマンドの実行は、リソースの始動または停止の定義時に INGUSS コマンドを指定することによって行われます。

USS 環境でコマンドを出すには、INGUSS コマンドを使用します (詳細については、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照)。

注: INGUSS を使用できるのは、基本 JES が使用可能な場合だけです。したがって、INGUSS を使用する z/OS UNIX リソースでは、JES への HASPARENT の依存関係が必要となります。ほとんどの z/OS UNIX アプリケーションには、この依存関係があります。prestart コマンドを出したい場合は、追加の PREPAVAILABLE 依存関係も必要となります。

z/OS UNIX および MVS のコマンドは、異なるシャットダウン・パスで混合することができます。

### コマンドの例

## プロセスの始動コマンド

カスタマイズ・ダイアログでコマンドとジョブ名を指定してプロセスを開始するには、そのリソースの STARTUP ポリシー項目の「始動コマンドの処理 (Startup Command Processing)」パネルで、「**コマンド・テキスト (Command Text)**」フィールドに `INGUSS JOBNAME=&SUBSJOB &SUBSPATH &SUBSFILTER` と入力します。

アプリケーションまたはモニターの始動で使用されたコマンドのみをモニターできます。同じプログラムを複数回始動する場合は、プロセスを区別するために、`prestart` コマンドとしてソフト・リンクを使用することができます。

同じ実行可能ファイルを `Prestart` コマンドとして使用するプロセスの区別でのソフト・リンクの使用

`&SUBSPATH` のソフト・リンク (コマンドを出すリソースのパス・パラメーター、例えば、`/u/user1/uss1`) と、ファイル `/u/user1/uss1test` へのリンクを作成するには、PRESTART の「コマンド処理 (Command Processing)」パネルの「**コマンド・テキスト (Command Text)**」フィールドに、以下のコマンドを入力します。

```
INGUSS /bin/ln -s /u/user1/uss1test &SUBSPATH
```

zFS は次のようになります。

```
USER1:/u/user1>ls -l
total 408
lrwxrwxrwx  1 USER1      DE#03243      7 Feb 13 12:44 uss1 -> uss1test
-rwxrwxr-x  1 USER1      DE#03243    163840 Jan 29 14:55 uss1test
```

## プロセスの停止コマンド

z/OS UNIX プロセスは、さまざまな方法で停止できます (エスカレーションでの受け渡しの使用)。例えば、まず始めに z/OS UNIX `kill` コマンドを使用し、それが機能しなかった場合に z/OS UNIX `kill -9` を使用し、それも機能しない場合は最後に MVS `cancel` コマンドを入力します。

88 ページの図 22 に示しているように、リソースの通常シャットダウン・フェーズ用に、「コマンド処理 (Command Processing)」パネルで (そのリソースの SHUTDOWN ポリシー項目を介して) この例の定義を入力します。

Cmd	Ps	AutoFn/*	Command Text
1			INGUSS /bin/kill &SUBSPID
3			INGUSS /bin/kill -9 &SUBSPID
4			MVS C &SUBSUSSJOB,A=&SUBSASID

図 22. プロセスの停止定義

`&SUBSPID` は実行時にプロセスの実際の PID で置き換えられます。

## ファイルの停止コマンド

ファイルの停止コマンドでは、ファイルを削除することがあります。カスタマイズ・ダイアログに入力したファイル名は、88 ページの図 23 に示されているように、`&SUBSFILE` にあります。

Cmd	Ps	AutoFn/*	Command Text
1			INGUSS /bin/rm &SUBSFILE

図 23. ファイルの削除

## 例: sshd

セキュア・シェル・デーモン・アプリケーション (SSH デーモンまたは `sshd`) は `ssh` 用のデーモン・プログラムです。このプログラムは `rlogin` および `rsh` の代替機能であり、安全でないネットワーク上の 2 つの非トラステッド・ホスト間における暗号化された通信を可能にします。`sshd` はポート 22 でクライアントからの接続を `listen` するデーモンです。



これは通常 z/OS UNIX の初期化時に開始します。着信接続ごとに新しいプロセスをフォークします。フォークされたプロセス/接続は、鍵交換、暗号化、認証、コマンド実行、およびデータ交換を処理します。これらの接続は、SSH デーモンと同じジョブ名、コマンド/パス、およびフィルターを示します。sshd の開始時、そのプロセス ID (pid) は /var/run/sshd.pid ファイルに書き込まれます。

sshd に対する適応および構成の変更は、sshd 構成ファイル sshd\_config 内で行います。このファイルは /etc/ssh ディレクトリーにあります。

セキュア・シェル・デーモン・アプリケーション (sshd) の高可用性を保つには、sshd をその親プロセスから切り離してはなりません。また、sshd は個別のシェル環境で開始する必要があります。このシェルは、モニター可能な固有のプロセスを確立しなければなりません。これを行うには、オプション -D を使用して sshd を開始します。

シャットダウンのために、プロセス ID ファイル (sshd.pid) をファイル・システムに書き込む必要があります。このプロセス ID は該当ファイルから読み取られ、終了のための sshd の識別に使用されます。

ps -ef コマンドは、フィルターとして参照されるプロセスを識別するためのさらに多くのパラメーターを提供します。次に例を示します。

```
ps -ef | grep ssh
UID      PID      PPID     C STIME TTY          TIME CMD
AUT01    83886553 67109368 - 14:53:27 ?           0:00 -sh -c /usr/sbin/sshd -D
AUT01    83886563 50335037 - 14:53:37 tty0000    0:00 grep ssh
AUT01    83887096 83886553 - 14:53:28 ?           0:00 /usr/sbin/sshd -D
```

プロセス 83886553 は、モニターのための対応する親シェル・プロセスを含むアドレス・スペースを示します。プロセス 83887096 は sshd 自体です。

この出力から、「フィルター (Filter)」を -c '/usr/sbin/sshd -D' と設定します。

注: コマンド ps -ef の出力に引用符が表示されていない場合であっても、「USS 制御 (USS Control)」ポリシーの「フィルター (Filter)」フィールドでは引用符を指定して定義する必要があります。

「コマンド/パス (Command/Path)」の必須情報を確認するには、ps -e コマンドを実行し、親シェルのプロセス ID を見つけます。

```
PID      TTY TIME      COMMAND
...
83886553 ?    14:53:27  /bin/sh
67109368 ?    14:34:28  BPXBATCH
83887096 ?    14:53:28  /usr/sbin/sshd
...
```

または、ps -e | grep 83886553 を実行してプロセスを直接取得します。

```
83886553 ?    14:53:27  /bin/sh
```

これにより、sshd モニター・プロセスのプロセス ID (PID) が表示されます。この出力から、「コマンド/パス (Command/Path)」を /bin/sh と設定します。

次に、プロセスが実行されている z/OS ユーザー ID を検索します。これを行うには、次の z/OS コマンドを発行し、プロセス ID (PID) が表示されている行の最初の列でユーザー ID を検索します。

```
D OMVS,PID=83886553
```

次の出力が表示されます。

```
BPX0070I 19.01.02 DISPLAY OMVS 035
OMVS      000E ACTIVE          OMVS=(PA,F9,11,L0,S0,09)
USER      JOBNAME  ASID      PID      PPID STATE  START  CT_SECS
OMVSKERN  SSHD      00D0    83886553 67109368 1WI----- 14.53.27 .0
LATCHWAITPID=          0 CMD=-sh -c /usr/sbin/sshd -D
THREAD_ID          TCB@    PRI_JOB  USERNAME  ACC_TIME SC  STATE
21ADF00000000001 008FF1C0 OMVS          .006 WAT  W
```

この出力から、「ユーザー ID (User ID)」を OMVSKERN に設定します。

前述のように、sshd がポート 22 で listen することを確認します。netstat -a コマンドを使用し、SSHD ジョブ名を見つけて出力を評価します。

```
# netstat -a
...
SSHHD    00000049 Listen
Local Socket:  0.0.0.0..22
Foreign Socket: 0.0.0.0..0
...
```

この出力から、ポートを 22 に設定します。

タイプが USS のアプリケーション (APL) に対して、「USS 制御 (USS Control)」ポリシー項目のフィールドを使用して、カスタマイズ・ダイアログで [90 ページの表 13](#) に示すようなデータなどを指定し、UNIX インターネット・デーモン (sshd) を定義できます。

表 13. カスタマイズ・ダイアログにおける sshd の定義の例

	プロセス	ファイル	ポート
アプリケーション名*	SSHHD/APL	SSH_FILE/APL	SSH_PORT/APL
ユーザー ID	OMVSKERN		
プロセス・コマンド/パス	/bin/sh		
フィルター	-c '/usr/sbin/sshd -D'		
ファイル名		/var/run/sshd.pid	
PORT			22
IP スタック			TCPIP**

\* これは、アプリケーションの作成時に指定された名前です。 \*\* システムで複数の IP スタックが構成されている場合のみ。

次のことを示す関係のあるすべてのリソースを含む基本グループを定義します。

- すべての sshd 関連リソースを含むグループが TCPIP に依存する
- ファイルが sshd プロセスによって作成され、SA z/OS によって直接開始または作成できない
- ポートで listen する sshd プロセスが SA z/OS によって直接開始または作成できない

[91 ページの図 24](#) では、ユーザー OMVSKERN によって開始されたプロセス /bin/sh -c '/usr/sbin/sshd -D' が示され、ファイル /var/run/sshd.pid が存在し、ポート 22 が「listen」の状況 (sshd はこのポートで着信ログイン要求を listen する) である場合に稼働中の SSHD (グループとしてモデル化) を示しています。/etc/ssh/sshd\_config に定義されたポートのみが選択可能です。



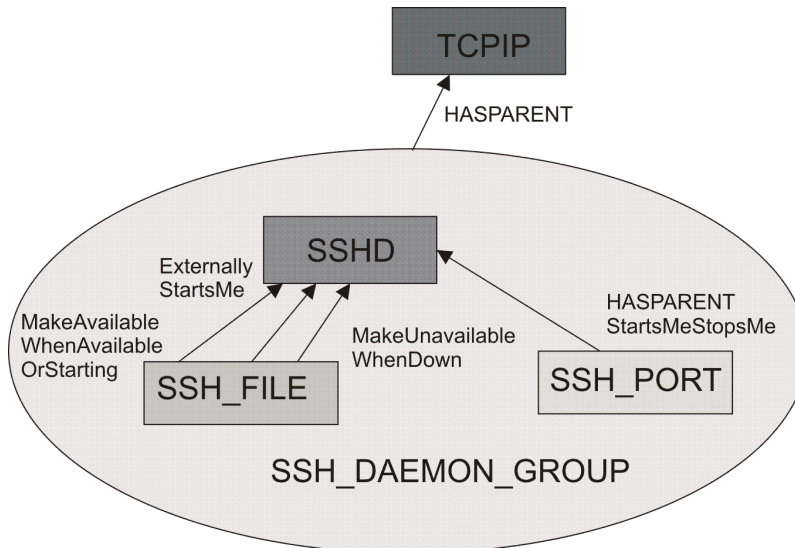


図 24. SSH デーモンの listen

**SSHD\_FILE/APL の開始定義**

なし。

**SSHD\_PORT/APL の開始定義**

なし。

**SSHD/APL の開始定義**

CMD: INGUSS JOBNAME=&SUBSJOB &SUBSPATH &SUBSFILTER (&SUBSxxx 変数は実行時に置換されます)

**SSHD\_FILE/APL の停止定義**

これらのコマンドは、ファイルが sshd プロセスによってまだ除去されていない場合にそのファイルを除去します。

表 14. sshd のパスに関する説明		
パス数	パスの説明	コマンド
1	sshd によってまだ実行されていない場合	INGUSS /bin/rm &SUBSFILE
4	パス 1 でファイルが除去されなかった場合	INGUSS /bin/rm -f &SUBSFILE

**SSHD/APL の停止定義**

SSHD を停止する場合は、sshd.pid ファイルから正しいプロセス ID を抽出する必要があります。sshd プロセス自体を終了するには、kill コマンドを使用します。sshd コマンドが発行されると、モニター目的で使用される親シェルが終了し、SSHD/APL が AUTODOWN 状況になります。

以下のコマンドを使用してください。

```
CMD:INGUSS /bin/sh -c '/bin/ps -ef | /bin/grep `'/bin/cat /var/run/sshd.pid`
| /bin/grep -q sshd && /bin/kill `'/bin/cat /var/run/sshd.pid`'
```

```
CMD:INGUSS /bin/sh -c '/bin/ps -ef | /bin/grep `'/bin/cat /var/run/sshd.pid`
| /bin/grep -q sshd && /bin/kill -9 `'/bin/cat /var/run/sshd.pid`'
```

```
CMD:INGRCLUP &SUBSJOB
```

ここでは、上記のコマンドと USS パイプについて説明します。

表 15. INGUSS コマンドおよび USS パイプの要約	
コマンド	タスク
INGUSS	USS へのインターフェース
/bin/sh -c '...'	sshd の PID を抽出するための完全なシェル環境を提供します
`/bin/cat /var/run/sshd.pid`	サブシェルが sshd の PID を抽出します。抽出 PID は左の列に 'extracted_pid' で示されています
/bin/ps -ef	すべての USS プロセスを要求します
/bin/grep extracted_pid	sshd プロセスのみを表示します
/bin/grep -q sshd	出力を抑止し、該当する戻りコードを返します: 戻りコード 0: sshd プロセスが存在します。戻りコード 1: sshd プロセスが存在しません。
&&	戻りコードが 0 の場合 (sshd が存在する場合) にのみ続行します
/bin/kill extracted_pid または /bin/kill -9 extracted_pid	sshd の 'extracted pid' に対して SIGTERM シグナルを出します

kill コマンドで sshd プロセスが終了しない場合は、INGRCLUP ルーチンを使用して sshd およびモニター対象の親シェルのアドレス・スペースに対して z/OS CANCEL コマンドを呼び出します。

SA z/OS では、sshd などのいくつかの自動化 USS デモンに対する定義を提供する \*USS ベスト・プラクティス・ポリシーが用意されています。USS リソースの共通定義は、C\_USS\_xxx で始まる APL クラスにあります。

使用する USS コマンドとそのパラメーターの詳細については、UNIX のマニュアル・ページも参照してください。

## ヒント

USS タスクが STC として始動するときの動作は、USS 環境内で直接始動するときとは異なります。

USS タスクが STC として始動するときは、開始ユーザー ID が異なることがあるので、たいていの場合、AOFUXMON モニター・ルーチンは ACTIVMSG UP=YES を内部的にトリガーできません。

この場合、SA z/OS が INGUSS を使用してこれらのアプリケーションを始動する際に、非常にやりやすくなります。UP メッセージで AT 項目が不要になります。SA z/OS は内部的にこれをシミュレートできるので、UP メッセージに関しての心配は不要です。

ジョブ名 (すなわち、ジョブ名の最終文字) は、USS リソースには予測可能ではありません。しかし、AOFUXMON は、USS 内のパスをモニターし、SA z/OS 内に定義されたジョブ名を変更することによってこれを処理することができます。

syslog デモンの場合は、SYSLOGD としてジョブ名を定義することになります。アプリケーションが始動されて、例えば SYSLOGD7 にジョブ名が変更されると、AOFUXMON はこれを反映するため SA z/OS データ・モデルを調整します。しかし、これは SYSLOGD\* の汎用項目がある AT 内では処理できません。ジョブ名の変更は、新しい名前の新規アドレス・スペースを作成する USS プロセスが原因となって行われるためです。したがって、古い名前の旧アドレス・スペースは終了します。これは、旧アドレス・スペースの終了メッセージと新規アドレス・スペースの UP メッセージを受け取ることを意味します。また、これらのメッセージの順序は予測不能です。

## UNIX syslogd メッセージのトラップ

UNIX syslogd メッセージをトラップするには、そのメッセージを MVS システム・ログに転送するために syslogd 構成ファイル /etc/syslog.conf に項目を追加する必要があります。このようにして、NetView 自動化テーブル (AT) によってメッセージを処理することができます。

すべてのメッセージを MVS ログに転送するには、次の項目を追加します。

```
 *.* /dev/console
```

特別なメッセージを MVS ログのみに送信するには、syslog メッセージの命名ガイドラインに従ってください (例えば、警告メッセージでは、\*.warn を使用する)。/dev/console は書き込み先の通常ファイルとして使用できます。

これをテストするには、以下の UNIX コマンドを USS コンソールから発行します。

```
echo 'This is a test message' >>/dev/console
```

UNIX メッセージには MVS メッセージ ID BPXF024I があり、これは複数行のメッセージです。

93 ページの図 25 に、システム・ログへの UNIX コマンドの出力例を示します。

```
M 13:45:21.34 STC03602 00000090 BPXF024I (USER) Feb 13 13:45:21 SYS1 syslogtest 67109100 : This
is
S
498
D
message          498 00000090 a test
```

図 25. UNIX のメッセージの例

## UNIX \_\_console () サービスで発行されるメッセージのトラップ

アプリケーションが \_\_console () (BPX1CCS/BPX4CCS) サービスを呼び出してメッセージを発行していて、サービスを呼び出すプロセスが UID =0 で実行されていない場合は、メッセージ BPXM023I がオペレーターに対して出されます。BPXM023I は、実際のメッセージを囲むラッパーです。単一行メッセージまたは複数行メッセージのいずれかです。

このメッセージを自動化しやすくするために、INGMSGSA 自動化テーブル・ステートメントは、1つの単一フォーマットのみを持つようにこのメッセージを再フォーマット設定します。この自動化テーブル・ステートメントは、BPXM023I が複数行メッセージの場合はメッセージの最初の行を除去し、BPXM023I が単一行メッセージの場合は接頭部を除去します。そのため、呼び出し元の特権にかかわらず、このメッセージに対して信頼性の高い自動化を行うことができます。

SA が BPXM023I を再フォーマットする方法は、INGMSGSA 自動化テーブルに以下のように記述されています。

```
* BPXM023I message
* This msgID is just a wrapper that is only added if the
* the process calling BPX1CCS is not running with UID=0.
* It is either a single or a multi line message.
* SA modifies the message to enable automation independent from the
* issuers authority.
*
* -- WTO request via BPX1CCS syscall - Multiline message --
* Modify message - drop first line of the multiline message
  IF MSGID = 'BPXM023I' & HDRMTYPE = '' & TOKEN(3) = ''
    THEN EDIT('FWDLINE 1 1.* 1 WRITELINE COPYREST');
*
* -- WTO request via BPX1CCS syscall - Singleline message --
* Modify message - remove header from first line
  IF MSGID = 'BPXM023I' & HDRMTYPE = 'E'
    THEN EDIT('WORD 3.* 1 WRITELINE COPYREST');
```

例えば、次の単一行 BPXM023I メッセージを受け取るとします。このメッセージは、IBM Common Data Provider for z Systems の UP メッセージです。

```
BPXM023I (CDPU) HB06001I The Data Streamer started successfully.
```

INGMSGSA 自動化テーブル・ステートメントは、最初の 2 ワードを削除して、次のメッセージを生成します。

## ヒント

```
HBO6001I The Data Streamer started successfully.
```

そのメッセージを自動化したい場合は、SA ポリシーで「実際の」メッセージ ID HBO6001I を UP メッセージとして定義します。

## デバッグ

デバッグは、AOCTRACE パネル上で z/OS UNIX モニターとコマンド実行用に活動化できます。モニター用の自動化プロシージャーは AOFUXMON であり、コマンド実行用は AOFRSUSS です。

AOFRSUSS のデバッグを暗黙にオンにすると、ingccmd (SA z/OS コマンド・サーバー) のデバッグがオンになります。

デバッグ・メッセージはネットログと z/OS UNIX システム・ログ (syslogd) に書き込まれます。

## 第9章 コマンド受信側

ここでは、汎用のコマンド受信側のセットアップ方法および使用方法について説明します。コマンド受信側は、NetView、SA z/OS、または MVS のコマンドを SA z/OS に渡して実行できます。また、コマンド受信側は、受信コマンドをコマンド作業タスクと呼ばれる、一連の専用タスクにディスパッチすることができます。コマンド受信側は、ラウンドロビン・アルゴリズムを使用して、関連するそれぞれの自動化機能に対して、処理するコマンドを同じ数だけ均等に与えます。すなわち、この手法により、複数の受信コマンドを並列に実行できます。

コマンド受信側を使用する機能は次のとおりです。

- バッチ・コマンド・インターフェース
- カスタマイズ・ダイアログの AT 上書き構文検査
- TSO からのリレーショナル・データ・サービス
- TSO で実行される SA z/OS 提供のその他の REXX 関数

### コマンド受信側のセットアップ

コマンド受信側のセットアップは、以下の項目から構成されます。

1. TSO/バッチ環境をセットアップする。110 ページの『TSO 用 RDS の使用』のインストール手順を参照してください。
2. SA z/OS によって自動化されるサブシステムとしてコマンド受信側を定義する。
3. コマンド受信側によって使用されるコマンド作業タスクを定義する。

アドオン・ポリシー \*IBMCOMP で定義されている SA リソースを使用することをお勧めします。以下のエントリーを選択します。

ポリシー・エントリー名	ポリシー・エントリー・タイプ
CMDRCVR	APL
CMD_RECEIVER	APG
CMD_RECEIVER_AUTOOPS	AOP

これらの SA z/OS リソースにより、コマンド受信側のタスクが SA z/OS の初期化時に開始し、コマンドを受信できる状態になります。オペレーターは INGLIST コマンドを使用してコマンド受信側の状況をモニターできます。

### TSO/バッチ環境のセットアップ

適切な TSO モジュールに REXX 関数パッケージ INGTXFPG を追加します。TSO/E は独自のロード・モジュールをコーディングするために使用できる以下のサンプルを SYS1.SAMPLIB に用意しています。

サンプル名	ロード・モジュール名
IRXREXX1	IRXPARM (MVS 用)
IRXREXX2	IRXTSPRM (TSO/E 用)
IRXREXX3	IRXISPRM (ISPF 用)

独自のパラメーター・モジュールを提供するにはさまざまな考慮事項があります。詳しくは、「TSO/E REXX 解説書」の章『言語プロセッサ環境』にあるセクション『関数パッケージ・テーブル』を参照してくだ

さい。次の項目のために初期化される環境に対してのみパラメーター値を変更するのかどうかによって、考慮事項が異なります。

- ISPF
- TSO/E セッションと ISPF セッションの両方
- TSO/E 以外のアドレス・スペース

適切なサンプル・パラメーター・モジュール (TSO/E の IRXREXX2、バッチ PGM=IKJEFT01 など) を選択し、システム関数パッケージとして関数パッケージ INGTXFPG を追加します。

以下の点を考慮する必要があります。

1. REXX デフォルト・パラメーター・モジュールを対応する名前にリンク・エディットする必要があります。例えば、サンプル IRXREXX2 のロードの名前は IRXTSPRM でなければなりません。
2. 結果の REXX デフォルト・パラメーター・モジュールを LPALST に置きます。
3. 関数パッケージ INGTXFPG も Linklist にあることを確認します。

## SA z/OS によって自動化されるサブシステムとしてのコマンド受信側の定義

サンプル・ポリシー \*IBMCOMP から CMDRCVR オブジェクトと CMD\_RECEIVER オブジェクトをインポートします。これにより、コマンド受信側を示す CMDRCVR という非 MVS サブシステムを含むアプリケーション・グループが作成されます。コマンド受信側のタスクは、VOST で実行されます。VOST の名前は関連リソースの名前です。この場合は CMDRCVR です。コマンド受信側 APL に関連付けられている作業オペレーター AOFCMDnn が、コマンド受信側を実行するための VOST の所有者として使用されます。

CMDRCVR が開始すると、次のコマンドによって PPI 受信側が開始します。

```
INGRCRCV START OPF=AOFCMDOPER
```

デフォルトの PPI 受信側 ID は INGRRCV ですが、これは他の名前に変更できます。変更したい場合は、パラメーター PPI と CSAKEY=EMULATOR を指定する必要があります。パラメーター CSAKEY=EMULATOR は、バッチ・コマンド・インターフェースで使用されるキーを定義し、コマンド受信側と通信するために INGRRCV がデフォルト・キーとして使用します。EMULATOR は、サポートされる唯一のキーです。INGRCRCV コマンドは次のようになります。xxxxxxx は選択した PPI 名です。

```
INGRCRCV START PPI=xxxxxxx CSAKEY=EMULATOR OPF=AOFCMDOPER
```

INGRCRCV で使用するために、複数のコマンド受信先を開始できます。追加のコマンド受信側にはそれぞれ固有の PPI ID が必要ですが、CSAKEY=EMULATOR を指定してはなりません。INGRCRCV と特定のコマンド受信側を相関させる方法について詳しくは、[102 ページの『関数 INGRRCV』](#)を参照してください。複数のコマンド受信側について詳しくは、[97 ページの『複数のコマンド受信側』](#)を参照してください。

OPF=AOFCMDOPER パラメーターが存在しない場合は、コマンド受信側が受信コマンドをいずれかのコマンド作業タスクに送ることができないので注意してください。この場合は、すべての受信コマンドが、コマンド受信側をホスティングする VOST タスクによって順次処理されます。

サブシステム CMDRCVR が終了すると、INGRCRCV STOP コマンドによってコマンド受信側が停止します。

サンプル定義では、停止プロセスのためにエスカレーション・コマンドが定義されています。これは停止パスに INGRCRCV DETACH コマンドと INGRCRCV TASK コマンドを使用します。これらのコマンドは、コマンド受信側を正常に終了するのではなく、コマンド受信側をホスティングする VOST を直接停止します。

## コマンド受信側によって使用されるコマンド作業タスクの定義

サンプル・ポリシー \*IBMCOMP から CMD\_RECEIVER\_AUTOOPS オブジェクトをインポートします。このオブジェクトのインポートにより、AOFCMDnn という一連の自動化機能が定義されます。この nn は 1 から上に開始します。デフォルトでは、5 つのコマンド作業タスクが定義されます。必要に応じて、さらにコマンド作業タスクを追加することができます。

注：



1. コマンド受信側は、自動化機能 (コマンド作業タスク) が定義されてアクティブであるかどうかを検査します。そうでない場合は、エラー・メッセージ ING347I が表示されてコマンド受信側が終了します。
2. NetView 自動化オペレーターの名前は、メンバー ING.SINGPRM(AOFOPFSO) で定義する名前に一致していなければならないことに注意してください。このサンプルは、必要に応じてカスタマイズすることができます。デフォルトでは、以下のマップが使用されます。

表 18. SA と NetView の機能およびオペレーター	
SA 自動化機能	NetView 自動化オペレーター
AOFCMD01   AOFCMDnn	AUTCMD01   AUTCMDnn

## コマンド受信側の開始および停止

コマンド受信側は、SA z/OS によって制御されます。これは非 MVS サブシステムとして定義されます。変更しない限り、コマンド受信側サブシステムの名前は CMDRCVR です。

コマンド受信側を開始するには、次のように適切なサブシステムに対して INGREQ REQ=START コマンドを実行します。

```
INGREQ CMDRCVR/APL/SYS1/ REQ=START
```

通常的环境下、SA z/OS エージェントが自動化マネージャーに登録されるとコマンド受信側が自動的に開始するため、ユーザーがコマンド受信側を開始する必要はありません。

コマンド受信側を停止するには、次のように適切なサブシステムに対して INGREQ REQ=STOP コマンドを実行します。

```
INGREQ CMDRCVR/APL/SYS1 REQ=STOP
```

コマンド受信側によって使用される VOST のリストを表示するには、INGRCRCV QUERY コマンドを使用します。

## 複数のコマンド受信側

複数のコマンド受信側を開始した場合は、INGRCRCV QUERY コマンドを使用して、コマンド受信側のリスト、および関連する NetView タスクと PPI ID を表示できます。以下に例を示します。

```
ING337I  PPI RECEIVER STATUS LIST
STATUS   PPI      TASK      VOST      VOSTOWNER  OPF
-----  -
ACTIVE   INGRRCV  DSI#0022  CMDRCVR   AUTWRK03   AOFCMDOPER
ACTIVE   INGRCVR2 MYTASK     EVTOPER
```

このコマンドの表示には、NetView タスクと PPI ID、VOST 所有者、およびコマンド受信側の開始コマンドで指定された OPF が示されます。

VOST は、コマンド受信側が INGVSTRT コマンドで開始された場合に、そのコマンド受信側を実行するために使用されます (サンプル PDB にある APL の開始コマンドで確認できます)。この場合、VOST は APL サブシステム名であり、VOST 所有者は関連する作業オペレーター AUTWRKnn です。

2 行目は、コマンドによって PPI ID INGRCVR2 で開始された別のコマンド受信側を示しています。

```
AOFEXCMD MYTASK INGRRCV START PPI=INGRCVR2 OPF=EVTOPER
```

この 2 番目のコマンド受信側は VOST では実行されませんが、タスク MYTASK で実行されます。コマンドは、単一の自動化オペレーター機能 EVTOPER によって実行されるようにスケジュールされます。

### VOST と OST

最初のコマンド受信側は VOST で実行され、2 番目のコマンド受信側は OST で実行されます。

VOST の利点は、別個の OST を必要としないことです。ただし、複数のコマンド受信側が実行されている場合、同じ VOST 所有者を使用することは推奨されません。そのため、別個のタスクで AOFEXCMD コマンドを使用してコマンド受信側を開始すると役立つ場合があります。

## バッチ・ジョブからの NetView コマンドの実行依頼

このセクションでは、バッチ・ジョブから NetView コマンドを実行する方法を説明します。これは特に Tivoli Workload Scheduler に対して有効ですが、単体でも使用できます。

### サンプル・バッチ・ジョブ JCL

サンプルのバッチ・ジョブが、Z System Automation インストール済み環境のライブラリー SINGSAMP にあります。メンバー EVJSJ001 に、サンプルの JCL が含まれています。

このバッチ・ジョブは、バッチ・ジョブが指定するコマンド受信側が含まれた SA z/OS エージェントと同じシステム上で実行する必要があります。ほとんどの場合、すべての SA z/OS エージェントで 1 つのコマンド受信側が実行されています。複数のコマンドの実行を同時に処理できるため、汎用のコマンド受信側の使用をお勧めします。汎用のコマンド受信側の使用は、SERVER=\* パラメーターによってトリガーされます。ただし、コマンド受信側をカスタマイズすると、コマンド受信側の名前、さらにはコマンド受信側の数および構成も変更される可能性があります。システム・プログラマーに問い合わせ、これらのバッチ・ジョブに使用する正しいシステムおよびコマンド受信側を決定してください。

### コマンド・ステートメントの構文

このバッチ・ジョブに定義されている、//SYSIN DD 名内のコマンドには、以下の構文規則があります。



1. ブランク行はすべて無視されます。
2. アスタリスク (\*) で始まるすべての行はコメント行であり、出力には表示されますが、それ以外は無視されます。
3. コマンドの後にコメントすることは許可されません。
4. 継続行の間にコメントすることは許可されません。
5. ダッシュ (-) を行に追加して 1 つのコマンドを継続させることができます。
6. コマンドは通常 //SYSTSPRT に出力されます。
7. コマンドの出力を他の DD 名にリダイレクトすることもできます。リダイレクトするには、デフォルトで右不等号括弧 (>) 記号を使用します。
8. PIPE > ステージは禁止されています。代わりに PIPE QSAM を使用してください。
9. フルスクリーン・コマンドは許可されません。

### 有効なコマンド・タイプ

関連する行メッセージを発行する任意のコマンド、CLIST、または REXX プログラムを使用することができます。

つまり、OUTMODE=LINE をサポートするほとんどすべての NetView コマンドおよび SA z/OS コマンドと、コンソールに対して、メッセージの SAY または PIPE を実行する、任意の CLIST または REXX プログラムを指します。

コマンドからの戻りコードを使用して、残りのコマンドの実行を中止することもできます。AOFRCMD プロシージャ定義の 101 ページの『MAXRC』を参照してください。

### コマンドの継続

次の例のように、コマンドの後にダッシュを追加することによって、行をまたいでコマンドを継続させることができます。



```
PIPE NETVIEW LIST STATUS=OPS | -
CONSOLE ONLY
```

### コマンド出力のリダイレクト

通常、コマンドは //SYSTSPRT DD 名に出力されます。しかし、コマンドの出力を他の DD 名にリダイレクトすることもできます。実行するには、リダイレクト記号を使用します。デフォルトは > 記号です。次に例を示します。

```
PIPE NETVIEW LIST STATUS=OPS | CONSOLE ONLY >MYOUTPUT
```

これによって、コマンドの出力を、当バッチ・ジョブまたは他のバッチ・ジョブの後続のステップで、それぞれの固有の目的のために使用することができます。

コマンド接頭部として > を使用している場合などは、AOFRYCMD の REDIRECT パラメーターを使用して、リダイレクト記号を変更することができます。リダイレクト記号は、コマンドに出現する文字と同じであってはならないことに注意してください。詳しくは、99 ページの『AOFRYCMD の説明』を参照してください。

出力 DD 名の DCB 特性は、次のようにしてください。

```
LRECL=132,RECFM=FB
```

### コマンド情報の取得

コマンド AOFRYCMD (または非推奨の EVJRYCMD) を使用してスケジュール化された REXX プログラムにより、タスク・グローバルを介して、以下のように発行元のジョブ名およびユーザー ID を取得できます。

```
jobname = TGLOBAL('AOFRYCMD_JOBNAME')
userid = TGLOBAL('AOFRYCMD_USERID')
```

注: SERVER=\* をコーディングして汎用のコマンド受信側を使用する場合、TGLOBAL AOFRYCMD\_USERID には、AOFRYCMD の userid() 関数により取り込まれる TSO ユーザー ID が入ります。

汎用のコマンド受信側を使用しない場合、TGLOBAL には、コマンド・スレーブ EVJRVCM 内の PIPE EDIT SESSID ステージによって取り込まれる、PPI 送信側の SAF ID が入ります。

## 別の NetView に対するコマンドの実行

ほぼすべての SA z/OS コマンドに、ターゲット・システムでコマンドを強制的に実行する TARGET= パラメーターを指定できます。

コマンドにこの機能がない場合 (NetView LIST コマンドなど) は、次の例に示すように PIPE ラベルを使用すると、適切な NetView にコマンドを送信できます。

```
PIPE CC dom01: LIST STATUS | CONSOLE ONLY
```

また、以下のようにもできます。

```
PIPE CC dom01/auto1: LIST STATUS=OPS | CONSOLE ONLY
```

## バッチ・コマンド・インターフェースの JCL

SA z/OS 製品サンプル・ライブラリー (SINGSAMP) には、SA z/OS/NetView エージェントでコマンドを実行するためのサンプル JCL EVJSJ001 があります。

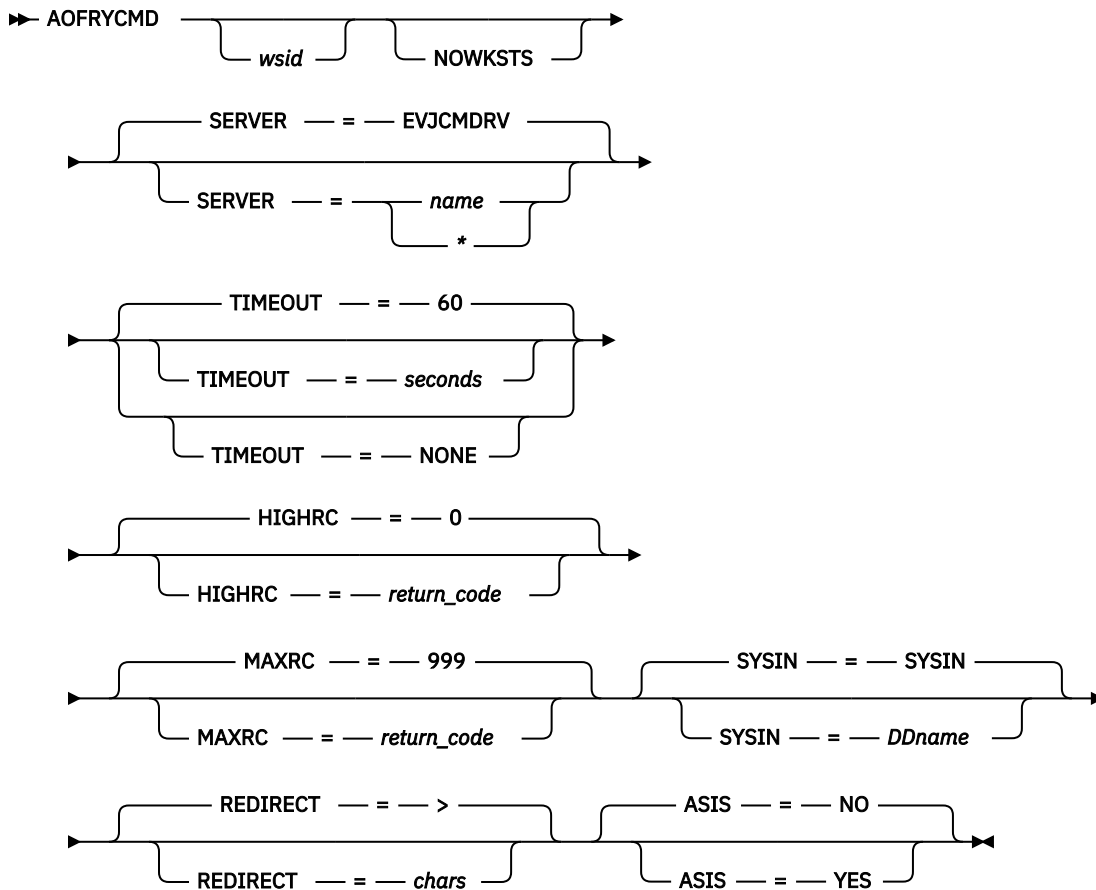
### AOFRYCMD の説明

#### 目的

AOFRYCMD は SA z/OS エージェントに対してコマンドを発行し、そのコマンドの結果を受け取る REXX プロシージャです。

注：AOFRYCMD は EVJRYCMD と同等です。マイグレーションを容易に行うために EVJRYCMD が SINGREXX ライブラリーに引き続き存在しており、一方で AOFRYCMD は SINGTRES に常駐しています。AOFRYCMD を使用することをお勧めします。

構文



パラメーター

**wsid**

このパラメーターはオプションです。

このパラメーターは、バッチ・ジョブを実行依頼した TWS ワークステーションの名前を指定します。コマンドは、この情報を使用して、バッチ・ジョブと SA z/OS エージェントの間の通信が確立できない場合に、そのワークステーションを無効にします。

このパラメーターが指定されている場合、NetView PPI 通信に問題が発生すると、そのコマンドは TWS WSSTAT コマンドを実行してワークステーションをオフライン状態にします。

**NOWKSTS**

このパラメーターはオプションです。非推奨でもあります。ワークステーション ID を省略することをお勧めします。

このパラメーターは、コマンドの動作を変更します。このパラメーターを指定すると、SA z/OS エージェントとの通信に障害が発生した場合に、wsid パラメーターで定義されている TWS ワークステーションがコマンドによって無効になりません。

**SERVER**

このパラメーターはオプションです。

このパラメーターのデフォルトは EVJCMDRV です。このパラメーターは、コマンドの送信先である SA z/OS エージェントの NetView の PPI 受信側の名前を指定します。SERVER=\* と指定すると、コマ

ンド受信側がコマンドの並列処理を有効にするために、コマンドを関連するいずれかの作業タスクに渡します。コマンド受信側の EVJCMDRV は非推奨ですが、互換性のために引き続き存在しています。

EVJCMDRV の定義について詳しくは、\*IBMCOMP アドオン・ポリシーも参照してください。

注：SERVER=\* をコーディングして汎用のコマンド受信側を使用する場合は、SINGTREX ライブラリーを SYSPROC 連結チェーンに追加する必要があります。

#### TIMEOUT

このパラメーターはオプションです。

このパラメーターのデフォルトは 60 秒です。

このパラメーターは、SA z/OS エージェントの NetView でコマンドが実行されるまでバッチ・ジョブが待機する時間 (秒) を指定します。このタイムアウトは各コマンドに個別に適用されます。このタイムアウトを NONE に設定すると、タイムアウトはバッチ・ジョブに適用されません。

注：(FDBK パラメーターで定義される) INGREQ/INGMOVE/INGGROUP のタイムアウトは、ジョブに対する TIMEOUT= パラメーターより小さくすることをお勧めします。

この理由は、INGREQ/INGMOVE/INGGROUP コマンドの FDBK パラメーターを使用して待機期間を指定すると、要求された状況変更が完了するまで、コマンドが待機することになるためです。例えば、TIMEOUT パラメーターがデフォルトの 60 秒である場合は、INGREQ FDBK パラメーターを、FDBK=(WAIT, :55) のようにコーディングしてください。

#### HIGHRC

このパラメーターはオプションです。

このパラメーターのデフォルトは 0 (ゼロ) です。

このパラメーターは、ジョブに対して許容する戻りコードの最大値を指定します。コマンドの戻りコードがこの値以下の場合、JCL ステップの戻りコードはゼロにリセットされます。コマンドの戻りコードがこの値より大きい場合は、そのままのコードが JCL ステップの戻りコードとして渡されます。

注：すべてのコマンドの戻りコードのうち、最大戻りコードが、JCL ステップの戻りコードとなります。

#### MAXRC

このパラメーターはオプションです。

このパラメーターのデフォルトは 999 です。

このパラメーターは、バッチ・ジョブによって実行されるコマンドの戻りコードの最大許容数を指定します。コマンドの戻りコードが、指定された値よりも大きい場合、そのバッチ・ジョブは中止され、残りのコマンドは実行されません。

JCL に報告される戻りコードは、HIGHRC パラメーターによって決定されます。

#### SYSIN

このパラメーターはオプションです。

このパラメーターのデフォルトは SYSIN です。

このパラメーターは、実行されるコマンドが含まれている入力ファイルの DD 名を設定します。

#### REDIRECT

このパラメーターはオプションです。

このパラメーターのデフォルトは > です。

このパラメーターは、リダイレクト文字を定義します。このストリングに等号などの特殊文字が含まれる場合は、引用符または二重引用符で囲んでください。これは、コマンドに出現する文字と同じではありません。

#### ASIS

このパラメーターはオプションです。

このパラメーターのデフォルトは NO です。

## TSO から SA NetView Agent へのリモート・プロシージャ呼び出し

このパラメーターは、パラメーター値が YES の場合に、大文字と小文字が混合する必須コマンドの実行依頼を指定されたとおりにそのまま実行します。

### 使用法

SA z/OS エージェントは、開始時に WSTAT コマンドを自動的に実行して、ワークステーションをオンラインとマークします。エージェントの再始動時にどのワークステーションにオンラインのマークを付けるかについては、トラッカーまたはコントローラーに対する WORKSTATION メッセージ/ユーザー・データのポリシー内に指定します。複数のワークステーションを定義できます。トラッカーに割り当てられるワークステーションの場合、その WORKSTATION ポリシーを、割り当てられるトラッカー自身に対して定義する必要があります。

各コマンドが順に実行依頼され、それらのコマンド結果が取得されます。その後、これらの結果は SYSTSPRT または出力リダイレクト DD 名のどちらかに書き込まれます。

コマンドが正常に完了したかどうかについては、正常に完了した場合はメッセージ ING330I、コマンドが失敗した場合はメッセージ ING332I に示されます。メッセージ ING332I には、コマンドの完了の戻りコードが表示されます。これらのメッセージは SYSPRINT (//SYSTSPRT DD 名) に書き込まれます。

注：NetView PIPE コマンドを MOE オプションと一緒に使用したときに要求されたか、コマンドが正常に完了したときに SA によって自動的に挿入された可能性のある DW0369I メッセージはもう渡されません。

## TSO REXX プログラムから NetView コマンドを実行する

このセクションでは、TSO REXX プログラムから NetView コマンドを実行する方法、および REXX 関数 INGRCRPC の詳細について説明します。

### TSO から SA NetView Agent へのリモート・プロシージャ呼び出し

TSO 下で実行される REXX プログラムは、関数 INGRCRPC を使用して、NetView コマンドまたは MVS コマンドをローカル SA NetView Agent に送信し、コマンドの実行を待つことができます。コマンドの応答は REXX ステムに返されます。

TSO アドレス・スペースと SA NetView Agent は、同じ z/OS システム上になければなりません。INGRCRPC 機能を使用する前に、例えば次のコマンドを使用して、コマンド受信側を SA NetView Agent で開始する必要があります。

```
INGRCRCV START PPI=ppi_rcv_id OPF=AOFCDOPER
```

このコマンドで、PPI パラメーターは PPI 受信側 ID を指定します。これは、INGRCRPC の ppi\_rcv\_id パラメーターと等しくなければなりません。SA コマンド受信側は SA NetView Agent で実行され、PPI 受信側 ID を使用して着信リモート・プロシージャ・コール (RPC) 要求を受信します。INGRCRPC によって送信された RPC 要求は、コマンド実行のために NetView タスクでスケジュール設定されます。OPF パラメーターは、SA オペレーター機能の名前を指定します。この機能は、単一のタスクまたはタスクのグループ (AOFCDOPER など) のいずれかです。コマンド受信側の詳しい説明については、[95 ページの『第 9 章 コマンド受信側』](#)を参照してください。

注：

複数のコマンド受信側を開始する場合は、コマンド受信側ごとに、PPI 受信側 ID と対応する SA オペレーター機能の固有のペアが必要です。

INGRCRPC による複数の要求の同時送信は、同じ TSO アドレス・スペースからはサポートされていません。要求は 1 つずつ送信する必要があります。

### 関数 INGRCRPC

REXX 関数 INGRCRPC は、TSO アドレス・スペースから同じ z/OS システム上の SA NetView Agent アドレス・スペースへのリモート・プロシージャ・コール (RPC) を提供します。目的は、標準コマンド (NetView コマンドや MVS コマンドなど) または自己作成 REXX プログラムを実行し、出力を呼び出し側 TSO プログラムに送り返すことです。

INGRCRCP は、指定された PPI 受信側 ID を使用して、ターゲット・コマンド受信側を判別します。INGRCRCP は内部的に固有の PPI ID を使用して応答を受信します。この PPI ID は、呼び出し側ジョブのジョブ ID です。

## 構文

```
rc = INGRCP(command, input_data, ppi_rcv_id, taskname, resp_name, timeout, security_ctx)
```

## パラメーター

### command

SA NetView Agent で実行されるコマンド。

コマンドの最大長は 31990 文字です。コマンドに 'FF'X または 'FE'X を含めることはできません。MVS コマンドを実行する場合は、「MVS」をコマンドの前に付けます。NetView コマンドまたは SA コマンドには接頭部は不要です。MVS コマンドの場合、INGRCRCP は最初の出力行 (複数行の場合もあります) がキャッチされるまで待機し、タイムアウトを超えることはありません。

### input\_data

オプションの入力データ。REXX プログラムに使用されます。標準コマンドを実行する場合は省略します。

REXX システムの名前 ('input' など) を指定できます。これは、コマンド実行用の NetView デフォルト・システムとして提供されるデータ行の配列です。この名前の末尾はドットでなければならず、input.0 は入力行数を指定する正数でなければなりません。input.n (n=1,2,...) には n 番目の入力行が含まれます。以下に例を示します。

```
input.0=2
input.1='abc'
input.2='xyz'
rc = INGRCP('MYREXPGM','input.')
```

この REXX プログラム MYREXPGM は、NETV デフォルト SAFE を使用して入力データを読み取ることができます。

```
'PIPE SAFE * | STEM input.'
```

### ppi\_rcv\_id

オプションのターゲット PPI ID。

コマンド受信側の PPI 受信側 ID です。指定しない場合は、デフォルト INGRRCV が使用されます。95 ページの『第 9 章 コマンド受信側』も参照してください。

PPI 受信側 ID は、英数字の大/小文字混合でなければなりません。この ID には「\$%&@#」を使用できます。長さは 8 文字で、ブランクで右寄せする必要があります。

### taskname

オプションのターゲット・タスク名。このパラメーターの最大長は 8 文字です。

指定しない場合、コマンド受信側は、どのタスクを使用するかをそれ自体で決定します。

セキュリティー・コンテキスト AUTOTASK の場合、コマンドは指定された **taskname** で実行されます。セキュリティー・コンテキスト USERTASK の場合、**taskname** は無視され、コマンドはユーザー・タスク下で実行されます。

**taskname** には以下の規則が適用されます。

1. **taskname** は、状況が ACTIVE である SA 自動化機能の名前です。可能な名前のリストについては、「IBM Z System Automation オペレーター・コマンド」の『DISPAOPS コマンド』を参照してください。
  - a. SA 自動化機能が単一の SA 自動化機能 (EVTOPER など) である場合は、関連する NetView タスクが選択されます。

- b. **taskname** が SA 自動化機能の配列を表す場合は、対応する NetView タスクが「ラウンドロビ  
ン」によってこの配列から選択されます。

例えば、**taskname** は、配列 (AOF CMD01、AOF CMD02...) を表す AOF CMDOPER です。CGlobals  
の具体的な定義には以下が含まれます: AOF CMDOPER.0 はタスク数、AOF CMDOPER. $n$  ( $n=1,2,\dots$ )  
は  $n$  番目の SA オペレーター機能です。したがって、 $n=1$  の場合は  
AOF CMDOPER.1=AOF CMD01 となります。CGlobals のこの配列は、INGRCRPC を呼び出す前に  
定義されている必要があります。

**taskname** の長さは 8 文字に制限されているため、CGlobal 配列の名前も 8 文字に制限されま  
す。

2. **taskname** は、ACTIVE 状況の有効な NetView AUTOTASK (AUTBASE など) です。  
3. そうでない場合 (タスク名が不明、タスクが非アクティブなど)、コマンドは実行されず、エラー・  
メッセージ ING332I が返されます。

#### resp\_name

応答を受け取る REXX ステムのオプションの名前。

指定しない場合、応答は要求されません (応答不要送信)。

REXX ステムの名前を指定する場合は、末尾がドット (output. など) でなければなりません。 .

返されたとき、output.0 には出力行数が含まれ、output. $n$  ( $n=1,2,\dots$ ) には  $n$  番目の出力行が含まれます。

#### timeout

INGRCRPC が応答を待つ秒数 (オプション)。

指定しない場合は、10 秒が使用されます。

タイムアウトが発生すると、INGRCRPC は以下のいずれかの戻りコードを返します。

#### 戻りコード 2

リモート・サイトは、これまで受信側 PPI キューに応答データを書き込んでいません。そのため、  
指定された時間フレーム内で、応答データはまったく受信されていません。コマンドが終了した  
か、まだ実行中である保証はありません。また、コマンドが実行を開始したという保証もありませ  
ん。コマンド受信側タスクのコマンド・キューに残っていて、実行を待機している可能性があります

#### 戻りコード 1

戻りコード 1 は、セキュリティー・コンテキスト USERTASK でのみ可能です。タイムアウトのため、  
非同期出力の待機中にコマンド受信側が中断されました。提供される応答データはメッセージ  
ING331I のみです。コマンドは実行中ですが、コマンドの実行は指定された時間内に完了しませ  
んでした。コマンドは、完了するまでリモート NetView タスクで実行される場合があります。ただ  
し、完了後にデータは返されません。

#### security\_ctx

オプションのセキュリティー・コンテキスト仕様。

#### AUTOTASK

コマンドは、コマンド受信側の開始パラメーターで指定された自動タスクである、SA オペレーター  
機能のセキュリティー・コンテキスト内で実行されます。

#### USERTASK

コマンドは、呼び出し側 TSO ユーザー ID と等しい NetView オペレーター・タスクのセキュリテ  
ィー・コンテキスト内で実行されます。

指定しない場合、セキュリティー・コンテキストは AUTOTASK です。詳しくは、[107 ページの『セキュ  
リティー考慮事項』](#)を参照してください。

#### 戻りコード

##### 0

コマンドが (正常または異常に) 実行され、コマンド応答が TSO 呼び出し元に返されました。詳しく  
は、[105 ページの『応答データ』](#)を参照してください。



- 1** NetView サイトでのタイムアウト (USERTASK でのみ提供されます)。
- 2** TSO サイトでのタイムアウト。
- 4** input パラメーターが欠落しているか無効です。  
 メッセージ INGPC012I を参照してください。ここに問題のある機能パラメーターが示されています。問題のある機能パラメーターの説明を調べて、正しい使用法を確認してください。
- 8** 応答システムへの書き込み中にエラーが発生しました。  
 REXX システム名が誤っているか、INGPCREX で内部エラーが発生した可能性があります。
- 9** 入力データ・システムからの読み取り中にエラーが発生しました。  
 システム・エレメント input\_data.0 に整数が含まれていないか、REXX システム名が正しくないか、または INGPCREX で内部エラーが発生したことが原因である可能性があります。
- 16** NetView PPI API の使用中にエラーが発生しました。  
 例えば、PPI キューへの書き込みが失敗したか、DSIPHONE OPEN、CLOSE、または READ が失敗しました。  
 詳しくは、付随するメッセージ INGPC010I または INGPC011I を参照してください。
- 17** セキュリティー・エラー。  
 TSO ユーザーは、この機能の使用を保護する SAF プロファイルにアクセスできません。詳しくは、[107 ページの『セキュリティ考慮事項』](#)を参照してください。
- 20** 内部エラー。REXX スクリプト・エラーです。例えば、REXX 構文エラーまたは NOVALUE 状態です。
- 24** 内部エラー。応答データ・ストリームが無効です。
- 25** 内部エラー。応答データ・ストリームをデコードできません。
- 28** 内部エラー。初期化に失敗しました。
- 32** 内部エラー。システム情報を取得できません。

### 応答データ

入力コマンドは、PIPE コマンドを介して NetView タスクで実行されます。

コマンドが REXX プログラムである場合は、REXX 命令 SAY または NetView コマンド **PIPE CONS ONLY** を使用して、応答データを書き込むことができます。

コマンド出力は自動的に収集され、システム「**resp.**」を使用して呼び出し側 TSO プログラムに返されます。戻りコードがゼロの場合にのみ、コマンド実行の応答を返すことができます。

```
RC=INGRCRPC(cmd,,,,'resp.')
```

resp.0	コマンド実行の出力行数に情報メッセージの 1 を加えたもの。
resp.1	常に情報メッセージ ING330I、ING331I、または ING332I のいずれか。
resp.n, n=2,3,...	コマンド実行から収集されたすべての出力行。



**RC=0、コマンド実行の成功**

コマンドが正常に実行されると、resp.1には常にメッセージ ING330Iが入ります。後続の応答行 (resp.n, n = 2,3 ...)には、コマンドによって書き込まれた出力データが含まれることがあります。

**RC=0、コマンド実行の失敗**

コマンドが正常に実行されないと、resp.1には常にメッセージ ING332Iが入ります。メッセージ ING332Iには、コマンドの非正常戻りコードが含まれています。

後続の応答行 (resp.n, n = 2,3 ...)には、コマンドによって書き込まれた出力データが含まれることがあります。この場合、コマンド実行の失敗の理由を説明する追加のエラー・メッセージが応答に含まれている可能性があります。

**RC=1、コマンド実行タスクへの接続が中断された**

戻りコードが1の場合、resp.1には常にメッセージ ING331Iが入ります。

コマンド・ハンドラーへの接続中断により、これ以上のコマンド応答を受け取りませんでした。

**RC=2、コマンド実行タスクへの接続でタイムアウトが発生した**

戻りコードが2の場合、接続がタイムアウトになったため、応答はまったくありません。そのため、要求を受け取ることができません。

**メッセージ ING330I、ING331I、ING332I、および ING335I**

INGRCRPCが応答を返す場合、応答の最初の行は常に以下のいずれかのメッセージです。

- ING330I Command execution was successful
- ING331I Command execution failed. Waiting for asynchronous output has been interrupted due to timeout.
- ING332I Command execution failed. RC=rc SYSTEM=sys Command=cmd REASON:xxxx

関数 INGRPCRPCがコマンドをターゲット・タスクに送信すると、コマンド実行される前にメッセージ ING335IがNETLOGに書き込まれます。例:

```
ING335I Execute remote command on behalf of JOB=BDOW USER=BDOW OPER=AUTCMD03
CLIST=TESTRPC COMMAND=TECHO
```

これらのメッセージの詳細な説明については、「IBM Z System Automation メッセージおよびコード」でこれらのメッセージを参照してください。

**例****使用例**

- NetView コマンド **RES** をデフォルトの受信側 **INGRCRCV** に送信します。最大 10 秒 (デフォルト) 間待機します。コマンド出力を受け取り、表示します。

```
rc = INGRPCRPC('RES',,,, 'out.')
if (rc=0) then
do i=1 to out.0
  say out.i
end
```

- コマンド **INGLIST** を PPI ID **PPIQNAME** の別のコマンド受信側に送信します。これにより、セキュリティー・コンテキスト **USERTASK** 内でコマンドが実行されます。応答を最大 30 秒間待機します。

```
rc = INGRPCRPC('INGLIST OUTMODE=LINE',, 'PPIQNAME', 'USERTASK', 'out.', 30)
```

- MVS コマンド **D A, BDOW\*** をデフォルトの受信側 **INGRCRCV** に送信します。TSO 側で応答を最大 30 秒待機しますが、最初の複数行メッセージを受け取るまで MVS コマンドの実行を停止します。

```
rc = INGRPCRPC('MVS D A, BDOW*',,,, 'out.', 30)
```

## プログラミング例

- この例は、RPC を介して **MYREXPGM** コマンドを呼び出し、出力を表示します。

```

/*REXX*/
context='AUTOTASK' /*auto task security context*/
context='USERTASK' /*user task security context*/
data.0=3
data.1='input line 1'
data.2='input line 2'
data.3='input line 3'
command = 'MYREXPGM' /*ANY NETV command */
RC = INGRCRPC(command,,
              'DATA.',,,
              'INGRCRCV',,, /*default command receiver*/
              ,, /*default task*/
              'RESP.',,
              10,,
              context)
SAY 'INGRCRPC RC='RC
if (rc=0) then
do i=1 to resp.0
  say resp.i
end
EXIT

```

- 以下の自己作成コマンド **MYREXPGM** は、入力データを受け取り、それを出力データとして返します。

```

/*REXX MYREXPGM*/
SAY 'MYREXPGM STARTED'
ADDRESS NETVASIS,
      'PIPE SAFE *',
      '| STEM SAFE.'
/*SAY may be used to return response data */
SAY 'MYREXPGM SAFE.0='safe.0
do i=1 to safe.0
  say 'OUTPUT('i'): ' safe.i
end
SAY 'MYREXPGM ENDED'
EXIT

```

## セキュリティ考慮事項

TSO 関数 INGRCRPC の使用と、対応する NetView タスクでのコマンドの実行の両方が、セキュリティ検査の対象になります。

### INGRCRPC の使用の保護

INGRCRPC 関数は、選択されたセキュリティ・コンテキスト (**security\_ctx** パラメーターで指定) に応じて、SAF プロファイル検査を実行します。

例えば、TSO ユーザー BOB が INGRCRPC 関数を使用する場合は、以下の許可が必要です。

#### AUTOTASK

```
PERMIT TSO.*.*.CMDRCVR.SEND CLASS(SYSAUTO) ID(BOB) ACC(READ)
```

#### USERTASK

```
PERMIT TSO.*.*.CMDRCVR.SEND.USERTASK CLASS(SYSAUTO) ID(BOB) ACC(READ)
```

注: SAF プロファイルが存在しない場合は、デフォルトでアクセス権限が付与されます。

### コマンド実行の保護

NetView タスク (**taskname** パラメーターで指定) によるコマンド (**command** パラメーターで指定) の実行は、SA サード・パーティー・セキュリティまたは NetView コマンド・セキュリティ (あるいはその両方) の対象となります。これは、選択したセキュリティ・コンテキストによって異なります。

### AUTOTASK

セキュリティ・コンテキスト AUTOTASK を使用すると、SA はサード・パーティー SAF 検査を実行します。

例えば、TSO ユーザー BOB が MYCMD コマンドを実行する場合は、次の許可が必要です。

```
PERMIT *.*.MYCMD CLASS(NETCMDS) ID(BOB) ACC(READ)
```

さらに、自動タスク (AUTCMDnn など) がコマンドの実行を許可されている必要があります。次の許可が必要です。

```
PERMIT *.*.MYCMD CLASS(NETCMDS) ID(AUTCMDnn) ACC(READ)
```

### USERTASK

セキュリティ・コンテキスト USERTASK の使用は、NetView コマンド・セキュリティの対象となります。

NetView には、さまざまな種類のコマンド・セキュリティが用意されています。NetView セキュリティ設定で CMDAUTH=SAF を選択すると、NetView はコマンド・セキュリティの SAF 検査を実行し、TSO ユーザー BOB は次の許可を必要とします。

```
PERMIT *.*.MYCMD CLASS(NETCMDS) ID(BOB) ACC(READ)
```

### その他の NetView セキュリティ

NetView タスクでコマンドが実行されている間、NetView は他の種類のセキュリティ検査を実行することがあります。例えば、データ・セット・セキュリティや組み込みコマンドのコマンド・セキュリティなどです。コマンドの実行は、純粋な NetView セキュリティの対象であり、コマンドを実行している NetView タスクによって異なります。

## 第 10 章 リレーショナル・データ・サービス (RDS) の使用可能化

SA z/OS は、INGRDS コマンドによって使用できるリレーショナル・データ・サービス (RDS) を提供しています。これは SA z/OS 組み込みリレーショナル・データ・テーブルの基本アクセス方式を備えています。INGRDS コマンドの構文は SQL の API に似ていますが、完全な SQL 言語構文解析プログラムは用意されていません。ただし、RDS は完全な SQL ではなく、SQL のすべての概念をサポートするわけではありません。SA z/OS はリレーショナル・データ・テーブルを VSAM ファイルに保存します。

高速なアクセスを実現するために、リレーショナル・データ・テーブルは、NetView アドレス・スペースの GETMAIN ストレージに保持されます。PTF UA54030 では、リレーショナル・データ・サービス・テーブルは 64 ビット・ストレージに保持されます。制御データのみが、NetView アドレス・スペースの 31 ビット・ストレージに保管されます。これにより、RDS テーブルに保管されるユーザー・データの容量が増加します。持続性は、DD INGEMUGL を使用して VSAM KSDS ファイルにテーブルを定期的に保存するアルゴリズムによって実現されます。テーブルのリストアは、NetView の開始時に SA の初期化で行われます。

各 SA z/OS エージェントは、その独自の VSAM データ・ファイルを、その関連データのみを保持するメモリー・テーブルに保持します。SA z/OS エージェントがダウンしていると、テーブル・データにはアクセスできません。

注：持続的な RDS テーブルを保持する SA/NetView エージェントごとに 1 つの VSAM KSDS ファイルがあります。SA/NetView が実行されている場合は、INGRDS を介してのみ RDS テーブルにアクセスできます。

### 永続リレーショナル・データ・テーブルの使用可能化/使用不可化

System Automation の初期化後、リレーショナル・データ・サービス (RDS) はいつでも使用可能です。ただし、アーカイブ・アプリケーション RDSARCH がバックグラウンドで定期的に行われる場合、またはコマンド INGRDS ARCHIVE を使用して特定の RDS テーブルを手動で保存する場合にのみ、新規または変更された RDS テーブルは保持されます。

RDS テーブルのアーカイブとリストアの前提条件は、VSAM ファイル INGEMUGL が存在することです。RDS テーブルは、System Automation / NetView の始動時にリストアされます。「IBM Z System Automation 計画とインストール」の章『SA z/OS のインストール』のステップ『リレーショナル・データ・サービスのインストール』を参照してください。

RDS が初期化された後、リソース RDSARCH を停止して RDS テーブルを持続的にすることを無効にすることができます。これにより、RDS テーブルのアーカイブが停止されます。ただし、RDS は引き続きメモリー内の RDS テーブルを処理します。

### System Automation リソースのインポート

アドオン PDB \*IBMCOMP で定義されている SA z/OS リソースを使用することをお勧めします。

以下のエントリーを選択します。

ポリシー・エントリー名	ポリシー・エントリー・タイプ
RDSARCH	APL
RDS_ARCHIVER	APG
RDS_AUTOOPS	AOP

これらの SA リソースにより、RDS アーカイブ・タスクが RDS テーブルを持続的にします。この利点は、アーカイブの開始または停止が簡単であることです。VSAM IO エラーなどのエラー・メッセージに反映さ

れたエラーの状態は、メッセージ・テーブルにトラップされ、SA リソースに関連付けられる場合もあります。リソースの状況が重大エラーを示す中断になることがあります。

RDS\_ARCHIVER と RDSARCH のインポート後、SA z/OS カスタマイズ・ダイアログで APL リソース RDSARCH を含む APG リソース・グループが定義されます。これは RDS のアーカイブを開始または停止する機能を備えています。PRESTART コマンドおよび REFRESHSTART コマンドとして INGTIMER を使用すると、自動化機能 AOFRDSAR で RDS アーカイブ・コマンドが *nn* 秒ごとに定期的にスケジュールされます。[110 ページの『通常のスナップショット』](#)を参照してください。

RDS\_AUTOOPS のインポート後、SA カスタマイズ・ダイアログで対応する自動化機能 AOFRDSAR と AOFRDSEV が定義されます。NetView 自動化オペレーターの名前は、メンバー ING.SINGPRM(AOFOPFSO) に定義される名前と同じでなければなりません。このサンプルは、必要に応じてカスタマイズすることができます。デフォルトでは、以下のマッピングが使用されます。

SA 自動化機能/オペレーター	NetView 自動化オペレーター
AOFRDSAR	AUTRDSAR
AOFRDSEV	AUTRDSEV

## 通常のスナップショット

デフォルトでは、RDS アーカイブ・リソース RDSARCH が 30 秒ごとにコマンド "INGVALUE ARCHIVE" を発行します。これは自動化機能 AOFRDSAR で実行されます。アーカイブとリストアは、常に同じタスクで実行されなければなりません。RDS アーカイブが正常に機能し、RDS テーブルのバックアップを定期的に行っていることを確認してください。

## RDS の初期化

DD INGEMUGL を持つ VSAM ファイルが存在する場合は、SA z/OS の初期化時に永続 RDS テーブルがメモリーにリストアされます。

この RDS 初期化の状況が OK でない場合は、次のメッセージが表示されることがあります。

```
ING388I Function or command INGRVCAC failed, RC=36 REASON=ARCHIVE rejected INIT STATUS=xxxx
```

xxxx が NULL スtring の場合は、RDS の初期化が実行されていません。

## TSO 用 RDS の使用

このセクションでは、TSO での System Automation (SA) リレーショナル・データ・サービスに関する一般的なガイドラインをいくつか示します。

INGRDS で提供される機能は TSO 下でも実行することができます。このような機能を実行するには、TSO クライアント・プログラム INGRCRDX を使用します (例: INGRCRDX QUERY mytable)。

このクライアント・プログラムは、ローカル SA NetView エージェントの背後でコマンド INGRDS を実行し、応答を受け取ります。INGRCRD を使用するには、以下のような、いくつかのインストール手順を実行する必要があります。

- コマンド受信側で使用する TSO 環境のセットアップ
  - SA 提供の TSO REXX 関数パッケージ INGTXFPG をインストールします。詳しくは、「IBM Z System Automation 計画とインストール」の『従来の SA z/OS 構成』で『TSO 用の関数パッケージを構成する』のステップを参照してください。
  - TSO プログラムが REXX ライブラリー SINGTRES を実行可能であることを確認します。
  - TSO 許可コマンド INGPAUTH をインストールします。詳しくは、「IBM Z System Automation 計画とインストール」の『従来の SA z/OS 構成』で『TSO 用の関数パッケージを構成する』のステップを参照してください。
- SA / NetView でのコマンド受信側のインストール



## TSO 環境のカスタマイズ

追加のインストール手順を実行する必要があります。110 ページの『TSO 用 RDS の使用』を参照してください。

## TSO における表示/編集のための RDS 作業データ・セットの定義

テーブルの表示と編集は、専用 PDS の一時メンバーを使用して行われます。NetView と TSO には、この PDS に対する読み取り/書き込みアクセス権限が必要です。PDS はこの機能のために予約されています。これは、ユーザーの要件に合った最大長のテーブル行に対応できる最大のレコード長を必要とします。

TSO ユーザーのために固有の一時メンバーが作成され、INGRDS EDIT または VIEW を呼び出します。この目的のために TSO ユーザー ID が使用されます。このメンバーは、以降の INGRDS EDIT または VIEW への呼び出しによって常に上書きされます。

RDS 作業データ・セットをすべての TSO アドレス・スペースおよび NetView が認識するように、次の AAO で設定する必要があります。

```
AOF_AAO_RDS_TSO_DSN=HLQ.RDS.WORK
```

データ・セット名 HLQ.RDS.WORK はカスタマイズ可能です。独自の名前が付いた PDS を作成できます。

最大レコード長が 1000 である PDS 特性の例 (テーブルのすべての列定義の合計が 1000 バイトより小さい場合):

```
Organization . . . . : P0
Record format . . . . : VB
Record length . . . . : 1000
Block size . . . . . : 32000
Data set name type : PDS
```

## TSO 内での RDS テーブルの表示

TSO で次のコマンドを使用します。

```
INGRCRDX VIEW mytab
INGRCRDX BROWSE mytab
```

これはテーブル **mytab** を取得し、それを ISPF エディターを使用して表示します。

## TSO を使用した RDS テーブルの編集

TSO で次のコマンドを使用します。

```
INGRCRDX EDIT mytab
```

これはテーブル **mytab** を取得し、それを ISPF エディターを使用して表示します。テーブルは変更可能です。下記の列指定に関する規則に従ってください。テーブルへの変更は一時ファイルに保存され、RDS テーブル **mytab** にインポートされます。テーブルのロックは、テーブルが保存または終了されるまでに取得されます。

次の規則が適用されます。

1. 列の長さを順守し、各列の間にブランクを 1 つ保持します。
2. 既存のテーブルの場合、列の定義は上書きできないため、列を削除したり、新しい列を挿入したりしないでください。
3. まだ存在しないテーブルの場合、行 1 のテーブル名を変更し、そのテーブルを保存した後、列の削除、挿入、および名前変更を行うことができます。括弧内の数値が列の幅と一致することを確認してください。新しいテーブル名は存在してはなりません。新しいテーブル名が既に存在すると、古い列定義が使用されます。



## 第 11 章 シスプレックス自動化の有効化

この情報では、Parallel Sysplex® 自動化のための SA z/OS 固有の機能、これらの機能を有効にするための SA z/OS カスタマイズ・ダイアログの使用法、およびご使用のシステムのカスタマイズ方法について説明します。

注：037 以外のホスト・コード・ページを使用している場合は、単価記号 (@) の 16 進数表記が異なることがあります。単価記号には、16 進コード X'7C' で表される文字を使用してください。

### シスプレックス機能

以下の機能について説明します。

- [113 ページの『結合データ・セットの管理』](#)
- [114 ページの『システム・ロガーの管理』](#)
- [115 ページの『カップリング・ファシリティーの管理』](#)
- [116 ページの『リカバリー・アクション』](#)
- [123 ページの『ハードウェアの検証』](#)

### 結合データ・セットの管理

結合データ・セット (CDS) には、シスプレックスとそのリソースに関する制御情報が含まれていて、並列シスプレックスが機能するためにきわめて重要なものです。

特に重要なのは SYSPLEX 結合データ・セットであり、これには、システムとシスプレックスの通信構造 (XCF グループ)、およびそのカップリング・ファシリティー (CF) と構造を指定する CFRM 結合データ・セットに関する情報が含まれています ([115 ページの『カップリング・ファシリティーの管理』](#)を参照)。並列シスプレックスの各 MVS システムは、これらの CDS と、SFM やアプリケーション応答測定 (ARM) などのインプリメントされた他のすべてのシスプレックス機能の CDS にアクセスできなければなりません。

メンバー・システムが CDS にアクセスできないと、対応するシスプレックス機能に影響し、場合によっては、そのシスプレックスがダウンすることがあります。したがって、シスプレックスのインプリメンテーションに必要な CDS のすべてのタイプの XCF に対して 2 つの CDS を定義することをお奨めします。その 1 つは、基本 CDS で、これは実際に使用する CDS です。もう 1 つは代替 CDS と呼ばれ、バックアップ・コピーとなります。この 2 つの CDS には同じデータが入っています。基本 CDS が変更されると、それに応じて XCF は代替 CDS を更新します。特定タイプの代替 CDS が使用可能な場合、メンバーが基本 CDS にアクセスできない時には常に、XCF はこの代替 CDS に自動的に切り替えます。

すべての CDS (シスプレックス結合データ・セットは除く) には、ポリシーと呼ばれる、1 つ以上のユーザー定義構成が含まれています。それぞれの CDS タイプでアクティブになるポリシーは 1 つだけです。ただし、実行時にアクティブ・ポリシーの切り替えが可能です。INGPLEX コマンドの詳細情報については、「[IBM Z System Automation オペレーター・コマンド](#)」を参照してください。

SA z/OS は、より簡単に CDS が管理できるように、次の 2 つの機能を提供します。

- 連続可用性のための代替結合データ・セットの自動作成とリカバリー
- 結合データ・セットの管理を単純化する INGPLEX CDS

#### 結合データ・セットの連続可用性の保証

指定された CDS タイプの代替 CDS が存在して、現行の基本 CDS が失敗すると、XCF はこの代替を基本 CDS にします。ただし、この切り替え後は、代替 CDS はもう存在しなくなり、現行の基本 CDS も失敗した場合は、代替の作成により回避された問題が再び起こります。この Single Point of Failure 状態を回避するため、SA z/OS は、使用されるどの CDS タイプでも代替 CDS が常に使用可能になるようにするリカバリー・メカニズムを提供しています。

SA z/OS は次の 2 つの状態で新規の代替 CDS を作成します。

- 初期化中に、それぞれの基本 CDS の代替 CDS が指定されていることを SA z/OS が検査します。代替 CDS が存在しない基本 CDS がある場合は、SA z/OS が自動的にそれを作成します。
- 実行時に、現行の代替が除去されるか、あるいは基本のものに切り替えられた時には常に、SA z/OS は新規の代替を作成するようにします。

### カスタマイズ

代替 CDS のリカバリーは、INGPLEX の CDS 機能、あるいはバックグラウンド (例えば、初期化の時) のいずれかで開始されます。バックグラウンド・リカバリーのスイッチは SA z/OS のカスタマイズ・ダイアログを使用してオン/オフにできます。INGPLEX CDS での自動再作成は常に使用能になります。

欠落の代替 CDS の作成のために SA z/OS が使用することがあるスペア・ボリュームを指定する必要があります (シスプレックス・グループの「ポリシーの選択 (Policy Selection)」パネルからポリシー項目 SYSPLEX を使用する)。また、これは INGPLEX CDS での自動作成にも必要となります。それぞれの CDS タイプには、それ自体のスペア・ボリュームのプールがあります。CDS タイプのスペア・ボリュームを定義しないと、このタイプのリカバリーが実行されないことに注意してください。カスタマイズ・ダイアログの使用の詳細については、[125 ページの『結合データ・セットの連続可用性を使用可能にする』](#)を参照してください。

シスプレックス構成を変更する INGPLEX CDS の機能へのアクセスを制御することができます。詳しくは、「*IBM Z System Automation* 計画とインストール」の『セキュリティおよび許可』の章を参照してください。

## システム・ロガーの管理

### 用語と概念

システム・ロガーは、シスプレックス全体にわたるロギング機能を提供します。システム・ロガーを使用するアプリケーションは、これらのログ・データをログ・ストリームに書き込みます。並列シスプレックス内では、これらのログ・ストリームは通常、カップリング・ファシリティ構造と関連付けられます。カップリング・ファシリティ構造の詳細については、[115 ページの『カップリング・ファシリティの管理』](#)を参照してください。カップリング・ファシリティ・ログ・ストリームを使用することによって、複数システム・アプリケーションのメンバーは、そのログが異なったシステム上に常駐する場合でも、それらをマージできます。

アプリケーションがログ・ストリームにデータを書き込む時は、このデータは初めに、関連の構造 (カップリング・ファシリティ・ログ・ストリーム) またはローカル・バッファ (DASD-only ログ・ストリーム) に一時的に保管されます。そこから、システム・ロガーによって自動的に割り振られるログ・ストリーム・データ・セットにオフロードされます。このログ・ストリーム・データ・セットが満杯になると、システム・ロガーは 2 番目のものを割り振ります。それ以降は同様です。

各ログ・ストリームのログ・ストリーム・データ・セットのディレクトリーを含むシステム・ロガーの制御情報は、LOGR 結合データ・セットに入れられます。システム・ロガーによって割り振りできるログ・ストリーム・データ・セットの総数は、LOGR 結合データ・セットをフォーマットする時に決定されます。

ログ・ストリーム・データ・セットと関連して起こり得る 2 つの問題は、LOGR CDS のディレクトリー・スペースの不足と、ログ・ストリーム・データ・セットの間違った共用オプションの問題です。SA z/OS はこれらの問題に対して以下のリカバリー・アクションを提供します。

- ディレクトリー不足が起こると、基本と代替 LOGR CDS が自動的にサイズ変更される
- ログ・ストリーム・データ・セットの共用オプションが正しく定義されない場合は、オペレーターに通知される

### ディレクトリー不足の場合の LOGR 結合データ・セットのサイズ変更

LOGR CDS には、システム・ロガーで使用されるログ・ストリーム・データ・セットに関する情報が入っています。この情報は、ディレクトリー範囲に保管されます。ディレクトリー範囲の各レコードは、最大 168 個のログ・ストリーム・データ・セットに関する情報を保持できます。LOGR CDS で使用できるディレクトリー範囲の数は、CDS をフォーマットする時に指定されます (DSEXTENT パラメーター)。使用可能なディレクトリー範囲をすべて使用してしまうと、システム・ロガーは新規のログ・ストリーム・データ・セットをそれ以上割り振ることはできません。システム・ロガーを使用するアプリケーションにとっては、これは大きな問題の原因になり得ます。

SA z/OS によって、この状態を回避することができます。ロガー・リカバリーのスイッチが入っていて、システムがディレクトリー不足を報告すると、常に SA z/OS は増加された DSEXTENT パラメーターにより基本と代替の LOGR CDS を自動的に再フォーマットします。

## カスタマイズ

システム・ロガー・リカバリーの自動化は、SA z/OS カスタマイズ・ダイアログを使用して使用可能になります。詳しくは、[168 ページの『システム・ログ障害のリカバリー』](#)を参照してください。

## カップリング・ファシリティの管理

カップリング・ファシリティ (CF) とは、並列シスプレックス内の異なるシステム上に分散されているアプリケーションのコンポーネント間のデータ交換のためのストレージを提供する論理区画です。並列シスプレックスには、複数の CF を入れることができます。

カップリング・ファシリティのストレージは、ストラクチャーと呼ばれる領域に分割されます。ストラクチャーは、特殊なデータ・セットと考えることができます。これは、ストラクチャー名によって識別され、アプリケーション・コンポーネントによる読み書きでアクセスされるストラクチャーです。

CF とストラクチャー間の関連付けは動的なものです。アプリケーションで使用されるストラクチャーは、(例えば、アプリケーションが実行中でない時は) 必ずしも割り振る必要がなく、別の時点で異なった CF 上に割り振ることができます。各ストラクチャーには、その割り振りが可能となる CF を定義する設定リストが存在します。そのリスト中の CF の順序は、リストの複数のメンバーがすべての割り振り要件 (例えば、十分なスペースを提供するなど) を満たす時に、どの CF を選択するかを決定します。

ストラクチャーの設定リスト、スペース所要量、およびその他のプロパティはアクティブ CFRM ポリシー内に定義されます。このポリシーは CFRM 結合データ・セット内に入れられます。詳しくは、[113 ページの『結合データ・セットの管理』](#)を参照してください。

XES は、アプリケーション・コンポーネントをストラクチャーに接続する必要がある時に、どの CF 上にもまだないストラクチャーを割り振ります。アプリケーション・コンポーネントでアクセスしたいストラクチャーの名前だけをコンポーネントが指定していることに注意してください。どの CF にそのストラクチャーを割り振るかを決定するのは XES です。この決定は、アクティブ CFRM ポリシーのストラクチャー定義の影響を受けます。ストラクチャーが割り振られると、要求元アプリケーション・コンポーネントはそのストラクチャーにアクセスすることができます。また、このアプリケーションの他のコンポーネントがそのストラクチャーへの接続を要求できます。割り振られたストラクチャーへのアクセスをもつアプリケーション・コンポーネントは、このストラクチャーへのアクティブ接続と呼ばれます。

最も単純な事例では、接続されたすべてのアプリケーション・コンポーネントをそのストラクチャーから切断した時に、XES はストラクチャーの割り振りを解除します。ただし、アプリケーション・コンポーネントは、そのストラクチャーまたはストラクチャーへのそれ自体の接続を持続するように要求できます。ストラクチャーが持続されているときは、アプリケーション・コンポーネントがもうそのストラクチャーに接続されていないときでも、割り振られたままになります。接続が持続されているときは、その接続の失敗の後もそのストラクチャーは割り振られたままとなります。問題のアプリケーション・コンポーネントはアクティブではないのですが、そのストラクチャーへのコネクタのままとなります。これは現在、失敗した持続コネクタです。両方の事例で、アクティブ・コネクタがストラクチャーになくなると同時に、その構造の割り振り解除を強制できます。

割り振られたストラクチャーは再構築することができます。再構築とは、同じ、または別の CF 上でストラクチャーを再構成する処理です。再構築は 3 つのメイン・ステップで構成されています。最初に、XES は新規のストラクチャー・インスタンスを割り振ります。次に、古いストラクチャーのデータを新しいストラクチャーで再構成します。最後に、XES は古いストラクチャー・インスタンスの割り振りを解除します。再構築の要求ではターゲット CF を指定できないことに注意してください。ストラクチャー割り振りと同じように、XES は設定リストからそれを選択します。

再構築には、ユーザー管理とシステム管理の 2 つの方式があります。ユーザー管理再構築では、アクティブ・コネクタがデータ再構成の責任を負います。システム管理再構築では、XES は新規のストラクチャー・インスタンスにデータを転送します。したがって、システム管理再構築はアクティブ・コネクタがないストラクチャーでも使用可能になります。これらのストラクチャー自体が持続接続であることも、失敗した持続接続であることもあります。

アプリケーション・コンポーネントがストラクチャーに接続する時に、そのストラクチャーの再構築をユーザー管理再構築またはシステム管理再構築によって可能にするかどうかをそのコンポーネントが指定し

まず、アクティブ・コネクタのあるストラクチャーでは、再構築の両方の方式で、すべてのアクティブ・コネクタがそれぞれの再構築方式を可能にする必要があります。

また、ストラクチャーを二重化することもできます。二重化とは、異なった CF 上に同時に同じストラクチャーの 2 つのインスタンスを維持することを意味します。二重化は、ストラクチャーの可用性と使用可能度を向上させるのに役立ちます。

CF の典型的な管理タスクは、シスプレックスから CF を除去し、それを再統合することです。これらのタスクには、特定の順序で実行する必要があり、かなり複雑になる可能性があるステップがいくつかあります。これらの操作を単純化するために、SA z/OS は INGCF コマンドを提供します。INGCF にはいくつかの機能があり、ストラクチャーと CF 自体の取り扱いに役立ちます。詳細については、「*IBM Z System Automation* オペレーター・コマンド」と、オンライン・ヘルプを参照してください。

特定の機能がカップリング・ファシリティーの送信者パスを扱います。これには、次の制限があります。まず始めに、自動化を実行しているシスプレックスの少なくとも 1 つのシステムが、カップリング・ファシリティーの制御装置 ID (CUID) を認識していなければなりません。そうでない場合は、欠落した送信側のパスを解決することはできません。

システム IPL (または再 IPL) の前にカップリング・ファシリティーが非活動化され、その後で活動化されると、送信側のパスの欠落が起こります。IPL (または再 IPL) されたシステムはカップリング・ファシリティーを認識しません。欠落した送信側のパスを決定するために、自動化は HCD の HOM インターフェースを呼び出します。欠落したパス情報の解決が可能になるのは、完全なネットワーク・アドレスがプロセッサ ID とともに HCD に定義されているか、あるいは自動化で使用する CPC シノニムをプロセッサ ID として指定した場合だけです。ただし、この両方を定義することをお奨めします。このいずれも定義しない場合は、送信側のパスの欠落しているシステムが自動化を実行しなければなりません。

## リカバリー・アクション

### WTO(R) バッファ不足の解決

WTO(R) のすべてのバッファが使用中の場合は、コマンドの処理ができないことがあります。これを解決するためのオプションがいくつかあります。すなわち、バッファを拡張、影響を受けるコンソールのプロパティを変更、あるいは WTO(R) を出すジョブをキャンセルすることができます。

SA z/OS はバッファ不足のリカバリーを 2 段階で提供します。最初に、バッファを拡張して、適用できる場合は、コンソールの特性を変更しようとします。これで解決しない場合は、WTO(R) を出すジョブをキャンセルします。バッファ不足がある場合は、どのジョブを SA z/OS でキャンセルできるかを指定する必要があります。

**カスタマイズ:** バッファ不足リカバリーの自動化は、SA z/OS カスタマイズ・ダイアログを使用して使用可能にすることができます。詳しくは、[126 ページの『WTO\(R\) バッファ不足のリカバリーを使用可能にする』](#)を参照してください。

SA z/OS によって異なったメッセージが受信されると、「WTO リカバリー」が実行されます。これらの各メッセージを受信した時に取られるアクションは、[117 ページの表 21](#) に説明されています。



表 21. WTOBUF リカバリー処理				
リカバリー	メッセージ		コマンド	
WTO	IEA405E	コンソール属性を設定します。		
		削除モードがロールまたはラップでない場合は、そのモードをロールに設定します。	K S,DEL=R,L=x	
		行外の表示域が存在する場合は、状況表示を削除します。	K E,D,L=x	
		メッセージ・ロール間の間隔が1秒より大きく、かつ「*」でない場合は、その間隔を0.25秒に設定します。	K S,RTME=1/4,L=x	
		コンソールで受信するメッセージがローカル・システムからのものだけではない場合、WTOメッセージ・バッファーがその最大サイズに達すると、バッファリング・システムをリストから除去し、ローカル・システムをリストに追加します。	V CN(x),MSCOPE=(1)	
	IEA404A	コンソールを中断します。		
		メッセージをハードコピー・ログに再キューイングします。	K Q,L=x	
		アクティブ・コンソール (COND=A) をオフラインに変更します。SMCSコンソールの場合、適切なVTAMコマンドを実行します。	V {CN(x),OFFLINE   NET,TERM,LU1=x, TYPE=FORCE }	
		不足の原因になったジョブまたはTSOユーザーをキャンセルしますが、カスタマイズ中に候補として定義された場合に限りです。	C {jobnm,A=asid  U=userid }	
	IEA406I	コンソールがサスペンドされた場合、SMCSコンソールでなければ再開します。	V CN(x),ONLINE	
		コンソール属性を復元します。		
		削除モードを、バッファ不足が起こる前の値に設定します。	K S,DEL=old,L=x	
		メッセージ・ロールの間隔を、バッファ不足が起こる前の値に設定します。	K S,RTME=old,L=x	
		コンソールが非送信請求メッセージを受信する元のリストを、バッファ不足が起こる前のリストに設定します。	V CN(x),MSCOPE=(1)	
		次のようにWTOメッセージ・バッファ・サイズを増加させて、将来の不足を最小化します。  new = min(9999,max(1500,1.2 * current MLIM))	K M,MLIM=new	
永続変更 (MLIM) の場合にメッセージ AOF929 を出します。				

表 21. WTOBUF リカバリー処理 (続き)			
リカバリー	メッセージ		コマンド
WTOR	IEA230E	シスプレックスの最大システム数が 8 以上であるか、あるいはシステムがローカル・モードで稼働する場合は、応答 ID の最大数を許容最大値に増加させます。	K M,RMAX=9999
		現行の RMAX 値が現行の RLIM 値より大きい場合は、次のように WTOR メッセージ・バッファのサイズを増加させます。  $\text{new} = \min(9999, \max(10 + 2 * \text{maxsys\_in\_sysplex}, 1.2 * \text{current RLIM}))$	K M,RLIM=new
	IEA231A	未処理の WTOR があり、カスタマイズ中に候補として定義された、ジョブと TSO ユーザーをすべて取り消します。	C {jobnm,A=asid  U=userid }
IEA232I	不可逆的な変更 (RMAX) の場合にメッセージ AOF928 を出します。永続変更 (RLIM) の場合にメッセージ AOF929 を出します。		

### 長時間エンキュー (ENQ) の処理

このタイプのリカバリーは、以下の独立した機能に区分されます。

- 長時間エンキュー・リカバリー
- 「ハング」コマンド・リカバリー
- コマンド・フラッディング・リカバリー

これらすべてのリカバリーは、個別にまたは一括で使用可能および使用不可にできます。

長時間エンキュー・リカバリー機能では、以下のことができます。

- どのリソースがブロックされるかの検査
- リソースをブロックするジョブをキャンセルまたは保持するための自動化のカスタマイズ
- ジョブをキャンセルする前にそれをダンプするための自動化のカスタマイズ

モニターしたいリソースをユーザーが決定できます。他のジョブがリソースを待っている間に、ジョブがそのリソースをロックできる最大時間の値を定義できます。この時間が超過した場合、リカバリーが行われます。これらの潜在的なボトルネックを識別して除去すると、並列シスプレックスの障害のリスクを削減できます。

組み込みリストは時間定義で記述されますが、モニターしないリソースの除外リストを定義することも可能です。

ENQ 機能の詳細については、[129 ページ](#)の『[長時間エンキュー \(ENQ\) を使用可能にする](#)』を参照してください。

この機能は、以下の 2 つの補足機能によって拡張されています。

- [118 ページ](#)の『[「ハング」コマンド・リカバリー](#)』
- [119 ページ](#)の『[コマンド・フラッディング・リカバリー](#)』

### 「ハング」コマンド・リカバリー

この機能の目的は、マルチシステム障害の原因となることが多いハング・コマンドを検出することです。次の 3 つの状態を区別します。

1. その他のコマンドの実行完了を妨げるコマンド
2. ジョブの実行完了を妨げるコマンド
3. コマンドの実行完了を妨げるジョブ

自動化は、コマンド処理に関連付けられた ENQ 競合を検査し、ブロッカーとウェイターのリストを作成します。次に、SA z/OS ポリシーが検査され、自動化アクションが実行される前に待機コマンドと待機ジョブが待機を許可される時間が調べられます。ポリシーも検査され、以下のようにブロッキング・コマンドまたはブロッキング・ジョブに対して実行されるアクション (DUMP、NODUMP、KEEP または除外) が判別されます。

1. あるコマンドが他のコマンドの完了を妨げ、さらにいずれの待機コマンドにもポリシー定義が存在しない場合、自動化アクションは実行されない。
2. コマンドがジョブの完了を妨げ、さらにブロッキング・コマンドにポリシー定義が存在しない場合、自動化アクションは実行されない。
3. ジョブがコマンドの完了を妨げ、さらにいずれの待機コマンドにもポリシー定義が存在しない場合、自動化アクションは実行されない。

長時間 ENQ リカバリーとハング・コマンド・リカバリーが、自動化アクションを必要とする同一リソースを同時に検出した場合、ハング・コマンド・リカバリーのポリシー定義が優先され、ハング・コマンド・リカバリーがリソースを自動化します。

実行されるアクション (DUMP、NODUMP、KEEP または除外) は、長時間 ENQ リカバリー・アクションと同一です。

いずれの場合も、ブロックされているリソース上で待機中のコマンドのみが対象となります。「ハング」コマンド・リカバリーは、長時間 ENQ リカバリーでモニターされていないリソースのみを対象とします。長時間 ENQ リカバリーが使用不可の場合、すべてのリソースは、長時間 ENQ リソースと定義されているものであっても、「ハング」コマンド・リカバリーの対象となります。長時間 ENQ リカバリーが使用可能であり、汎用「catchall」リソース定義が適用される場合、長時間 ENQ リカバリーが常に優先されるために、「ハング」コマンド・リカバリーは実行できないことも理解しておいてください。

コマンドは、マスター・アドレス・スペースおよびコンソール・アドレス・スペースによって実行されます。したがって、リソース・ブロッカーがこれらのアドレス・スペースのいずれかから実行されたものであれば、そのブロッカーはブロッキング・ジョブではなくブロッキング・コマンドと見なされます。

リソースと同様に、別のコマンドがリソースを待機しているときに、あるコマンドがそのリソースをロックできる時間を決定する同様の定義をコマンドに対して行うことができます。

リソース・ブロッカーがジョブの場合、リカバリー・アクションが実行されるのは、ジョブがコマンドをブロックし、「ハング」コマンド・リカバリー処理が 3 回連続して繰り返された場合のみです。この結果、90 秒から 120 秒を超えない時間の間、ジョブがコマンドをブロックします。

ブロッキング・ジョブまたはブロッキング・コマンドを出したジョブに対するリカバリー・アクションは、長時間 ENQ リカバリーの自動化に指定されたリカバリー・アクションと同一です。

### コマンド・フラッディング・リカバリー

この機能の目的は、コマンド・クラスをあふれさせるジョブを検出することです。コマンド・フラッディングは、ログ・バッファが不足する原因となり、またその他のコマンドの実行も妨げます。どちらもマルチシステム障害をもたらす可能性があります。

コマンド処理用に予約済みのすべての (50) TCB が使用中の場合、新規コマンドは待機キューに入れられません。この場合、システムはメッセージ IEE806A を出します。これは、この機能を起動して、状態の原因となっているジョブを評価します。

200 個 (またはそれ以上) の「VARY dev,ONLINE」コマンドなど、一連のコマンドを発行するだけのジョブは、評価時に考慮しないでください。これは、影響を受けているコマンド・クラスの現在のスナップショットと以前のスナップショットとを比較することで行います。

スナップショット処理は、メッセージ IEE806A がトラップされるとスケジュールに入れられます。スナップショットの時間間隔は、デフォルトでは 3 秒です (この値の調整が必要な場合、詳しくは、[129 ページの『長時間エンキュー \(ENQ\) を使用可能にする』](#)を参照してください)。この時間間隔では、最初のスナップショットが取られる前に、これらのジョブがコマンド発行を完了するのに十分な時間を与える必要があります。連続した 2 つのスナップショットにコマンドを出すジョブのみが、リカバリー・アクションの対象になります。



リカバリー・アクションが実行される前に、ジョブが実行するコマンドの回数がしきい値(下記を参照)を超えている必要があり、少なくともコマンドの1つが「ハング」コマンド・リカバリーによって処理されるロック競合に含まれてはなりません。

リカバリー・アクションは、ジョブ定義によって決まります(129 ページの『[長時間エンキュー \(ENQ\) を使用可能にする](#)』を参照)。ジョブがキャンセルされると、リカバリーはその待機コマンドも削除し、その実行中のコマンドも終了します。リカバリー・アクションは、メッセージ ING922E またはメッセージ ING924E のいずれかで完了します。後者のメッセージは、待機キューが空になるまで約 1 分間隔で繰り返し出されます。

しきい値の計算は、コマンド処理用に予約済みの TCB (50) の合計数から、コマンド・クラス内でコマンドを実行しているジョブの数を減算して行います。これにより、数少ないコマンドを繰り返し実行するジョブが評価されないようにします。

メッセージ IEE061I が出されると、リカバリーは終了です。

**注:** ジョブがキャンセルされてダンプが取られる場合は、ダンプ定義は無効になります。これは、キャンセル中のジョブのリカバリー・ルーチンがダンプを抑止するためです。

### カスタマイズ

長時間エンキューの処理の自動化は SA z/OS カスタマイズ・ダイアログで使用可能にすることができます。詳しくは、129 ページの『[長時間エンキュー \(ENQ\) を使用可能にする](#)』を参照してください。

### システムの除去の管理

この機能の目的は、失敗したシステムをできるだけ迅速に除去して、それを並列シスプレックスから分離することです。また、回避できない障害が起こった場合、すぐに再始動したいそれらのシステム・イメージの平均復旧時間 (MTTR) を短縮することです。

**注:** この機能が使用可能な場合であっても、z/VM のもとで実行する z/OS イメージで実行中は、この機能は使用できません。

特に、この機能はメッセージ IXC102A と IXC402D を自動化します。

IXC102A メッセージの自動化がシスプレックス障害管理 (SFM) を実行します。特定の環境では、SFM が失敗したシステムの分離を完了できないことがあります。SFM の HW 分離 (失敗したシステムのチャネル・サブシステム (CSS) のリセット) は、CF によって行われるためです。システム・イメージとカップリング・ファシリティ間の接続が失われると、SFM はハードウェア分離 (ISOLATE コマンド) を実行できないので、手動のオペレーター介入が行われるまで、システム・イメージのリセットが据え置かれます。メッセージ IXC102A はオペレーターに対して HW の手動リセットを通知してから、メッセージに対して「DOWN」と応答し、その後 SFM がシスプレックスからシステム・イメージを安全に区分します。遅延が長くなれば、XCF メッセージングに依存するコンポーネントやアプリケーションへの影響が大きくなります。失敗したシステムに保留の入出力操作がある時には、この遅延によりシスプレックス障害が生じる結果となることがあります。このメッセージの自動化により、この遅延を最小限にします。

メッセージ IXC402D は IXC102A と同じ影響があります。ただし、状況更新が欠落しているために、このメッセージはシステムで起こりうる一時的な操作不能の状況を示します。このため、この自動化では、最初にメッセージ IXC402D が現れる位置で「INTERVAL=sss」と応答することによって、除去が行われる前にリカバリーのチャンスをシステムに提供します。インターバル時間の sss は、コマンド D XCF,CPL によって表示される障害検出間隔です。

自動化では、システムの除去は 2 段階で行われます。最初の段階では、「サポート・エレメント」にハードウェア・コマンドを送信して、保留中の入出力操作を消去します。これには、ハードウェアで実行しているソフトウェアに関する情報が必要です。システム送出メッセージ IXC102A または IXC402D は、失敗したシステムのハードウェアにアクセスできるとは限らないので、自動化では、ハードウェアとソフトウェア間の事前定義マッピングが必要です。このマッピングによって、失敗したシステムのハードウェアにアクセスできるシステムに対して、ハードウェア・コマンドが発送されます。マッピングを実行する方法については、127 ページの『[システムの除去を使用可能にする](#)』を参照してください。ハードウェア要件の詳細については、「*IBM Z System Automation* 計画とインストール」を参照してください。

2 番目の段階では、未解決の WTOR に対して、シスプレックスからのシステムの除去をトリガーする「DOWN」として応答します。

## カスタマイズ

メッセージ IXC102A の自動化は SA z/OS カスタマイズ・ダイアログを通じて使用可能になります。詳しくは、[128 ページの『ステップ 4: IXC102A と IXC402D メッセージの自動化』](#)を参照してください。

### 補助記憶域不足のリカバリー

自動化フラグ MVSESA.PAGE のリカバリー・フラグで制御されるローカル・ページ・データ・セットの自動化を使用して、必要な場合にスベア・ローカル・ページ・データ・セットを動的に割り振ることによって補助記憶域不足の障害を回避できます。

この機能は、どのジョブが不足状態の原因であるか、さらにページ・データ・セットを追加できるかどうかを検査します。これが可能でない場合は、不足の原因となっているジョブがキャンセルされます (そのように定義されている場合)。詳しくは、[122 ページの表 22](#)を参照してください。

ローカル・ページ・データ・セットの自動化を使用可能にするには、(IPL 中に使用される IEASYSxx PARMLIB メンバーの 1 つに定義されている) PAGTOTL パラメーターをカスタマイズします。PAGTOTL パラメーターは、現在使用しているローカル・ページ・データ・セット数より大きい値に設定してください。

ローカル・ページ・データ・セットはマスター・カタログに定義する必要があり、SMS 管理にしないでください。動的に割り振るのではなく、事前割り振りのローカル・データ・セットを使用することをお勧めします。新規に割り振るページ・データ・セットのフォーマット設定は時間がかかるため、事前割り振りのローカル・データ・セットを使用することで処理が速くなります。事前定義の各ローカル・ページ・データ・セットは、システムで現在使用されているローカル・ページ・スペースの 10% のスペースで割り振られます。事前定義のページ・データ・セットを割り振ることができない場合は、新規のローカル・ページ・データ・セットが動的に作成されます。

以下の表では、リカバリー・アクションの詳細を示します。一般的には次のようになります。

- リカバリーが正常に完了できない場合は、常にメッセージ AOF953 と特定の理由コードを出して終了します。
- ページ・データ・セット不足のために呼び出されると、リカバリーは最初に、フォーマット済みページ・データ・セット、および動的に作成されるデータ・セットの HLQ が定義されているかどうかを検査します。何も定義されていない場合、リカバリーは RSN=4 で終了します。
- コマンド PAGEADD が IEE782I と IEE783I 以外のメッセージを戻した場合、リカバリーは RSN=24 で終了します。
- IDCAMS サービスが失敗した場合、リカバリーは RSN=16 で終了します。

補助記憶管理 (ASM) を担当する WorkLoad Manager (WLM) では、z/OS リリース 1.10 以降にメッセージ IRA204E から置き換えられたメッセージ IRA206E ではなく、メッセージ IRA210E を自動化することをお勧めします。これは、不足の原因となったアドレス・スペースがシステムで既に識別されているためです。

zBC12 および zEC12 で使用可能なストレージ・クラス・メモリー (SCM) では、APAR OA38660 適用済みの z/OS リリース 1.13 以上が必要です。

表 22. 補助記憶管理			
リカバリー	メッセージ		コマンド
MSG	IRA205I IRA200E IRA201E IRA265I IRA260E	ローカル・ページ・データ・セットの最大数を検査します。PAGTOTL の値に達すると、リカバリーは RSN=20 で終了します。	MVS D ASM, LOCAL
		前のリカバリーで割り振られて、フォーマットされた空きページ・データ・セットを追加します。	MVS PAGEADD
		前のアクションが失敗し、メッセージが IRA201E でない場合、ページ・データ・セットを作成および追加します。	MVS PAGEADD
		前のアクションが失敗した場合、フォーマット済みデータ・セットを追加します。	MVS PAGEADD
		前のアクションが失敗し、メッセージが IRA201E である場合、ページ・データ・セットを作成および追加します。	MVS PAGEADD
		PAGEADD が正常に完了した場合は、このアクションが十分でない場合に備えてさらにページ・データ・セットを追加するために、タイマー INGT742P をスケジュールします。正常に完了しなかった場合、リカバリーは RSN=8 または RSN=12 で終了します。	AFTER 00.00:10, ID=INGT742P
IRA203E IRA204E IRA210E	最初のメッセージのみのジョブ名を保存します。次に、ジョブがキャンセルに対して定義されているかどうかを検査します。最後に、タイマー INGT742P より 10 秒遅くポップアップするようにタイマーをスケジュールします。このタイマーが存在しない場合は、現在時刻を使用します。	AT hh:mm:ss ID=INGT742J	
IRA202I IRA262I	保留中のリカバリー・アクションを削除します。	MVS D ASM, LOCAL PURGE TIMER=INGT742P PURGE TIMER=INGT742J PURGE TIMER=INGT742S	
IRA250I	SCM オフライン・ストレージが使用可能かどうかを判別します。ストレージをこれ以上追加できない場合、リカバリーは RSN=0 で終了しますが、IRA200E が出された場合は続行します。	MVS D M=SCM	
	SCM ストレージの「増分サイズ」を追加します。	CF SCM(incr), ON	
	CF SCM が正常なスケジュールであった場合は、このアクションに備えて SCM ストレージをさらに追加するためのタイマーが不十分でした。それ以外の場合、リカバリーは RSN=28 で終了します。	AFTER 00:00:10, ID INGT742S	
IRA252I	SCM 保留リカバリー・アクションを削除します。	PURGE TIMER=INGT742S	
ILR009E	ページ・データ・セットがスベア・データ・セットである場合、そのページ・データ・セットに「使用不可」とマークし、動的割り振りに使用可能なボリュームのリストからそのボリュームを除去します。	MVS D ASM, LOCAL	
	データ・セットがスベア・データ・セットであった場合: <ul style="list-style-type: none"> <li>ローカル・ページ・データ・セットの最大数を検査します。PAGTOTL の値に達すると、リカバリーは RSN=20 で終了します。</li> <li>ページ・データ・セットを作成および追加します。PAGEADD が失敗した場合、リカバリーは RSN=8 または RSN=12 で終了します。</li> </ul>	PAGEADD	
IEE205I	ページ・データ・セットがスベア・データ・セットである場合、そのページ・データ・セットに「空き」とマークします。		

表 22. 補助記憶管理 (続き)			
リカバリー	メッセージ		コマンド
PAGE	該当なし	タイマー INGT742P が満了した後にメッセージ IRA200E、IRA205I、IRA260E、および IRA265I のリカバリーを繰り返します。	
PAGECRIT	該当なし	タイマー INGT742P が満了した後に IRA201E メッセージのリカバリーを繰り返します。	
SCM	該当なし	タイマー INGT742S が満了した後に IRA250I メッセージのリカバリーを繰り返します。	
JOB	該当なし	タイマー INGT742J が満了した後にリカバリーを起こしたジョブをキャンセルします。	MVS CANCEL

## ハードウェアの検証

この機能は、カスタマイズ・ダイアログでマップしたハードウェア構成と、実行している実際のハードウェア構成間の相互検証を実行します。この情報は、BCP 内部インターフェースを通じて HMC/SE LAN 内のサポートされる任意の CPC 上の論理区画 (LPAR) を実際に制御するために重要となります。

ハードウェア検証は CPC 名、区画名と区画番号を使用して、カスタマイズ・ダイアログで定義された LPAR が正しい CPC 上にあり、正しい区画番号に配置されていることを確認します。ただし、それらの区画 ID は、アクティブ CFRM ポリシー内に定義する必要があるため、これはカップリング・ファシリティーでのみ有効です。

MVS イメージでは、HMC/SE からの情報 (初期化中に保管されるシステム名やシスプレックス名など) は、対応するカスタマイズ・ダイアログ定義を検査するために使用されます。自動化のハードウェア・コマンド・インターフェースの初期化中、および介入要求を区画に送信する直前に、すべてのものが正確に一致していることを確認するための新規の検査が行われます。

**注:** 検査できるのはアクティブ・イメージだけです。非アクティブ・イメージについては、カスタマイズ・ダイアログで行った定義を信頼する必要があります。

この場合のアクティブ・システムとは、ハードウェアの検証を実行するシステムと同じシスプレックスに属しているシステムのことです。すなわち、SA z/OS はそれ自体のシスプレックス内のシステムとカップリング・ファシリティーだけを検査します。

ハードウェアの検証は、主に開始時に SA z/OS システム上で実行され、それ以降は、カスタマイズ・ダイアログの定義に対する変更が、INGAMS REFRESH コマンドで適用される時に実行されます。検証では、登録された全システムの定義を検査します。すなわち、SA z/OS システムがハードウェアの検証を行う時には常に、その時点でシスプレックス内でアクティブであるシステムとカップリング・ファシリティーをすべて検査します。登録されたシステムとは、同じ XCF グループに加わった SA z/OS を実行するシステムのことです。

アクティブ・システムとカップリング・ファシリティーの検証では、アクティブ・システムのホストとなっている CPC はすべてカスタマイズ・ダイアログに定義されていなければなりません。

非アクティブ・システムのデータは検査できません。ただし、これらの定義の整合性については、登録されたすべてのシステム間で検査されます。これらの非アクティブ・システムまたはカップリング・ファシリティーの 1 つがシスプレックスに加わるか、あるいは使用可能になると、直ちに、特定のイメージのみの検証が実行されます。

ハードウェア情報の実際の検索には、CPC 当たり最大 5 分ほどかかりますが、これは、モデルやその LPAR によって異なります。ハードウェアの検証を実行する間、他のすべてのハードウェア関連の自動化は遅延するか、あるいは実行不能となりますが、これは、リカバリーのタイプによって異なります。このために、検証では「デルタ」処理が実行されます。これは、変更したデータのみを検査します。また、これには、検証の前に適用された CPC 定義が欠落している時に CPC 接続が終了する結果となるデータの不在も含まれます。検証の結果によるアクションは登録された「すべての」システム上で実行されます。これには、次の 2 つの利点があります。

- ハードウェア定義の変更の際、NetView の再生が不要である。
- あるシステムに対して使用できる変更を実行するだけで済む。

ACF コマンドまたは自動化の開始によって起動されるハードウェアの検証の最初の部分で、CPC のどの接続を終了して開始 (すなわち、この順序で) する必要があるかを決定します。その結果アクションが登録されたすべてのシステムで実行されます。このステップが正常に完了すると、イメージの検証が行われます。

イメージの検証では、実際のハードウェア情報を収集して、現在のハードウェア定義と、登録された他のすべてのシステムで検出した実際のデータや定義とを検証します。次の場合に通知が出されます。

- 実際のハードウェア情報またはユーザー定義が使用できないために、実システムまたはカップリング・ファシリティーの検証が実行できなかった。
- 実際のハードウェア情報が使用できないために、イメージ定義を評価できなかった。
- 実システムまたはカップリング・ファシリティーがアクティブではなく、登録された特定のシステムのイメージ定義が異なっている。
- 適切に定義されなかった定義の値が訂正されたか、あるいはまったく定義されていない。

ハードウェア定義に対する変更は、登録されているシステムのうち 1 つのシステムでのみ コマンド `INGAMS REFRESH` を実行するだけで、これらすべての登録済みシステムで変更を使用可能にできます。しかし、特定 CPC との通信で使用される許可トークン値の変更は例外です。この値の変更には、次の 3 つのステップが必要となります。

1. 最初のステップでは、特定 CPC の定義を除去してから、上記のように `ACF` コマンドを実行します。
2. そのコマンドが正常に完了したら、次のステップで、「サポート・エレメント」の CPC の許可トークン値を変更します。
3. 最終ステップでは、CPC を新規のトークン値で再び定義して、`ACF` コマンドをもう一度実行します。

注: `INGAMS` コマンドのこの動作が適用されるのは、ハードウェア定義の場合のみです。

2 番目の検証では、カップリング・ファシリティーを使用可能にする時に送られるメッセージ `IXC517I` によって、あるいはシステムがシスプレックスに加わったことを通知された際、その自動化自体のいずれかによって起動されます。いずれの場合も、自動化によって新規のシステムまたはカップリング・ファシリティーの検証のみを実行します。このシステムまたはカップリング・ファシリティーの検証中は、同じシステムまたはカップリング・ファシリティーのメッセージが複数回か表示されても無視されます。新規のシステムの場合、システムが `NetView` と自動化を開始する前に実際のハードウェアが検証されることが利点となります。この自動化で、現行定義と登録された他のシステム定義とで相違点が検出されない場合 — これが通常 — には、整合性検査のみが行われます。この検査では、実際のハードウェア情報は不要です。

### 前提条件

注: `z/VM` のもとで実行される `MVS` システムでは、ハードウェアの検証はサポートされていません。

## ハードウェア関連の自動化を使用可能にする

リカバリー・アクションやカップリング・ファシリティー管理のために `SA z/OS` が提供するシスプレックスの自動化を使用可能にするには、カスタマイズ・ダイアログで次の定義を行う必要があります。

### ステップ 1: プロセッサの定義

#### このタスクについて

項目タイプ `PRO` の新規のプロセッサを定義するには、カスタマイズ・ダイアログを使用します。名前は、`HCD` に定義されたプロセッサの実物理名でなければなりません。詳しくは、オンライン・ヘルプまたは「*IBM Z System Automation* 自動化ポリシーの定義」のセクション『新規プロセッサの作成』を参照してください。

### ステップ 2: ポリシー項目 `PROCESSOR INFO` の使用

項目タイプ `PRO` を使用してプロセッサを定義するには、「プロセッサ情報 (Processor Information)」パネルを使用します。

注: 接続タイプ・プロトコルは `INTERNAL` でなければなりません。



詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 PROCESSOR INFO の詳細』節を参照してください。

### ステップ 3: 論理区画の定義

ユーザーが定義したプロセッサが LPAR モードで実行される場合は、「LPAR 定義 (LPAR Definitions)」パネルを使用してその論理区画を定義します。プロセッサ上で物理的に使用できるすべての LPAR は、それらの上で実行されるシステムと一緒に定義する必要があります。

詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 LPARS AND SYSTEMS の詳細』節を参照してください。

### ステップ 4: システムの定義

システムは、項目タイプ SYS や、「新規項目の定義 (Define New Entry)」パネルを使用して定義します。

**注:** SA z/OS 初期化時にハードウェアの検証メッセージの受信を回避するためには、すべてのシステム (カップリング・ファシリティを含む) を定義する必要があります。

詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『新規システムの作成』節を参照してください。

### ステップ 5: システムのプロセッサへの接続

124 ページの『ステップ 2: ポリシー項目 PROCESSOR INFO の使用』で定義したプロセッサと、その論理区画 (プロセッサ・モードを LPAR として設定した場合) に対してこのシステムを接続します。

このシステムはシスプレックスまたは標準グループに接続します (125 ページの『ステップ 6: 論理シスプレックスの定義』および 125 ページの『ステップ 7: 物理シスプレックスの定義』を参照)。

**注:** MVS SYSNAME と Image/ProcOps 名は同じでなければなりません。

**制約事項:** 通常、MVS SYSNAME は数値で始めることができます。ただし、この場合、これは Image/ProcOps 名と同じでなければならないので、数値で開始することはできません。したがって、この命名上の制約は MVS SYSNAME にも適用されます。

### ステップ 6: 論理シスプレックスの定義

グループ・タイプが SYSPLEX の項目タイプ GRP を使用して、「それぞれの」論理シスプレックス (同一 XCF グループ ID 内のシステム) を定義します。

ポリシー SYSPLEX を使用して、実物理シスプレックス名を入力します。複数の SYSPLEX GRP で同一の名前を使用することができます。

ポリシー SYSTEMS を使用して、同一の XCF グループ ID 内にあるすべてのシステムを SYSPLEX GRP に接続できます。システムは 1 つの SYSPLEX GRP にのみ接続できます。

### ステップ 7: 物理シスプレックスの定義

グループ・タイプが STANDARD の項目タイプ GRP を使用して、実物理シスプレックスを定義します。

ポリシー SYSTEMS を使用して、物理シスプレックスのすべてのシステムを STANDARD GRP に接続します。

## 結合データ・セットの連続可用性を使用可能にする

結合データ・セット (CDS) には、シスプレックスの特定の性質を管理する方法に関する情報が含まれています。例えば、SFM CDS (シスプレックス障害管理結合データ・セット) は、システムとシグナル通知接続の失敗、PR/SM (Processor Resource/Systems Manager) 再構成アクションをシステムがどのように管理するかを定義します。

並列シスプレックスの機能において特に重要な結合データ・セットは次の通りです。

- SYSPLEX 結合データ・セット。これは、シスプレックスのシステムと XCF グループを定義します。

## WTO(R) バッファ不足のリカバリーを使用可能にする

- CFRM 結合データ・セット。これは、カップリング・ファシリティとシスプレックスの構造を定義します。

シスプレックスのすべての結合データ・セットについて、代替結合データ・セットを定義することをお奨めします。これらの代替結合データ・セットは、基本 CDS が失敗した時のバックアップとして使用されます。

カスタマイズ・ダイアログによって、CDS のそれぞれのタイプ (例えば、SYSPLEX、ARM、CFRM など) に対して、スペア・ボリュームのシリーズを指定できます。このシリーズの最初のボリュームは、基本代替 CDS の 1 つが失敗した場合に、代替 CDS を作成するために使用されます。

カスタマイズ・ダイアログでは、グループ項目タイプを使用して潜在的代替結合データ・セットを定義します。シスプレックス・グループを選択し、次に「Policy Selection (Policy Selection)」パネルからポリシー項目 SYSPLEX (Sysplex ポリシーを定義する) を選択します。

シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルからポリシー項目 SYSPLEX を選択すると、「Sysplex ポリシー定義 (Sysplex Policy Definition)」パネルが表示されます。

このパネルの説明については、オンライン・ヘルプ、または「IBM Z System Automation 自動化ポリシーの定義」の『ポリシー項目 SYSPLEX の詳細』節を参照してください。

## WTO(R) バッファ不足のリカバリーを使用可能にする

MVS Component 項目タイプ (MVC) について、カスタマイズ・ダイアログの MESSAGES/USER DATA ポリシー項目を使用して、SA z/OS の WTO(R) バッファ不足のリカバリーをカスタマイズできます。

WTO(R) バッファ不足が起こりそうな場合にキャンセルまたは保持するジョブを指定するには、メッセージ ID WTOBUF のコード定義を使用します。キャンセルするように選択したジョブでは、WTO(R) はもう出されません。

メッセージ ID WTOBUF のコード定義を、以下の値を使用して指定します。

### CODE1

キャンセルされるかされない可能性のあるジョブの名前を指定します。

### CODE2

これは、どの要求についてジョブをキャンセルするかしないかを示すために、WTO、WTOR、または\*のいずれかにする必要があります。WTO および WTOR 要求を指定するには、単に\*を使用します。

### CODE3

これは、ブランクにします。

### 返される値

これは、ジョブがキャンセルされることを示す場合は CANCEL、キャンセルされないことを示す場合は KEEP にします。

### 例:

Code 1	Code 2	Code 3	Value Returned
JOB1	WTOR		KEEP
JOB2*	WTO		KEEP
JOB3*	*		CANCEL
JOB4*	*		KEEP
*	*		KEEP

明示的に定義されていないすべてのジョブについてデフォルトの動作をセットアップするには、CODE1=\* および CODE2=\* の指定が必要です。他のすべてのジョブをキャンセルすることを示すには、「戻された値 (Value Returned)」フィールドで CANCEL を指定するか、または KEEP を指定します。

「コード 1 (Code 1)」フィールドには、ジョブ名 \*MASTER\* を指定することはできません。デフォルトの動作が、明示的に定義されていないすべてのジョブをキャンセルするようにセットアップされていても、\*MASTER\* に対しては、それが検査中のジョブ名であれば、キャンセル・コマンドは発行されません。これは、\*MASTER\* がキャンセル不可であるためです。



## システムの除去を使用可能にする

SA z/OS は、シスプレックスから除去されるシステムの保留状態の入出力を解決するのに役立ちます。

詳しくは、[116 ページの『リカバリー・アクション』](#)および [120 ページの『システムの除去の管理』](#)を参照してください。

自動化では、適切な「サポート・エレメント」にコマンドを送信するためにシステムがどこにあるかを知る必要があるため、カスタマイズ・ダイアログを使用してそのハードウェア構成を定義しなければなりません。

BCP 内部インターフェースにより、シスプレックス・メンバーのそれぞれの NetView からハードウェア操作を実行することができます(そのプロセッサ・ハードウェアがサポートしている場合)。詳しくは、「*IBM Z System Automation* 計画とインストール」を参照してください。

**ヒント:** IXC102A の自動化を使用する場合は、自動化ポリシーに定義されたプロセッサ操作関連の IXC102A の自動化がないことを確認してください。同様に、メッセージ IXC102A と IXC402D のプロセッサ操作ベースの自動化の使用を継続する場合は、IXC102A の自動化フラグが使用不可でなければなりません。

### ステップ 1: プロセッサとシステムの定義

プロセッサとシステムは、[124 ページの『ハードウェア関連の自動化を使用可能にする』](#)で説明したように定義する必要があります。

### ステップ 2: アプリケーション・タイプ IMAGE のアプリケーションの定義

アプリケーション・タイプが IMAGE で、([125 ページの『ステップ 4: システムの定義』](#)に定義された) このアプリケーションを表わすシステムの「イメージ名」と同じサブシステム名の新規のアプリケーションを定義するには、項目タイプ APL を使用します。

項目タイプ APL を使用して、ユーザー・システムのポリシー項目 APPLICATION INFO を選択します。「アプリケーション情報 (Application Information)」パネルでは、新規のアプリケーション・タイプ IMAGE を定義できます。詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『アプリケーションのポリシー項目』節を参照してください。

アプリケーションは IMAGE のタイプとして定義されているため、そのジョブ名はデフォルトでサブシステム名に設定され、変更はできません。

「サブタイプ (Subtype)」、 「スケジューリング・サブシステム (Scheduling Subsystem)」、 「JCL プロシージャ名 (JCL Procedure Name)」、 「ARM エレメント名 (ARM Element Name)」と 「WLM リソース名 (WLM Resource Name)」は強制的にブランクになります。

また、次のように、ポリシー項目である APPLICATION INFO の他の定義にもデフォルトがあります。

- 「ジョブ・タイプ (Job Type)」は強制的に NONMVS
- 「モニター・ルーチン (Monitor Routine)」のデフォルトは INGMTSYS (何も指定されない場合)
- 「外部始動 (External Startup)」のデフォルトは ALWAYS (「モニター・ルーチン」が INGMTSYS の場合)
- 「外部シャットダウン (External Shutdown)」のデフォルトは ALWAYS (「モニター・ルーチン」が INGMTSYS の場合)

詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 APPLICATION INFO の詳細』節を参照してください。

### ステップ 3: アプリケーション・グループの定義

アプリケーションの自動化は、アプリケーション・グループを介してそのアプリケーションをシステムにリンクすることによってしか行うことができないため、IMAGE アプリケーション用にアプリケーション・グループを定義する必要があります。

## ステップ 4: IXC102A と IXC402D メッセージの自動化

シスプレックス障害を回避するために、IXC102A と IXC402D メッセージを自動化することができます。

注: メッセージ IXC102A のコマンドとコードを定義する例は次の通りです。

自動化するシスプレックス内の各システムに対して、次の 4 つのハードウェア・コマンドのいずれかを指定できます。何も指定しない場合は、SYSRESET CLEAR が使用されます。

- SYSRESET [CLEAR]
- DEACTIVATE
- ACTIVATE [P(image\_profile\_name)]
- LOAD [P(load\_profile\_name)]

この場合、

### CLEAR

そのストレージが消去されることを示します。

### P

使用するプロファイルを指定します。この名前は 16 文字までの英数字で構成できます。パラメーターを省略した場合は、最後に使用したプロファイルが指定されます。

注: ハードウェア・コマンドの ACTIVATE と LOAD に対して、次の制約事項が適用されます。

この両方のコマンドは、BCP (基本制御プログラム) 内部インターフェースの初期化の時、またはプロセッサ・ハードウェアの待ち状態の時のオペレーター・メッセージなどの非同期イベントを引き起こすことがあるプロセッサ機能を起動します。現在、BCP 内部インターフェースによってこれらのイベントをモニターして制御することはできません。

メッセージ IXC102A と IXC402D のコマンドとコードを定義するには、作成したアプリケーション・タイプ IMAGE で SA z/OS カスタマイズ・ダイアログのポリシー項目 MESSAGES/USER DATA を使用します。「Cmd」列には C または S を入力し、「メッセージ ID (Message ID)」列には IXC102A (また、IXC402D メッセージの自動化では IXC402D) を入力します。詳しくは、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『アプリケーションの MESSAGES/USER DATA ポリシー項目』節を参照してください。また、ここでの定義はメッセージ IXC402D にも適用されます。

Enter を押すと、128 ページの図 26 に示される「CMD 処理 (CMD Processing)」パネルが表示されます。このパネルで、「コマンド・テキスト (Command Text)」列にイメージの有効なハードウェア・コマンドと「パス/選択 (Pass/Selection)」値を指定します。「パス/選択 (Pass/Selection)」値は、「コード処理 (Code Processing)」パネルに指定される「返される値 (Value Returned)」の定義と一致しなければなりません。

Cmd Ps	AutoFn/*	Command Text
ACTCODE		LOAD P(LOADPROF)

図 26. コマンド処理のサンプル・パネル

「メッセージ処理 (Message Processing)」パネルで、k を入力してコードを定義します。128 ページの図 27 に示す「コード処理 (Code Processing)」パネルでは、次のように指定します。

Code 1	Code 2	Code 3	Value Returned
IXC102A	BCPII		ACTCODE

図 27. コード処理のサンプル・パネル

メッセージ IXC102A と IXC402D を自動化する場合は、両方のメッセージに対して Code 1 に IXC102A を入力し、Code 2 に BCPII を入力する必要があります。

## ステップ 5: 自動化テーブルのエントリーの確認

事前定義メッセージの IXC102A および IXC402D のエントリーが自動化テーブルで使用され、自動化オペレーター AUTXCF および AUTXCF2 が定義されていることを確認します (\*BASE サンプル・ポリシーを参照)。

## 長時間エンキュー (ENQ) を使用可能にする

長時間 ENQ を自動化する場合は、以下を定義する必要があります。

- 検査されるリソース (複数の場合あり)
- 長時間 ENQ を検出する時間枠

「ハング」コマンドを自動化する場合は、以下を定義する必要があります。

- モニターするコマンド (1 つ以上) あるいはモニターから除外するコマンド (1 つ以上)
- コマンドが完了を認可される、各コマンド用の時間フレーム、またはコマンドがモニターから除外される場合の除外キーワード
- このコマンドが別のコマンドまたはジョブのブロッカーであると判別された場合は、このコマンドに対してアクションが実行されます。

さらに、以下の定義を行うことができます。

- 長時間 ENQ、「ハング」コマンド、またはコマンド・フラッディングの検出時にキャンセルまたは保持する必要があるジョブの名前
- コマンド・クラスのスナップショット間隔
- ジョブをキャンセルする前に行われるダンプのタイトル
- ダンプするデフォルトのストレージ域
- PARMLIB メンバーによってダンプ仕様を提供する時に使用するシンボル定義

カスタマイズ・ダイアログ内で項目タイプ GRP を使用して、以下のポリシーを定義します。

- リソース定義
- JOB/ASID 定義
- IEADMCxx シンボル
- コマンド定義
- スナップショット間隔の定義

## ステップ 1: リソースの定義

リソースを定義するには、「長時間 ENQ リソース定義 (Long Running ENQ Resource Definition)」パネルを使用します。シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルの「長時間エンキュー・ポリシー (Long Running Enqueue Policy)」セクションからポリシー項目、RESOURCE DEFINITIONS を選択した場合に、このパネルが表示されます。詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 RESOURCE DEFINITIONS の詳細』節を参照してください。

## ステップ 2: ジョブ/ASID 定義の実施

シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルの「長時間エンキュー・ポリシー (Long Running Enqueue Policy)」セクションからポリシー項目、JOB/ASID DEFINITIONS を選択した場合に表示される「長時間 ENQ ジョブ/ASID 定義 (Long Running ENQ Job/ASID Definitions)」パネルを使用します。詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 JOB/ASID DEFINITIONS の詳細』節を参照してください。

## ステップ 3: IEADMCxx シンボルの定義

シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルの「長時間エンキュー・ポリシー (Long Running Enqueue Policy)」セクションからポリシー項目 IEADMCxx SYMBOLS を選択した場合

## 補助記憶域不足のリカバリーを使用可能にする

に表示される「長時間 ENQ IEADMCxx シンボル (Long Running ENQ IEADMCxx Symbols)」パネルを使用します。詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 IEADMCxx SYMBOLS の詳細』節を参照してください。

### ステップ 4: コマンドの定義

コマンドを定義するには、「長時間コマンドの定義 (Long Running Command Definition)」パネルを使用します。シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルの「長時間エンキュー・ポリシー (Long Running Enqueue Policy)」セクションからポリシー項目 COMMAND DEFINITIONS を選択した場合に、このパネルが表示されます。詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 COMMAND DEFINITIONS の詳細』節を参照してください。

### ステップ 5: スナップショット間隔の定義

個々のスナップショット期間を定義するには、「コマンド・フラッディングの定義 (Command Flooding Definition)」パネルを使用します。シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルの「長時間エンキュー・ポリシー (Long Running Enqueue Policy)」セクションからポリシー項目 COMMAND FLOODING を選択した場合に、このパネルが表示されます。詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 COMMAND FLOODING の詳細』節を参照してください。

## 補助記憶域不足のリカバリーを使用可能にする

補助記憶域不足の障害を回避するには、以下を定義するための項目タイプ GRP の SA z/OS カスタマイズ・ダイアログを使用して、ローカル・ページ・データ・セットを事前定義することができます。

- ローカル・ページ・データ・セット
- ジョブ定義

### ステップ 1: ローカル・ページ・データ・セットの定義

シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルの「ローカル・ページ・データ・セット・ポリシー (Local Page Data Set Policy)」セクションからポリシー項目、LOCAL PAGE DATA SET を選択した場合に表示される「ローカル・ページ・データ・セット・リカバリー (Local Page Data Set Recovery)」パネルを使用します。

詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 LOCAL PAGE DATA SET の詳細』節を参照してください。

### ステップ 2: ジョブの処理の定義

シスプレックス・グループ用の「Policy Selection (Policy Selection)」パネルの「ローカル・ページ・データ・セット・ポリシー (Local Page Data Set Policy)」セクションからポリシー項目、JOB DEFINITIONS を選択した場合に表示される「ローカル・ページ・データ・セット・リカバリー・ジョブ定義 (Local Page Data Set Recovery Job Definition)」パネルを使用します。

詳細については、オンライン・ヘルプ、または「*IBM Z System Automation* 自動化ポリシーの定義」の『ポリシー項目 JOB DEFINITIONS の詳細』節を参照してください。

## 共通自動化項目の定義

開始済みタスクとして実行されるユーティリティーと関連した 2 つの定義があります。最初の定義 (一時データ・セット HLQ/TEMPHLQ) では、使用されている自動化状況ファイルの最初の修飾子を置き換えます。

2 つ目の定義 (開始済みタスク・ジョブ名/STCJOBNM) は、自動化で提供されたプロシージャー名と一致しないジョブ名を割り当てた場合に、自動化でスケジュールした開始済みタスク・ジョブの名前の固有の割り当てを可能にします。

一時データ・セット HLQ/TEMPHLQ を定義することをお奨めします。これが定義されていない場合、自動化では自動化状況ファイルの最初の修飾子が使用されます。

シスプレックス用の「Policy Selection (Policy Selection)」パネルからポリシー項目、SYSPLEX を選択した場合に表示される「Sysplex ポリシー定義 (Sysplex Policy Definition)」パネルを使用して、この両方の項目を定義できます。詳細については、オンライン・ヘルプ、または「*IBM Z System Automation 自動化ポリシーの定義*」の『ポリシー項目 SYSPLEX の詳細』節を参照してください。

## 機能を使用するシステムのカスタマイズ

### 追加の自動化オペレーター ID

並列シスプレックスを自動化するには、追加の自動化オペレーター ID を定義する必要があります。

これらの自動化オペレーター定義は、提供されている \*BASE サンプル・ポリシーからインポートできます。

### シスプレックス機能のオン/オフの切り替え

次のマイナー・リソース名を指定するには、SA z/OS カスタマイズ・ダイアログを使用します。

#### CDS

代替 CDS のリカバリーの場合

#### ENQ

次の 4 つの独立したリカバリーの処理を使用可能に設定します。

#### ENQ.CMDFLOOD

ある特定のコマンド・クラスをあふれさせるコマンドの処理を使用可能に設定します。

#### ENQ.HUNGCMD

その他のコマンドの実行完了を妨げるジョブおよびコマンドの処理を使用可能に設定します。

#### ENQ.LONGENQ

長時間 ENQ の処理を使用可能に設定します。

#### LOG

システム・ログのリカバリーの場合

#### LOGGER

システム・ロガーのリカバリーの場合

#### PAGE

補助記憶域不足のリカバリーの場合

#### WTO

WTO(R) バッファ不足のリカバリーの場合

#### XCF

メッセージ IXC102A と IXC402D の自動化の場合

デフォルトで、すべてのリカバリー・アクションが使用可能になります。これらを使用不可にしたい場合は、カスタマイズ・ダイアログ「フラグ自動化の指定 (*Flag Automation Specification*)」を使用して、リカバリー・フラグを「いいえ」に設定します。

注: 自動化リカバリー・フラグは、コマンド INGAUTO を使用して実行時に変更できます。





## 第 12 章 ネットワークの自動化

### 自動化ネットワークの定義プロセス

この節では、SA z/OS に自動化ネットワークを定義するためのステップを要約しています。プロセスの各ステップの詳細は、本書の後半で説明されています。

- 異なるシステム間でのメッセージ転送パスを定義します。これを行うには、以下を定義してください。
  - 1 次フォーカル・ポイント。すべての通知がここへ送信されます。
  - オプションのバックアップ・フォーカル・ポイント。1 次フォーカル・ポイントが使用不可の場合に使用されます。
  - ターゲット・システム。フォーカル・ポイント・システムによりモニターおよび制御されます。
  - システム間のゲートウェイ・セッション。  
ゲートウェイ・セッションを定義する方法については、134 ページの『ゲートウェイ・セッションの定義』を参照してください。
- 自動化ネットワーク構成を反映するように NetView 定義を変更します。詳細は、「*IBM Z System Automation* 計画とインストール」の、ホスト・システムへの SA z/OS のインストール方法に関する章を参照してください。

自動化ネットワークの定義プロセスの例は、「*IBM Z System Automation* 計画とインストール」のホスト・システムへの SA z/OS のインストールに関する章も参照してください。

これらを定義すると、ターゲット・システムからフォーカル・ポイント・システムへのメッセージ転送を可能にするパスが作成されます。

メッセージ転送パスは、以下のトップダウン方式で各システムを定義すると最適に実装されます。

- 1 次フォーカル・ポイント・システム
- バックアップフォーカル・ポイント・システム
- ターゲット・システム

1 次フォーカル・ポイントを最初に定義することで、ターゲット・システムに対して転送がオンになるとすぐに、転送されたメッセージを処理する準備ができるようになります。

自動化ネットワーク内で、メッセージ転送パスが実装されていないシステムがある場合、メッセージはターゲット・システムの通知オペレーターに対して表示されます。メッセージ転送パスが実装されると、通知はフォーカル・ポイント・システムに転送されます。

ターゲット・システムが最初に実装されると、通知を転送する試みが正常に行われず、その通知は NetView ログに記録されるため、追加のオーバーヘッドが生じます。

### SDF フォーカル・ポイント・システムの定義

フォーカル・ポイント・システムおよびバックアップ・フォーカル・ポイント・システムは、カスタマイズ・ダイアログのネットワーク項目タイプを使用して定義します。各システムは、メッセージ転送パスに次のシステムまたはドメインを定義する自動化ポリシー内に、項目を 1 つ持ちます。134 ページの図 28 に、自動化ネットワーク例を示してあります。この例では、1 次フォーカル・ポイント・システムは CHI01 です。バックアップ・フォーカル・ポイントは CHI02 です。



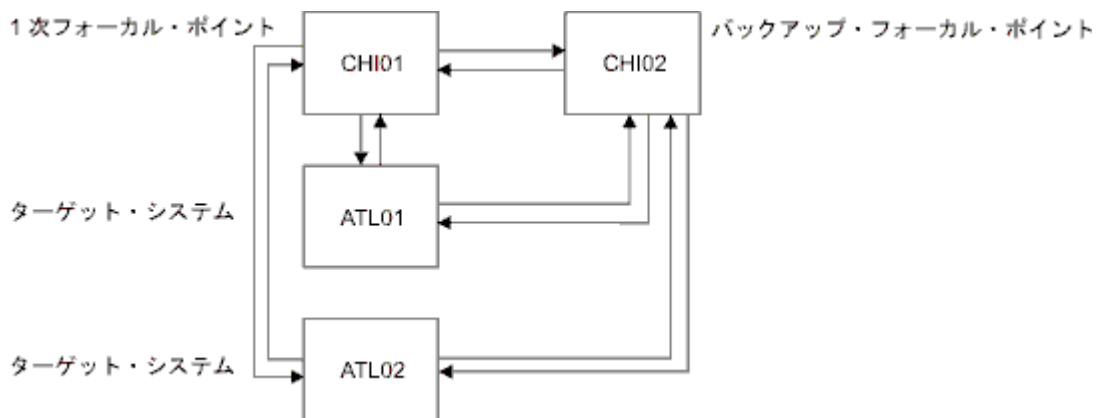


図 28. システムに対するフォーカル・ポイント転送の定義

SA z/OS は常にローカル・システム上のメッセージを表示します。フォーカル・ポイントが定義されている場合は、そのメッセージをフォーカル・ポイントに転送します。

## ゲートウェイ・セッションの定義

### 手順

ゲートウェイ・セッションを定義するには、次のようにします。

1. システムごとに、カスタマイズ・ダイアログの自動化オペレーター ポリシー・オブジェクトでアウトバウンド・ゲートウェイ自動タスク (GATOPER) を定義します。「*IBM Z System Automation 自動化ポリシーの定義*」の『自動化オペレーター項目タイプ』を参照してください。
2. SA z/OS の Network ポリシー・オブジェクトで GATEWAY ポリシー項目を使用して、ネットワーク内の相互に通信する必要があるすべてのシステムのドメインを定義します。「*IBM Z System Automation 自動化ポリシーの定義*」の『ネットワーク項目タイプ』を参照してください。
3. システムのすべてのインバウンドおよびアウトバウンドのゲートウェイ自動タスクで使用されるオペレーター ID を、NetView DSIPARM データ・セット・メンバー DSIOPF に定義します。詳細は、「*IBM Z System Automation 自動化ポリシーの定義*」の、ホスト・システムへの SA z/OS のインストール方法に関する章を参照してください。

## 自動開始 TAF フルスクリーン・セッションの定義

SA z/OS がモニターするアプリケーションについて、端末アクセス機能 (TAF) フルスクリーン・セッションを自動的に確立することができます。これにより、オペレーターは毎日セッションを定義する必要がなくなります。

これらの TAF フルスクリーン・セッションは、ネットワーク ポリシー・オブジェクトについて FULL SESSIONS ポリシー項目で定義されます。

カスタマイズ・ダイアログを使用して TAF フルスクリーン・セッションを定義することに加えて、「*IBM Z NetView カスタマイズ・ガイド*」で概説されているように、NetView プロセスに従って TAF フルスクリーン・セッションをカスタマイズします。

TAF フルスクリーン・セッションがセットアップされると、SA z/OS オペレーター・インターフェースの TAF フルスクリーン・メニューを使用して管理できます。TAF フルスクリーン・セッションの管理について詳しくは、「*IBM Z System Automation ユーザーズ・ガイド*」の『TAF フルスクリーン・セッションの管理』を参照してください。

ネットワーク ポリシー・オブジェクトの FULL SESSIONS ポリシー項目を選択して表示される「フルスクリーン TAF アプリケーション定義」パネルでアプリケーションを定義するには、以下を指定します。

- セッション名、または、TAF フルスクリーン・セッションが確立されるアプリケーションの名前。例えば、TSO など。この名前は、TAF フルスクリーン・メニューのオペレーター・パネルの「説明 (Description)」フィールドに表示されます。この値は、アプリケーション ID に使用されるものと同じでもかまいません。

- アプリケーション ID。この値は、ネットワーク定義を含むライブラリー (VTAMLST)、または、ネットワーク・システム・プログラマーから取得することができます。
- アプリケーションを実行するシステム名。例えば CHI01。これは情報用のみのフィールドであり、TAF フルスクリーン・メニューのオペレーター・パネルの「システム (System)」フィールドに表示されます。

例えば、以下のパネルでは、システム CHI01 の TSO についての TAF フルスクリーン・セッションを定義します。

```

COMMANDS  HELP
-----
AOFPI3E3          Fullscreen TAF Application Definition  Row 1 to 10 of 20
Command ==> _____ SCROLL==> PAGE

Entry Type : Network          PolicyDB Name   : USER_PDB
Entry Name  : FOCAL_NETWORK   Enterprise Name : USER_ENTERPRISE

Enter the applications with which SA z/OS operators can establish TAF
sessions automatically using the operator interface.

Session Name  Application ID  System
  TSO         TAIN1        CHI01
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____
  _____  _____  _____

```

図 29. 「フルスクリーン TAF アプリケーション定義」パネル



## 第 13 章 SA z/OS への VTAM アプリケーションの定義

VTAM アプリケーションでは、アプリケーションが正しく動作するためにアプリケーション・ノードを活動化させる必要があります。これは、通常、アプリケーションを単一システムで実行する場合には問題になりません。しかし、あるシステムから別のシステムへ (MOVE グループまたは SERVER グループを介して) アプリケーションを切り替える場合には、アプリケーションの移動元のシステムからアプリケーション・ノード定義を削除しなければなりません。これを行わないと、アクティブでないアプリケーション・ノードの定義が存在するため、つまり、アプリケーションがノード ACB を開いていないため、ユーザーはアプリケーションにログオンできなくなる可能性があります。

この問題を改善するために、古いシステムからアプリケーション・ノードを削除して、新しいシステム上に作成する必要があります。あいにく、VTAM でノードを削除する唯一の方法は、そのメジャー・ノードを非活動化すること、つまり、そのノードが定義されているメンバーを非活動化することです。

VTAM のより新しいリリースでは、モデル・ノード定義の概念を導入しています。ここでは、メジャー・ノードは、ノード・タイプと、ワイルドカードを含む名前を付けて作成されます。そのタイプのノードがアクセスされると、VTAM は必ず、モデルと一致するよう要求された名前を使用します。次に、モデル定義に基づいて、要求された名前でノードを作成します。そのノードが不要になると、VTAM はそれを削除します。これがアプリケーション・ノードにとって何を意味するかは、そのアプリケーションが実行される可能性のあるネットワーク内のそれぞれの VTAM においてモデル定義を 1 回定義できるということです。その後、アプリケーションが開始し、その ACB を開くと、VTAM はアプリケーションに対し動的にノードを作成します。アプリケーションを移動する場合と同じように、ACB を閉じるとき、VTAM はそのノードを削除し、別のシステム上の別の VTAM が動的にノードを作成します。

VTAM アプリケーションでは、VTAM を再始動する場合、または VTAM アプリケーション・ノードを再活動化する場合に、リカバリー・コマンドの発行を必要とする場合があります。これらのコマンドはサブシステムごとに異なり、次のセクションで説明されているように、Messages/User Data ポリシー項目で指定することができます。

INGVTAM コマンドは、アプリケーションの開始前にそのアプリケーションのメジャー・ノードを活動化する方法、およびアプリケーションがダウンした後メジャー・ノードを非活動化する方法を提供します。この機能を要求するには、PRESTART、REFRESHSTART、および SHUTDOWN FINAL の定義内に INGVTAM コマンドをコーディングする必要があります。また、アプリケーションの実行中に VTAM を再始動しなければならない場合、メジャー・ノードを再活動化する必要があります。これは、VTAM サブシステムの UP Messages/User Data ポリシーに INGVTAM コマンドをコーディングして行うことができます。

### SA z/OS のリカバリーへの VTAM アプリケーション・サブシステムの登録

VTAM アプリケーションのリカバリーの実行を有効にするために、サブシステムを SA z/OS リカバリー・コードに登録する必要があります。

#### このタスクについて

これは、「IBM Z System Automation プログラマーズ・リファレンス」で説明されている INGVTAM コマンドを使用して行います。以下のアプリケーション・ポリシー項目をカスタマイズする必要があります。

#### 手順

1. STARTUP ポリシー内の PRESTART ポリシーには、メジャー・ノードのリストを活動化する INGVTAM コマンドに、少なくとも 1 つの NORMAl 開始項目が必要です。例として次のコマンドを使用できます。

```
INGVTAM &SUBSAPPL REQ=ACTIVATE MAJNODE=(majnode1,majnode2,...)
```

ここで、majnoden は VTAM アプリケーションのメジャー・ノードを表します。各メジャー・ノードは、サブシステム事前開始コマンドが発行されると、VTAM に対してアクティブに変更されます。VTAM アプリケーション・サブシステムが使用するマイナー・ノードを含むメジャー・ノードは 1 つのみであると考えられることに注意してください。

- SHUTDOWN ポリシー内の FINAL 定義は、SA z/OS VTAM アプリケーション・リカバリーからサブシステムの登録を解除するために使用されます。ポリシーで INGV TAM REQ=DEACTIVATE コマンドを使用します。例えば、次のようなコマンドを使用します。

```
INGVTAM &SUBSAPPL REQ=DEACTIVATE
```

DEACTIVATE 要求は、REQ=ACTIVATE で登録されている各メジャー・ノードについて、NET の非アクティブ変更を実行します。メジャー・ノードが、そのメジャー・ノードの登録も済んでいる他のサブシステムによって共有される場合、変更は行われません。登録された最後のサブシステムが INGV TAM REQ=DEACTIVATE を発行すると、メジャー・ノードは非アクティブに変更されます。これに対する唯一の例外は、メジャー・ノードが、ワイルドカードを含むモデル・リソースをノード定義内に持っている場合です。この場合、メジャー・ノードは非活動化されません。

- REFRESHSTART 定義には、STARTUP ポリシー内の PRESTART 定義と同じ定義が必要です。このポリシー項目は、SA z/OS VTAM アプリケーションのリカバリーへサブシステムを再登録するために使用します。
- VTAM サブシステムを再始動するときに発行されるコマンドを入力します。通常、これらのコマンドは、サブシステムが VTAM と通信するために使用する VTAM ACB を再オープンします。
- オプションで、サブシステムに対する関係を入力して、事前開始コマンドが確実に VTAM の起動中のみ発行されるようにします。必要な関係は以下のとおりです。

```
PrepAvailable(WhenAvailable),Passive,Weak -> VTAM/APL/=
```

ここで、VTAM は VTAM サブシステムの名前であり、サポートしているリソースです。Passive は、VTAM が起動するまで関係を強制的に待機させます。Weak は、サポートしているリソースの状況のみを検査するよう指定します。

- また、VTAM に対する UP メッセージは、以下のコマンドを含む必要があります。

```
INGVTAM REQ=ACTIVATE
```

## タスクの結果

REQ=ACTIVATE を指定して、定位置サブシステムなしで INGV TAM を実行する場合、INGVTAM を通じて前に登録されたすべてのサブシステムが検索され、それらのメジャー・ノードに対して Vary NET ACT コマンドが発行されます。これが完了すると、INGVTAM は、それらのサブシステムの USER MESSAGE VTAMUP に指定されているすべてのポリシー・コマンドを実行します。

## 第 14 章 z/OS システムのシャットダウン

このセクションでは、システム・シャットダウンに関する考慮事項について説明します。SA z/OS は、システムのシャットダウンに使用される時に、SA z/OS ポリシーで定義されているすべての SA z/OS リソース (アプリケーション (APL)、アプリケーション・グループ (APG)、モニター・リソース (MTR) など) をシャットダウンします。これには、APL 定義によって表される自動化マネージャー (AM) や NetView 自体も含まれます。AM と NetView 自体はどちらも、最後に終了されるアプリケーションになるように意図されています。

### システム・シャットダウン・プロセス

システム・シャットダウンは以下の概念的な順序で構成されています。

1. ガード・アプリケーションのシャットダウン。これは、システム・シャットダウンが終了したときの定義を行う特定アプリケーションです。最も一般的なガード・アプリケーションは JES サブシステムです。ガード・アプリケーションの終了前に、お客様が行う準備手順が存在することがあります。このフェーズはフェーズ 0 と呼ばれます。

ガード・アプリケーションの終了後に、他の定義済みアプリケーションが依然として存在することがあります。そのようなアプリケーションは、フェーズ 1 という後続のフェーズで終了させる必要があります。ガードアプリケーションの後で終了させるようなサブシステムの例としては、RRS、VLF、LLA などがあります。

2. OMVS アプリケーションのシャットダウン。AM は OMVS プロセスです。AM が存在する場合、OMVS の終了はブロックされます。そのため、最後のローカル 1 次自動化マネージャーは OMVS 自体の前に終了されなければなりません。

OMVS サブシステムの終了後に終了すべき他の APL があります。このような APL のシャットダウンや、さらなるシステム・クリーンアップ・アクションは、いわゆるフェーズ 2 で行われます。OMVS の後で終了させるようなサブシステムの例としては、SMF 記録を表す疑似アプリケーションや、データ・セットおよび SMF 記録データの暗号化を担当するデータ暗号化アプリケーション Integrated Cryptographic Service Facility (ICSF) などがあります。

リンクされた MVS コンポーネント項目 (MVC) の MESSAGES/USER DATA ポリシー内でメッセージ ID SYSTEM\_SHUTDOWN がホスト・コマンド定義に対して定義されます。フェーズ 0、1、および 2 に属するコマンドはそこで PHASE0 コマンド、PHASE1 コマンド、および PHASE2 コマンドとして定義されます。

システムのシャットダウンは \*BASE アドオン・ポリシーの定義に基づきます。AM、NetView、および OMVS サブシステムを表すアプリケーションに対する既存の SA z/OS ポリシー定義は前提条件です。

### 異なるシステム・シャットダウン・シナリオ

異なる 2 つの SA z/OS 環境が存在します。それぞれの環境に、z/OS 環境をシャットダウンするための独自メカニズムがあります。

1. 純粋な SA z/OS 環境で z/OS システムをシャットダウンする。

この環境ではガード・アプリケーション定義はサポートされません。

システム・シャットダウンの場合は、システム自体 (ローカル SYG リソース) が STOP 要求を受け取りません。OMVS サブシステムの前に終了できるすべてのサブシステム (存続するローカル 1 次自動化マネージャー・アプリケーションを含む) が終了させられます。

定義済み SYSTEM\_SHUTDOWN メッセージの PHASE2 コマンドが存在する場合は、NetView APL のシャットダウンが遅延させられます。その時点で、定義済み PHASE2 コマンドが順番に同期実行されます。

PHASE2 コマンドが存在しない場合、Netview アプリケーションは即時に終了します。

このシャットダウン・シナリオについて詳しくは、[140 ページの『純粋な z/OS 環境で z/OS システムをシャットダウン』](#)を参照してください。

2. GDPS 環境で z/OS システムをシャットダウンする。

この環境では、ガード・アプリケーションはいわゆる GDPS STOPAPPL として認識されます。GDPS STOPAPPL を表す APL はシャットダウン要求を受け取ります。後続のシステム・シャットダウンは以下のリスト順に行われます。

- a. 定義済み PHASE0 コマンドの実行。
- b. GDPS STOPAPPL の終了。
- c. 定義済み PHASE1 コマンドの実行。
- d. OMVS アプリケーションの終了。
- e. 定義済み PHASE2 コマンドの実行。
- f. 制御が GDPS に移り、LPAR がシスプレックスから削除される。

このシャットダウン・シナリオについて詳しくは、[141 ページの『GDPS 環境での z/OS システムのシャットダウン』](#)を参照してください。

異なる SA z/OS バージョンの共存に関する考慮事項については、「[IBM Z System Automation 計画とインストール](#)」のトピック『[前のリリースと SA z/OS 4.2 の共存](#)』を参照してください。

## 純粋な z/OS 環境で z/OS システムをシャットダウン

SA z/OS による z/OS システムのシャットダウンは、INGREQ ALL コマンドによって実行されます。

OMVS サブシステムの後で終了しなければならない定義済みサブシステムがある場合は、最終シャットダウン処理で別のフェーズ (**フェーズ 2**) が存在します。このフェーズは、MVS コンポーネント項目タイプについて MESSAGES/USER DATA ポリシーで特別メッセージ ID SYSTEM\_SHUTDOWN を使用して定義されます。

フェーズ 2 は、すべてのローカル自動化マネージャー (PAM および SAM)、ならびに OMVS が終了された後で開始されます。指定できるコマンドは、NetView コマンド、または NetView を使用して発行された z/OS コマンドのみです。例えば、INGSHRES ICSF コマンドなどです。

注:

- 最後のローカル 1 次自動化マネージャーは常に SA z/OS によって自動的にシャットダウンされます。
- JES3 環境で実行する際には、コマンド MVS F BPX0INIT, SHUTDOWN=FORKS を、JES3 SHUTINIT フェーズでルーチン INGRYSHU を介してトリガーすることを検討してください。INGRYSHU は、システム・シャットダウンが検出されたときのみコマンドを発行します。同様に、コマンド MVS F BPX0INIT, SHUTDOWN=FILESYS を JES3 SHUTFINAL フェーズでトリガーできます。

このシナリオは、提供されているベスト・プラクティス・ポリシー \*BASE に基づいています。詳しくは、MVC 項目 SYSTEM\_SHUTDOWN を参照してください。

## 定義の例

z/OS システムをシャットダウンするアクションは、MVS コンポーネント項目タイプについて MESSAGES/USER DATA ポリシーで疑似メッセージ SYSTEM\_SHUTDOWN を使用して定義されます。戻りコード検査がオンになっていて 0 以外のコードが返される場合、SA z/OS は定義済みコマンド・チェーンの実行を即時に停止します。そのような場合、NetView アプリケーションの終了は、さらなるエラー分析のためにスキップされます。

Cmd	Ps/Select	AutoFn/*	コマンド・テキスト
	PHASE2	*	INGSHRES SMF_REC
	PHASE2	*	INGSHRES ICSF

140 ページの表 23 に、疑似メッセージ SYSTEM\_SHUTDOWN で定義される定義の例を示します。

ローカル・システム上の 1 次および 2 次の自動化マネージャー (PAM および SAM) は、SA z/OS によって自動的にシャットダウンされます (ただし、それらの自動化マネージャーが別のシステムに移行されていない



場合)。SA z/OS ポリシー定義によってまだ開始されていない場合、OMVS も SA z/OS によって自動的にシャットダウンされます。

INGSHRES コマンドにより、指定のローカル・アプリケーションに対してポリシー定義のシャットダウン・コマンドが起動されます。そのシャットダウン・コマンドが発行された場合、INGSHRES は、影響を受けるリソースの終了を待ちます。コマンド・チェーンが順番に実行されます。

INGSHRES SMF\_REC コマンドを例にします。

アプリケーション SMF\_REC の場合は、SHUTDOWN ポリシーに、SMF 記録を停止する定義済みコマンド MVS Z EOD があります。SMF 記録が停止されたことを示すメッセージが自動化テーブルでトラップされます。コマンド「INGSMFMP AUTODOWN」が自動化テーブルから呼び出されて、アプリケーション SMF\_REC が状況 AUTODOWN に設定されます。

SMF 記録が正常に停止されると、「INGSHRES ICSF」が発行され、同じように、データ暗号化を担当するアプリケーションが停止されます。すべての PHASE2 コマンドが処理された後で、NetView アプリケーションが終了します。

INGSHRES コマンドおよび INGSMPMP コマンドについて詳しくは、「IBM Z System Automation プログラムズ・リファレンス」を参照してください。

## GDPS 環境での z/OS システムのシャットダウン

SA z/OS では、z/OS システムのシャットダウンを、GDPS 実稼働環境で INGREQ ALL コマンドを使用するかどうか、または、GDPS 制御システムから行うことができます。

どのような場合でも SA z/OS は GDPS STOPAPPL に対してシャットダウン要求を実行します。STOPAPPL リソースが拡張自動化オプション (AAO) AOF\_AAO\_SHUTDOWN\_STOPAPPL で定義されます。GDPS STOPAPPL および AAO AOF\_AAO\_SHUTDOWN\_STOPAPPL について詳しくは、[204 ページの『読み取り/書き込み変数』](#)を参照してください。

最終シャットダウン処理には、MVS コンポーネントの項目タイプについて MESSAGES/USER DATA ポリシー項目で特別メッセージ ID SYSTEM\_SHUTDOWN を使用して定義される以下の 3 つの異なるフェーズがあります。

### フェーズ 0

このフェーズは、GDPS STOPAPPL パラメーター (STOPAPPL リソース) で識別されるリソースをシャットダウンする前に開始します。このフェーズでは、実際のシステムのシャットダウンが開始される前に任意のアクションを実行できます。

### フェーズ 1

このフェーズは、GDPS STOPAPPL パラメーター (STOPAPPL リソース) で識別されるリソースが終了したときに開始します。このフェーズでは、依然として存在するすべてのサブシステムを終了させるために、追加の INGREQ 停止コマンド、または他の任意の NetView コマンドを指定できます。

### フェーズ 2

このフェーズは、すべてのローカル自動化マネージャー (PAM および SAM) ならびに OMVS の終了後に開始されます。NetView コマンドまたは z/OS コマンドのみを指定できます。例えば、INGSHRES SMF\_REC コマンドなどです。

#### 注:

1. OMVS およびすべてのローカル自動化マネージャーは、常に SA z/OS によって自動的にシャットダウンされます。PHASE1 または PHASE2 で、OMVS または自動化マネージャーに終了コマンドを指定しないでください。JES3 環境で実行する際には、コマンド MVS F BPXOINIT,SHUTDOWN=FORKS を、JES3 SHUTINIT フェーズでルーチン INGRYSHU を介してトリガーすることを検討してください。INGRYSHU は、システム・シャットダウンが検出されたときのみコマンドを発行します。同様に、コマンド MVS F BPXOINIT,SHUTDOWN=FILESYS を JES3 SHUTFINAL フェーズでトリガーできます。
2. NetView アドレス・スペースがまだ存在していることに注意します。また、システムをシスプレックスから取得する準備が整っていることを GDPS に伝えるために NetView アドレス・スペースが存続していなければならないことに留意します。

3. NetView がその CLOSE コマンドを呼び出さずに終了した場合は、最新メッセージをそのアクティブな Canzlog に自動的にアーカイブすることはできません。このようなメッセージをアーカイブするには、フェーズ 2 で CANZLOG CUE コマンドを使用します。

このシナリオは、提供されているベスト・プラクティス・ポリシー \*BASE および \*GDPS に基づいています。詳しくは、\*GDPS ベスト・プラクティス・ポリシー内の MVC 項目 GDPS\_SYSTEM\_SHUTDOWN を参照してください。

## 定義の例

GDPS 内で z/OS システムをシャットダウンするために実行するアクションは、MVS コンポーネント項目タイプについて MESSAGES/USER DATA ポリシー項目で SYSTEM\_SHUTDOWN メッセージ ID を使用することで定義されます。このようなアクションには、影響を受ける GDPS STOPAPPL の従属ツリーからリソースをシャットダウンしたり、ファイル・システムをシャットダウンしたりするように SA z/OS に指示するアクションを含めることができます。

Cmd	Ps/Select	AutoFn/*	コマンド・テキスト
	PHASE1		INGREQ RACF/APL/&SYSNAME. REQ=STOP PRECHECK=NO REMOVE=SYSGONE VERIFY=NO OUTMODE=LINE
	PHASE1		INGREQ RRS/APL/&SYSNAME. REQ=STOP PRECHECK=NO REMOVE=SYSGONE VERIFY=NO OUTMODE=LINE
	PHASE1		INGREQ GDPS_ALL/APG/&SYSNAME. REQ=STOP PRECHECK=NO REMOVE=SYSGONE VERIFY=NO OUTMODE=LINE
	PHASE1		INGREQ VLF/APL/&SYSNAME. REQ=STOP PRECHECK=NO REMOVE=SYSGONE VERIFY=NO OUTMODE=LINE
	PHASE1		INGRCHCK JES2/APL/&SYSNAME. OBSERVED=(SO HA)
	PHASE1		INGRCHCK VLF/APL/&SYSNAME. OBSERVED=(SO HA)
	PHASE2		PIPE CC LIST STATUS=CANZLOG   TAKE 1  NLOC / subsystem *none* /   EDIT /CANZLOG CUE/   NETV   EXPOSE TOTRAP   CON
	PHASE2	*	INGSHRES SMF_REC
	PHASE2	*	INGSHRES ICSF

142 ページの表 24 に、PHASE1 でリストされたリソースに対する停止投票を順番に配置する GDPS\_SYSTEM\_SHUTDOWN 項目の定義例を示します。リソース・シャットダウンの望ましい完了は、並列に処理されます。PHASE1 シーケンスの最後に指定された INGRCHCK コマンドは、指定されたリソースへの停止要求の完了を待ちます。INGRCHCK コマンドについて詳しくは、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照してください。

同期が必要な場合は、INGREQ コマンドの FDBK オプションを使用すると、適切なサブシステムがシャットダウンするまで待つことが可能になります。FDBK=WAIT オプションにより、INGREQ 停止コマンドが順次処理されます。このようにして、このオプションはシャットダウン・プロセスの速度を落とします。

ローカル・システム上の 1 次およびすべての 2 次自動化マネージャー (PAM および SAM) は、他のシステムに移行されない限り、SA z/OS によって自動的にシャットダウンされます。OMVS も、SA z/OS によって自動的にシャットダウンされます。

SMFDUMP に対してリカバリー・フラグを設定する処理、CANZLOG を停止する処理、SMF 記録 (APL SMF\_REC) 自体を停止する処理、続いてデータ暗号化 (APL ICSF) を停止する処理は、PHASE2 コマンド・チェーンの指定どおりに同期実行されます。

戻りコード検査がオンになっていて 0 以外のコードが返される場合、SA z/OS は定義済みコマンド・チェーンの実行を即時に停止します。

## INGREQ の考慮事項

何も指定しない場合は、デフォルトの `OVERRIDE=(TRG FLG SUS)` および `TYPE=NORM` が使用されます。必要であれば、INGREQ ユーザー出口 `AOFEXC01` によってデフォルトをオーバーライドできます。

## システム・シャットダウン時の System Recovery Boost の自動活動化

IBM Z System Recovery Boost は、IBM z15 (z15) と一緒に使用できる新しい機能です。これは、リカバリー活動や保守活動による全体のダウン時間を短縮するために IPL 時やシャットダウン時に追加プロセッサ・キャパシティーを一時的に活用します。この新機能には、z/OS V2.3 と OA57849、または z/OS V2.4 が必要となります。IPL 時には、IEASYSxx における BOOST システム・パラメーターが NONE に設定されている場合を除き、システムはブーストに自動的に関与できます。シャットダウン時には、オペレーターは、z/OS に付属する新規開始プロシージャ IEASDBS を開始することでブーストを明示的に活動化する必要があります。System Recovery Boost について詳しくは、[z/OS の資料](#)を参照してください。

SA z/OS はシステムのシャットダウン時に System Recovery Boost を自動的に活動化できます。SA z/OS は、運用上の前提条件が満たされているかどうかを検査し、運用上の前提条件が満たされていれば開始プロシージャ IEASDBS を開始します。具体的には、SA z/OS は z/OS V2.3 以上での実行かどうかを検査し、IEASYSxx においてブーストが有効かどうかを検査し、システムがまだブースト・モードで作動していないことを確認します。すべての条件が満たされた場合にのみ IEASDBS が開始されます。SA z/OS は、z15 上での実行かどうかは検査しません。

System Recovery Boost がシステム・シャットダウン時に自動的に活動化されるべきではない場合、System Recovery Boost がオペレーティング・システムで有効になっている場合であっても、管理者は INGREQ ALL 要求で新規 INGREQ パラメーター `BOOST=NO` を渡すことができます。管理者は、システム・シャットダウン時の System Recovery Boost を永続的に無効にする場合は、グローバル変数 `INGREQ_BOOST` を NO に設定します。

また、システムが GDPS でシャットダウンされる場合に運用上の前提条件が満たされているのであれば、SA z/OS は System Recovery Boost を自動的に活動化します。管理者は、System Recovery Boost の自動活動化を回避する場合は、拡張自動化オプション `AOF_AAO_SHUTDOWN_BOOST` を NO に設定します。

IEASDBS は System Recovery Boost を 1 回のみ活動化します。オペレーターが SA z/OS によってシャットダウンを要求する前に、これが手動で開始されている場合、SA z/OS はそれを検出し、そのプロシージャを再び開始することはありません。オペレーターが z/OS を使用してシャットダウンを要求した後で、z/OS が既にそのプロシージャを自動的に開始しているにもかかわらず、このプロシージャが手動で開始されると、エラーが発生したが追加アクションは必要ないというメッセージがシステム・コンソール上に表示されます。



## 第 15 章 WTOR 処理

Z System Automation は、WTOR (要応答オペレーター宛メッセージ) を受け取ると、そのメッセージに自動的に応答するか、あるいは、そのメッセージをリカバリーに使用する場合またはメッセージを発行したサブシステムをシャットダウンするために使用する場合、そのメッセージを保管します。後で使用するために保管される WTOR は、未解決の WTOR と呼ばれます。

### WTOR の処理フロー

システムで発行される WTOR はすべて NetView に転送する必要があります。そうでなければ、SA z/OS は、WTOR を処理できません。

これを行うには、項目タイプ MSG のポリシー BUILD CONTROL で「すべての WTOR を AT に送付 (Route all WTORs to AT)」に YES を指定します。この項目タイプとポリシーについて詳しくは、「IBM Z System Automation 自動化ポリシーの定義」を参照してください。あるいは、メッセージ改訂テーブル内の以下の定義を追加する必要があります。

```
UPON (OTHERMSG)
  SELECT
  * Ensure all WTORs are being automated
    WHEN (WQE SUBSTR 345 C2D ^= "+0")
      REVISE("Y" AUTOMATE)
    OTHERWISE
  END
```

着信 WTOR は、NetView 自動化テーブル (AT) によって処理されます。これにより、処理目的に従ってコマンドがトリガーされます。

呼び出される自動化ルーチン	処理目的
ISSUEACT	<ol style="list-style-type: none"> <li>定義されたコマンドまたは応答 (あるいはその両方) をサブシステムに発行します。</li> <li>WTOR がまだ応答されていない場合は、その WTOR を保管します。</li> </ol>
ACTIVMSG、HALTMSG、TERMMSG	<ol style="list-style-type: none"> <li>WTOR を発行したサブシステムの状況を更新します。</li> <li>定義されたコマンドまたは応答 (あるいはその両方) を発行します。</li> <li>WTOR がまだ応答されていない場合は、その WTOR を保管します。</li> </ol>
INGMON	<ol style="list-style-type: none"> <li>定義されたコマンドまたは応答 (あるいはその両方) をモニター・リソースに発行します。</li> <li>WTOR がまだ応答されていない場合は、その WTOR を保管します。</li> </ol>
OUTREP	WTOR を保管します。

コマンド (OUTREP 以外) は、AT の同義語 %AOFOPGSSOPER% に定義された最初のアクティブ・タスクに送られます。したがって、通常、これらの汎用ルーチンは、WTOR を発行したサブシステムの作業オペレーターに送られます。これは、WTOR に関連付けられたジョブ名に基づいて実行されます。

SA z/OS に定義されていないサブシステムからの WTOR、あるいは MVS コンポーネントのサブシステムからの WTOR を処理する自動化ルーチンは、WTOR がそのメッセージ ID に基づいて割り当てられているタスクに送られます。

OUTREP コマンドは、AT の同義語 %AOFOPSYSOPER% に定義された最初のアクティブ・タスクに送られます。

## 着信 WTOR に対応するアクション

以下のように、MESSAGES/USER DATA 自動化ポリシー項目を使用して、アプリケーション、モニター・リソースおよび MVS コンポーネントの着信 WTOR に対して SA z/OS が実行すべき応答を定義することができます。

- CMD アクション (場合によっては CODE アクションとの組み合わせ) を使用し、着信 WTOR に対応して発行されるコマンドを定義する。
- REP アクション (場合によっては CODE アクションとの組み合わせ) を使用し、着信 WTOR に対応して即時に発行される応答を定義する。
- AUTO アクションを使用し、WTOR を発行したサブシステムの状況を変更する状況メッセージとして着信 WTOR を定義する。

定義されたアクションに応じて適切なコマンドを呼び出す、NetView 自動化テーブル・ステートメントが作成されます。

CODE 定義を使用してアクションを定義した場合は、作成された自動化テーブル・ステートメントを OVR アクションで補足して、WTOR から抽出する可変情報とそのデータをコード値として関連したコマンドに渡す方法を、SA z/OS に通知する必要があります。

アクションが何も定義されていない WTOR は、OUTREP を介して、SA z/OS により保管されます。この目的のために、適切な自動化テーブル・ステートメントが作成されます。

## SA z/OS による WTOR の保管方法のカスタマイズ

SA z/OS は、まだ応答のないすべての未解決の WTOR をトラッキングし、それらを SDF を介して表示します。

これらの未解決の WTOR には、以下があります。

- 始動時にアプリケーションによって発行され、重大なオペレーター通信およびシャットダウンのインターフェースを提供する永続的に未解決の WTOR
- SA z/OS 自動化ポリシーで応答が定義されていない WTOR
- SA z/OS の初期化の前に、または SA z/OS のダウン時間中に発行された WTOR

自動化ポリシーを使用して、未解決の WTOR の重大度、および 1 次 WTOR と 2 次 WTOR の識別を可能にする優先順位を定義することができます。

### 重大度

WTOR の重大度によって、SDF での WTOR のカラーが決まります。重大度には、以下の値を指定できます。

#### **NORMAL**

通常のメッセージであり、問題を示しているわけではありません。

#### **UNUSUAL**

問題を示している可能性があるメッセージです。

#### **IMPORTANT**

重大な問題を示すメッセージです。

#### **CRITICAL**

危険な問題を示すメッセージです。

#### **IGNORE**

SA z/OS によって無視されるメッセージです。

### 優先順位

1 次 WTOR は保管され、オペレーター通信に使用するために、および、その 1 次 WTOR を発行したサブシステムをシャットダウンするために、後に使用することができます。一方、2 次 WTOR は、即時に応答があるか、または SDF に表示するために保管することができます。

このカスタマイズは、WTOR のメッセージ ID の MESSAGES/USER DATA ポリシー項目でコード定義を使用して実行できます。詳細については、「*IBM Z System Automation プログラマーズ・リファレンス*」に記載されている OUTREP コマンドの説明を参照してください。



## 1 次 WTOR の処理

SA z/OS が 1 次 WTOR の受信後にそれに直ちに応答しないようにするには、応答は、1 次 WTOR のメッセージ ID に対して直接定義しないようにします。代わりに、1 次 WTOR への応答の発行は、他のメッセージまたは実行されたコマンドによって起動します。

したがって、据え置かれる 1 次 WTOR への応答は、これらの起動側メッセージの ID に対して定義します。そうしない場合、発行される応答は、事前判別されたメッセージ ID に対して提供されます。例えば、シャットダウン中に発行される応答を定義するには、SHUTDOWN 自動化ポリシー項目を使用します。

保管された、サブシステムに対する 1 次 WTOR の応答 ID は、オペレーター通信またはこのサブシステムのシャットダウンに使用できます。

SA z/OS が応答を発行してサブシステムと通信する必要があるが、そのサブシステムに関する未解決の WTOR がまだ保管されていない場合は、RETRY オプションを使用して必要な WTOR を待機します。

1 つのメッセージ ID に対して、同じパスまたは選択オプションを持つ複数の応答を定義できます。これらの応答は、着信 1 次 WTOR のシーケンスへの応答に使用できます。

### 例

始動時に、アプリケーション ABCAPPL によってメッセージ ABC123D が、永続的な、未解決の WTOR として発行され、SA z/OS は、そのメッセージをこのアプリケーション用の 1 次 WTOR として保管します。アプリケーションの存続期間中に、特定のシチュエーションでメッセージ ABC789I が発行されると、このアプリケーションの永続的な、未解決の WTOR ABC123D に対して応答が発行される必要があります。この応答の定義には、アプリケーションのメッセージ ID ABC789I の MESSAGES/USER DATA 自動化ポリシー項目を使用します。

147 ページの図 30 に示すように、アプリケーションによってメッセージ ABC789I が発行されると、SA z/OS は、永続的な、未解決の WTOR の応答 ID を検索し、コマンド **MVS R 117,ABC RESTORE** を実行します。

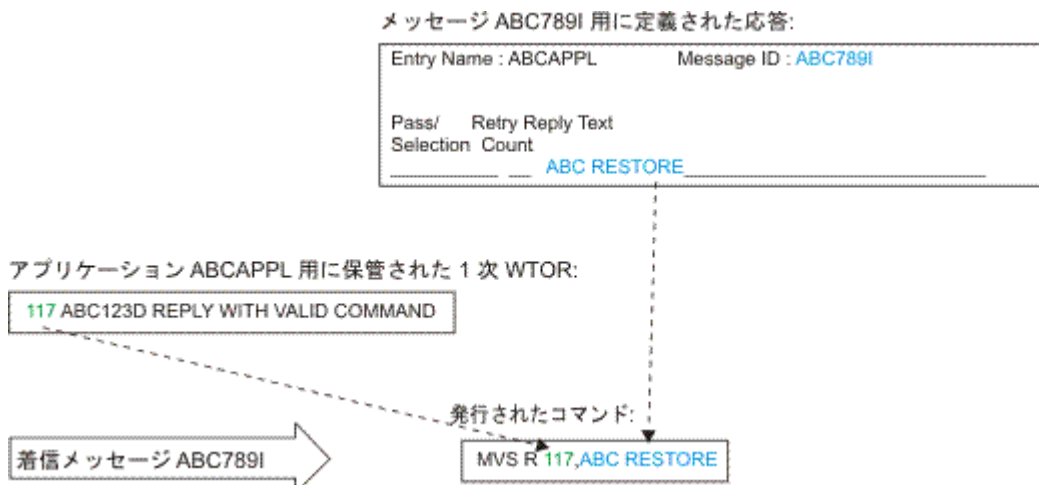


図 30. 1 次 WTOR の処理例

### 制約事項

サブシステムの未解決の 1 次 WTOR の応答 ID は、先行ゼロの付かないブランクで分離されたリストで SA z/OS に保管されます。このストレージは、255 バイトに制限されます。この制限に到達すると、追加の着信 1 次 WTOR の応答 ID は無視されます。

### 使用上の注意

着信 WTOR を保管する場合、メッセージ ID WTOR のコード定義の検索が、WTOR を発行したサブシステムの項目内で最初に実行されます。

サブシステム自体を自動化ポリシー内で検出できない場合、または検索されるコード定義がサブシステムで検出されない場合、それらは、MVSESA 項目下で検索されます。永続的な未解決の応答を持つ IMS また



は NetView などのサブシステムの場合、MVSESA ではなく、サブシステム 項目自体にコード定義を指定する必要があります。これにより、自動化ポリシー内での検索が削減され、パフォーマンスが向上します。

## 第 16 章 SA z/OS ユーザー出口

カスタマイズ・ダイアログでは扱わないユーザー固有のアクティビティーを可能にするために、SA z/OS は、次のようなクラスのユーザー出口をサポートしています。

- **カスタマイズ・ダイアログ出口。** カスタマイズ・ダイアログを使用して作動する際に、特定のフェーズの間に呼び出すことができます。 [149 ページの『カスタマイズ・ダイアログ 出口』](#)を参照してください。
- **初期化出口。** SA z/OS 初期化の開始時に、メッセージ AOF603D が出される前に呼び出されます。 [152 ページの『初期化出口』](#)を参照してください。
- **静的出口。** SA z/OS の処理中の固定点で呼び出されます。 [156 ページの『静的出口』](#)を参照してください。
- **フラグ出口。** SA z/OS が自動化フラグを評価する必要があるときに呼び出します。 [157 ページの『フラグ出口』](#)を参照してください。
- **コマンド出口。** 特定のコマンドの処理中に呼び出すことができます。 [160 ページの『コマンド出口』](#)を参照してください。

サンプル出口のコピーおよび更新については、「*IBM Z System Automation* 計画とインストール」を参照してください。

さらに、SA z/OS には、出口同様の方法で動作する機能がいくつかあります。 [165 ページの『疑似出口』](#)を参照してください。

### カスタマイズ・ダイアログ 出口

SA z/OS は、カスタマイズ・ダイアログを処理する一方で、特定フェーズ中に呼び出すことができる一連の以下のユーザー出口を提供します。

- [149 ページの『BUILD 処理用のユーザー出口』](#)
- [150 ページの『COPY 処理用のユーザー出口』](#)
- [151 ページの『DELETE 処理用のユーザー出口』](#)
- [151 ページの『CONVERT 処理用のユーザー出口』](#)
- [151 ページの『RENAME および IMPORT 機能のユーザー出口』](#)
- [152 ページの『その他の機能用のユーザー出口』](#)

[152 ページの『カスタマイズ・ダイアログ出口の呼び出し』](#)に、ユーザー出口を活動化する方法についての説明があります。

### BUILD 処理用のユーザー出口

自動化制御ファイルのビルド処理用に以下のユーザー出口が提供されます。

- **INGEX10:** 自動化制御ファイル・ビルド機能を開始する前に呼び出されます。この出口は、ビルド処理がカスタマイズ・ダイアログから開始されたときにのみ使用できます。
- **INGEX01:** 自動化制御ファイル・ビルド機能を開始する前に呼び出されます。この出口は、ビルド処理がカスタマイズ・ダイアログ、カスタマイズ・ダイアログ経由で実行依頼されたバッチ・ジョブ、またはカスタマイズ・ダイアログとは無関係に実行依頼されたバッチ・ジョブから開始されたときに使用できます。

カスタマイズ・ダイアログで BATCH の BUILD モードが選択されていると、バッチ・ジョブの JCL が実行依頼され、ジョブが実行を開始すると、自動化制御ファイル・ビルド機能がバッチで開始される前に INGEX01 が呼び出されます。

- **INGEX02:** これは、構成ファイルのビルド終了後に呼び出されます。この出口は、BUILD 処理がカスタマイズ・ダイアログ、カスタマイズ・ダイアログ経由で実行依頼されたバッチ・ジョブ、またはカスタマイズ・ダイアログとは無関係に実行依頼されたバッチ・ジョブから開始されたときに使用できます。

以下のパラメーターは、INGEX01 出口と INGEX02 出口の両方に、コンマで区切って渡されます。

- Parm1 = PolicyDB 名
- Parm2 = エンタープライズ名
- Parm3 = BUILD 出力データ・セット
- Parm4 = 項目タイプ (またはブランク)
- Parm5 = 項目名 (またはブランク)
- Parm6 = BUILD タイプ (MOD/ALL)
- Parm7 = BUILD モード (ONLINE/BATCH)
- Parm8 = 構成 (0=NORMAL/1=ALTERNATE/2=TERTIARY)
- Parm9 = シスプレックス名 (またはブランク)
- Parm10 = ビルド・オプション (1、2、または3)
- Parm11 = 戻りコード (INGEX02 の場合のみ)

ユーザー出口 INGEX10 が戻りコード RC = 0 を出す場合、ビルド処理は継続します。戻りコード RC > 0 が出されると、エラー・メッセージが返され、ビルド処理は終了します。

ユーザー出口 INGEX10 が戻りコード RC > 0 で終了した場合、ユーザー出口 INGEX01 および INGEX02 は呼び出されません。処理は終了します。

ユーザー出口 INGEX10 が戻りコード RC > 0 で終了し、カスタマイズ・ダイアログで BATCH の BUILD モードが選択されていた場合、バッチでビルドを実行するための JCL は実行依頼されません。処理は終了します。

ユーザー出口 INGEX01 が戻りコード RC = 0 を出す場合、ビルド処理は継続します。戻りコード RC > 0 が出されると、エラー・メッセージが返され、ビルド処理は終了します。ビルドがバッチ・モードで実行され、戻りコード RC > 0 が出された場合、ジョブは戻りコード RC 08 で終了します。

ユーザー出口 INGEX01 が戻りコード RC > 0 で終了した場合は、ビルド機能が開始されなかったため、ユーザー出口 INGEX02 は呼び出されません。処理は終了します。

BUILD 処理が開始されると、その処理が完了したかどうかに関係なく、常にユーザー出口 INGEX02 が呼び出されます。

ユーザー出口 INGEX02 が戻りコード RC > 0 を出すと、エラー・メッセージが表示されます。ビルドがバッチ・モードで実行され、戻りコード RC > 0 が出された場合、ジョブは戻りコード RC 04 で完了します。重大なビルド・エラーが発生した場合、ジョブは戻りコード RC 20 で完了します。

戻りコードとその意味は次のとおりです。

- 0** 正常終了
- 4** ビルド (マイナー・エラーあり)
- 12** ビルドなし (データに矛盾あり)
- 20** ビルドなし (重大エラー)

## COPY 処理用のユーザー出口

COPY 処理用に、2つのユーザー出口がインプリメントされます。

- INGEX03:** これは、COPY 機能を開始する前に呼び出されます。以下のパラメーターが渡されます。
  - コピー先 (ターゲット) の項目の項目名
  - コピー元 (ソース) の項目の項目名
  - 項目タイプ (例えば APL)
- INGEX04:** COPY 機能の終了後に呼び出されます。以下のパラメーターが渡されます。
  - コピー先 (ターゲット) の項目の項目名

- コピー元 (ソース) の項目の項目名
- 項目タイプ (例えば APL)
- COPY 処理が正常に行われたか否かのインディケータ (S= 成功、U= 失敗)

ユーザー出口 INGEX03 が戻りコード RC = 0 を出す場合、COPY 処理は継続します。戻りコード RC > 0 が出されると、エラー・メッセージが表示され、COPY 機能は開始されず、処理は終了します。

ユーザー出口 INGEX03 が戻りコード RC > 0 で終了した場合は、COPY 処理が終了するため、ユーザー出口 INGEX04 は呼び出されません。

COPY 機能が開始されると、常にユーザー出口 INGEX04 が呼び出されます。COPY 機能の成功または失敗に関する情報は、パラメーターとして渡されます。

ユーザー出口 INGEX04 が戻りコード RC > 0 を出すと、エラー・メッセージが表示されます。

## DELETE 処理用のユーザー出口

DELETE 処理用に、2つのユーザー出口がインプリメントされます。

1. **INGEX05:** これは、DELETE プロセスを開始する前に呼び出されます。以下のパラメーターが渡されます。
  - 削除される項目の項目名
  - 項目タイプ (例えば APL)
2. **INGEX06:** これは、DELETE プロセスの終了後に呼び出されます。以下のパラメーターが渡されます。
  - 削除される項目の項目名
  - 項目タイプ (例えば APL)
  - DELETE 処理が正常に行われたか否かを示すインディケータ (S= 成功、U= 失敗)

ユーザー出口 INGEX05 が戻りコード RC = 0 を出す場合、DELETE 処理は継続します。戻りコード RC > 0 が出されると、エラー・メッセージが表示され、DELETE 機能は開始されず、処理は終了します。

ユーザー出口 INGEX05 が戻りコード RC > 0 で終了した場合は、DELETE 処理が終了するため、ユーザー出口 INGEX06 は呼び出されません。

DELETE 機能が開始されると、常にユーザー出口 INGEX06 が呼び出されます。DELETE 機能の成功または失敗に関する情報は、パラメーターとして渡されます。

ユーザー出口 INGEX06 が戻りコード RC > 0 を出すと、エラー・メッセージが表示されます。

## CONVERT 処理用のユーザー出口

CONVERT 処理用に、2つのユーザー出口がインプリメントされます。

1. **INGEX07:** これは、CONVERT プロセスを開始する前に呼び出されます。パラメーターは渡されません。
2. **INGEX08:** これは、CONVERT プロセスの終了後に呼び出されます。パラメーターは渡されません。

ユーザー出口 INGEX07 が戻りコード RC = 0 を出す場合、CONVERT 処理は継続します。戻りコード RC > 0 が出されると、エラー・メッセージが表示され、CONVERT 機能は開始されず、処理は終了します。

ユーザー出口 INGEX07 が戻りコード RC > 0 で終了した場合は、CONVERT 処理が終了するため、ユーザー出口 INGEX08 は呼び出されません。

CONVERT 機能が開始されると、常にユーザー出口 INGEX08 が呼び出されます。

ユーザー出口 INGEX08 が戻りコード RC > 0 を出すと、エラー・メッセージが表示されます。

## RENAME および IMPORT 機能のユーザー出口

リネームおよびインポート機能用に以下のユーザー出口が提供されます。

- **INGEX15:** これは、項目がリネームされる前に呼び出されます。以下のパラメーターが渡されます。
  - 項目名
  - 項目タイプ

## 初期化出口

注：項目を名前変更すると、新しい項目が作成され、古い項目は削除されます。そのため、古い項目が削除される前に INGEX05 が呼び出され、項目が削除された後で INGEX06 が呼び出されます。

- **INGEX16:** これは、項目がリネームされた後で呼び出されます。以下のパラメーターが渡されます。
  - 項目タイプ
  - 古い項目名
  - 新規項目名
- **INGEX17:** これは、IMPORT 機能の実行中、ソース・ポリシー・データベースからデータが読み取られるときに、呼び出されます。1つのパラメーターが渡されます。
  - コピー・データ作業テーブルの名前。このテーブルには、コピーされるデータの項目タイプおよび項目名が入ります。
- **INGEX18:** これは、IMPORT 機能の終了後に呼び出されます。INGEX18 が呼び出されるのは、IMPORT 機能の開始時に INGEX17 が呼び出された場合のみです。検査が実行された結果、INGEX17 が呼び出されなかった場合、INGEX18 も呼び出されません。  
1つのパラメーターが渡されます。
  - IMPORT 処理が正常に行われたか否かのインディケーター (S= 成功、U= 失敗)

## その他の機能用のユーザー出口

その他の機能用に以下のユーザー出口が用意されています。

- **INGEX09:** これは、ログ・データ・セットが切り替えられる (通常、現行データ・セットがフルになったことが原因) と、呼び出されます。1つのパラメーターが渡されます。
  - 現行ログ・データ・セット (例えば、スペース不足になったデータ・セット) の名前
- **INGEX20:** これは、リンクの変更後に呼び出されます。パラメーターは渡されません。
- **INGEX21:** これは、ポリシー・データベース・レポートを起動する前に呼び出されます。パラメーターは渡されません。

## カスタマイズ・ダイアログ出口の呼び出し

ユーザー出口は SA z/OS 製品の一部です。したがって、他のすべての ISPFREXX モジュール (SINGTREX の一部) と同じデータ・セットで提供されます。提供されているすべてのサンプルでは、戻りコード RC=0 で「RETURN」が実行されます。

ユーザー変更を適用する方法として、次の2つの可能性があります。

1. 提供されたライブラリーでユーザー出口を編集する。ユーザーの変更は、SA z/OS 製品のコードに重大な結果をもたらすようなことはありません。ユーザー出口は、製品の出荷段階ではコードを何も組み込んでいないため、IBM による (PTF の) 保守はありません。
2. 変更済みのユーザー出口を専用データ・セットに渡す。その後、専用データ・セットを SYSEXEC ライブラリー・チェーンに連結する必要があります。INGDLG は、DD 名 SYSEXEC に指定される複数のデータ・セット名をサポートしているため、これは以下の方法で行うことができます。

```
INGDLG SELECT(ADMIN) ALLOCATE(YES) HLQ(SYS1)
SYSEXEC(usr.private.dsn SYS1.SINGTREX)
```

この例では、IBM 提供パーツが存在するデータ・セットの高位修飾子が SYS1 であると想定しています。

INGDLG 呼び出しで SYSEXEC パラメーターを指定する場合は、IBM 提供のライブラリーを、完全修飾データ・セット名で明示的に指定する必要があります。

## 初期化出口

これらの出口は、SA z/OS の初期化中に呼び出されます。

- AOFEXDEF
- AOFEXI01

- AOFEXI02
- AOFEXI03
- AOFEXI04
- AOFEXI05
- AOFEXI06
- AOFEXINT
- 環境設定出口

153 ページの図 31 は、SA z/OS の初期化時に出口が呼び出される順序を示しています。

SA z/OS 初期化時の出口の順序

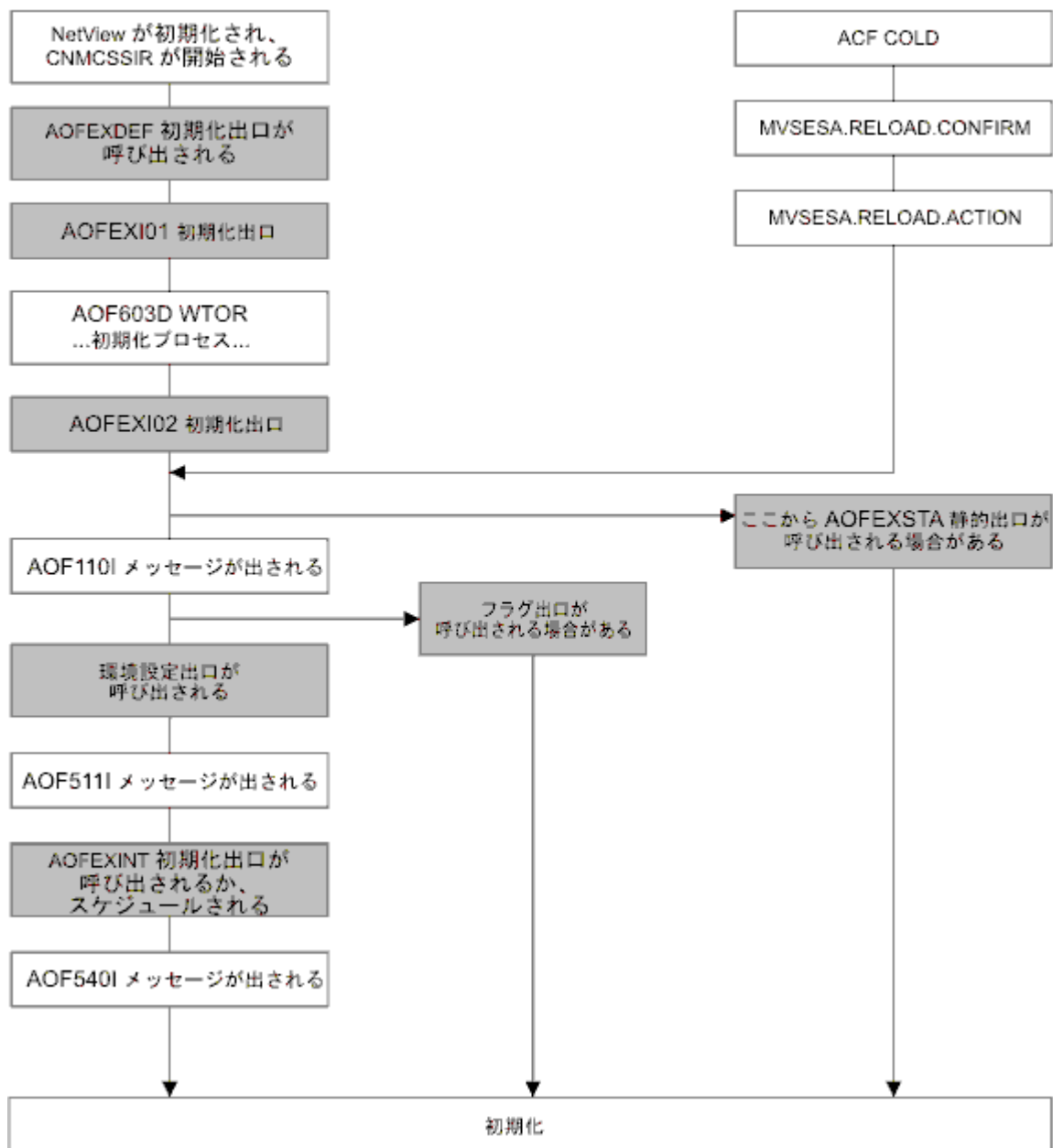


図 31. SA z/OS 初期化時の SA z/OS 出口の順序

### AOFEXDEF

この出口は、SA z/OS 初期化の開始時に、メッセージ AOF603D が出される 前に呼び出されます。例えば、AOFEXDEF を使用すると、異なる MPF テーブルをロードできます。

この出口は AUTINIT1 で実行されます。

パラメーター: なし。

戻りコード: 0 が予期されます。

### AOFEXI01

この出口は、AOF603D ENTER AUTOMATION OPTIONS 応答が出される前に呼び出されます。この出口は NetView PIPE 内に呼び出され、デフォルトの SAFE への入力として AOF767I メッセージで表示されるデータを入手します。この出口により、メッセージに行を追加したり、メッセージから行を除去することができます、また応答にさらにオプションを追加することもできます。

パラメーター: 入力でロード・タイプが渡されます。

#### IPL|RECYCLE

##### IPL

SA z/OS が IPL の後に開始されたことを示します。

##### RECYCLE

NetView と SA z/OS が再始動されたことを示します。

戻りコード: 0 が予期されます。

### AOFEXI02

この出口は、オペレーターが AOF603D 応答に応えた後で呼び出されます。この出口はデフォルトの SAFE への入力として、応答に対するオペレーターの応答を入手し、オペレーターが入力したオプションを除去、追加、または変更することができます。

パラメーター: 入力でロード・タイプが渡されます。154 ページの『AOFEXI01』を参照してください。

戻りコード: 0 が予期されます。

### AOFEXI03

この出口は、SA z/OS が NetView 自動化テーブルおよびメッセージ改訂テーブルをロードする前に呼び出されます。これを使用して、現在ロードされている AT の統計を作成できます。これらの統計は、ロードで SA z/OS が生成した AT リストとともに、任意の目的に使用できます。

パラメーター: なし。

戻りコード: 0 が予期されます。

### AOFEXI04

この出口は、SA z/OS が NetView 自動化テーブルおよびメッセージ改訂テーブルをロードした後に呼び出されます。これは、ロードで SA z/OS が生成した AT リストを保管するために使用できます。

パラメーター: なし。

戻りコード: 0 が予期されます。

### AOFEXI05

この出口は、ACF ロードまたはリフレッシュのいずれかが行われる前に呼び出されます。パラメーターは、自動化エージェントが処理するアクション (REFRESH または LOAD) を示します。

パラメーター: ACF アクションのタイプ (REFRESH または LOAD)。

戻りコード: 0 (注: この戻りコードは、呼び出し側により無視されます)。



## AOFEXI06

この出口は、ACF 処理 (LOAD または REFRESH) が完了した (AOFCOMPL=YES) 後、かつ AOF540I メッセージが発行される前に呼び出されます。

パラメーター: ACF アクションのタイプ (REFRESH または LOAD)。

戻りコード: 0 (注: この戻りコードは、呼び出し側により無視されます)。

## AOFEXINT

この出口は、SA z/OS 初期化が完了したとき、メッセージ AOF540I が出される前に呼び出されます。SA z/OS が終了した後で、ユーザー独自の初期化処理を行うため、AOFEXINT を使用することができます。AOFSERXINT グローバル変数のグローバル変数 AOFSERXINT の説明も参照してください。

パラメーター: 入力パラメーターは *Starttype* であり、これは RESYNC、IPL、REFRESH、RELOAD、RECYCLE の 1 つです。

戻りコード: 0 が予期されます。

## 環境設定出口

SA z/OS カスタマイズ・ダイアログを使用すると、SA z/OS 初期化処理中に呼び出される出口のストリングを定義することができます。

これらの出口は、システム (SYS) 項目タイプの SYSTEM INFO ポリシー項目を使用して定義します。詳しくは、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。

環境設定出口は、SA z/OS がいろいろなタスクを始動した後、1 次自動化テーブルをロードする前に呼び出されます。これらの出口を使用して、ユーザー独自の自動化を開始することができますが、出口を呼び出す時点では、SA z/OS が初期化をまだ終了していないため、一部の SA z/OS サービスは使用できないことがあります。特に、SA z/OS が再同期を終了していない場合は、状況情報が不正確になることがあります。環境設定出口は AUTINIT1 で実行されます。

### パラメーター

パラメーターは、ブランクで区切って、順番に渡されます。

### INITIALIZATION

INITIALIZATION は定数です。

### RELOAD|REFRESH|IPL|RECYCLE

#### RELOAD

自動化制御ファイルが再ロードされたことを示します。

#### REFRESH

自動化制御ファイルがリフレッシュされたことを示します。

#### IPL

システム IPL の後で SA z/OS が再始動されたばかりであることを示します。

#### RECYCLE

NetView が再始動されたことを示します。

### 戻りコード

0 が予期されます。ゼロ以外の戻りコードが返された場合に、他の出口が呼び出されないようにしたり、SA z/OS の初期化を中止することができます。

### 使用上の注意

- RESYNC を実行すると、これらの出口は駆動されません。
- 他の静的出口とは異なり、自動化制御ファイルの中で呼び出すルーチン (1 つ以上) の名前を指定する必要があります。

## 静的出口

この出口は、SA z/OS 処理における一定の個所で呼び出されます。

これは、DSICLD 連結の中で検出されると、常に呼び出されます。常に、戻りコード 0 で終了することをお勧めしますが、一般には、この出口からの正の戻りコードは無視されます。

静的出口の主な目的は、SA z/OS 処理の特定の個所でユーザー固有のアクション を実行できるようにすることにあります。使用できる静的出口について、以下で説明します。

### AOFEXSTA

この出口は、アプリケーションの自動化状況が更新されるたびに、AOCUPDT から呼び出されます。


**注:** この出口が呼び出されるように、AOCUPDT がこの出口のアプリケーション自動化状況を変更する必要はありません。この出口は、更新しても状況が変更されない場合にも呼び出されます。

AOFEXSTA は、他の方式では起動できないすべての特殊な状況推移 処理を実行するために使用することができます。

**注:** この出口は頻繁に呼び出されます。また、SA z/OS が完全に初期化されていない時点で呼び出されま。出口プログラムは、できる限り耐久性があり、効率的なものにしてください。

SA z/OS は、初期化時に AOFEXSTA をストレージにロードしようとします。この試みが失敗すると、いずれの AOCUPDT 呼び出しの際にも、AOFEXSTA が呼び出されることはありません。出口を活動化するには、自動化制御ファイルのロード時または再ロード時に、これが DSICLD 連結の中にある必要があります。

AOFEXSTA は、他のすべての処理が終了してから、AOCUPDT を呼び出した タスクで実行されます。

 **重要:** AOFEXSTA は、EXCMD opid() によってスケジュールされます。オペレーターがアプリケーション状況を変更するコマンドを出しているときに、ユーザーが AOFEXSTA を使用したい場合は、コマンド権限の定義を修正する必要がある場合があります。

**パラメーター:** パラメーターは、コンマで区切られて順番に渡されます。

#### リソース・タイプ

SA z/OS は、タイプ SUBSYSTEM、MVSESA、WTORS、および SPOOL を使用します。他のユーザーは、他のリソース・タイプを使用できます。

#### Resource Name

アプリケーションの場合、これは、定義されているとおりのサブシステムの名前です。

#### 自動化状況 (Automation status)

アプリケーションの場合、これは、SA z/OS でサポートされる自動化状況の 1 つです。

#### SDF Root

これは、カスタマイズ・ダイアログで指定したとおり、状況更新を発信したシステムの SDF ルートです。一般に、出口が駆動されるのは、自動化フォーカル・ポイントにおける他のシステムでの状況変化の場合だけです。

**戻りコード:** 0 が予期されます。

#### 制約事項:

- 出口は EXCMD によってスケジュールされるため、呼び出し側での状況更新 および後続の処理は、出口を呼び出す前に完了しています。
- リソース・タイプと SDF ルートを調べて、正しいものの処理のみを行っていることを確認します。
- この出口から、アプリケーションの状況を変更するために、何らかのアクションを行う場合は、慎重に計画を立ててから行ってください。間違えると、ループ (AOCUPDT - AOFEXSTA - AOCUPDT - AOFEXSTA) が生じることがあります。

**注:**

1. AOFEXSTA は、すべての状況更新で呼び出されるためにパフォーマンスの大幅な低下を引き起こすので、AOFEXSTA の代わりに、ISSUEACT または状況変更コマンドを使用することを検討してください。

2. 拡張自動化オプションが適切に設定されていれば、ACTIVMSG および TERMMSG コマンドは、アプリケーションが特定の状況に変化すると必ず コマンドを実行します。ここにコマンドを入れる方が、(すべてのリソースのあらゆる状況更新について駆動される) 状況変更出口に入れるよりも適切であることがあります。パフォーマンスを改善するために、状況変更コマンドの使用をお勧めします。

## AOFEXX02

この出口は、インストール済み環境が、指定したリソースに対して SDF 更新が実行されるか否かを判別できるようにします。

出口からゼロ以外の戻りコードが返された場合、SDF 更新処理は、フォーカル・ポイントだけでなくローカルでもスキップされます。

この出口は、特定のイベントをフィルターに掛ける機能を提供するため、SDF への項目の転送の前に呼び出されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## AOFEXX04

この出口は、指定リソースの自動化状況ファイルに記録されたエラー数を、自動化制御ファイルで定義されたエラーしきい値と対照して検査するために CHKTHRES ルーチンが呼び出されるたびに、そのルーチンから呼び出されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## AOFEXX05

この出口は、SDF のパネルおよびツリーのセットアップ時に初期化中に SA z/OS によって、または RESYNC SDFDEFS コマンドによって駆動されます。

この出口はインストールで使用され、SDF パネル定義でユーザー変数を置き換えます。ユーザー変数は z/OS のシステム・シンボルと同じ規則に従う必要があります。すなわち、ユーザー変数はアンパーサンド (&) で始まり、ドット (.) で終わらなければなりません。例えば、&MVDOMAIN のように指定します。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## フラグ出口

自動化フラグ出口を使用すると、自動化された演算コードが、通常の SA z/OS 処理を終了して外部ソース (スケジューリング機能など) に移動し、その特定の時点で指定されたリソースの自動化がオンになっているかオフになっているかを判別できるようになります。

任意のメジャー・リソースまたはマイナー・リソース上の任意のフラグ (AUTOMATION、INITSTART、START、RECOVERY、TERMINATE、または RESTART) について、フラグ出口を定義できます。マイナー・リソースの詳細については、「*IBM Z System Automation* 自動化ポリシーの定義」に記載されている MINOR RESOURCES ポリシー項目の説明を参照してください。

各フラグごとに複数の出口を指定することができます。

フラグ出口が呼び出されるのは、「*IBM Z System Automation* プログラマーズ・リファレンス」で説明されているように、AOCQRY のフラグ評価処理中に、SA z/OS が、現行フラグ設定の値を検査する場合のみです。継承の階層レベルでの評価処理の際に 1 回の反復ステップで検査する必要がある、グローバル・フラグまたは特定フラグの 1 つが「NO」に設定されている場合、他のフラグを検査する必要はなくなります。

AOCQRY のデフォルトの BYPASS オプションを使用すると、フラグ評価の際に自動化フラグが検査され、フラグ値が EXITS のときに、リソースの自動化フラグに定義された出口が実行されます。

AOCQRY の FORCED オプションを使用すると、フラグ評価の際に自動化フラグが検査されたときに、フラグ値がブランクでない限りはフラグ値に関係なく、リソースの自動化フラグに定義された出口が実行されます。

自動化フラグが EXITS に設定されている場合、検査対象リソースに定義されているどの出口もフラグを「NO」に切り替えない限り、フラグ検査中のフラグ値は「YES」であると想定されます。

## フラグ出口

フラグ設定値は、以下によって決まります。

- 自動化ポリシー設定
- NOAUTO 期間 (NOAUTO 期間中フラグは OFF になっている)。
- ユーザー入力の INGAUTO コマンド

例えば、以下のフラグ設定値を入力する場合、

リソース	フラグ	設定値
DEFAULTS	AUTOMATION	YES
SUBSYSTEM	RESTART	NO
JES2	AUTOMATION	出口 J2AUT
JES2	START	出口 J2STR
JES2	RECOVERY	NO
JES2	TERMINATE	出口 J2SD1 および J2SD2

JES2 の有効フラグは、以下のとおりです。

フラグ	有効な設定
AUTOMATION	YES、出口 J2AUT
INITSTART	YES
START	YES、出口 J2STR
RECOVERY	NO
TERMINATE	YES、出口 J2SD1 および J2SD2
RESTART	NO

SA z/OS が JES2 アプリケーションの任意のフラグの現行値を調べる場合、処理は次のようになります。

フラグ	処理
AUTOMATION	<ol style="list-style-type: none"> <li>1. 出口 J2AUT を呼び出します。</li> <li>2. 出口が戻す値が、 <ul style="list-style-type: none"> <li>• <b>RC=0:</b> AUTOMATION フラグは YES</li> <li>• <b>RC&gt;0:</b> AUTOMATION フラグは NO</li> </ul> </li> </ol>
INITSTART	<ol style="list-style-type: none"> <li>1. 出口 J2AUT を呼び出します (AUTOMATION グローバル・フラグのため)。</li> <li>2. 出口が戻す値が、 <ul style="list-style-type: none"> <li>• <b>RC=0:</b> INITSTART フラグは YES</li> <li>• <b>RC&gt;0:</b> INITSTART フラグは NO</li> </ul> </li> </ol>
START	<ol style="list-style-type: none"> <li>1. 出口 J2AUT を呼び出します (AUTOMATION グローバル・フラグのため)。</li> <li>2. 出口が RC=0 を戻す場合、出口 J2STR を呼び出します。</li> <li>3. また、 <ul style="list-style-type: none"> <li>• <b>両方のフラグが RC=0 を戻す場合:</b> START フラグは YES</li> <li>• <b>いずれかのフラグが RC&gt;0 を戻す場合:</b> START フラグは NO</li> </ul> </li> </ol>

フラグ	処理
RECOVERY	RECOVERY フラグは NO
TERMINATE	<ol style="list-style-type: none"> <li>1. 出口 J2AUT を呼び出します (AUTOMATION グローバル・フラグのため)。</li> <li>2. 出口が RC=0 を戻す場合、出口 J2SD1 を呼び出します。</li> <li>3. 出口 J2SD1 が RC=0 を戻す場合、出口 J2SD2 を呼び出します。</li> <li>4. また、 <ul style="list-style-type: none"> <li>• 両方のフラグが <b>RC=0</b> を戻す場合: TERMINATE フラグは YES</li> <li>• いずれかのフラグが <b>RC&gt;0</b> を戻す場合: TERMINATE フラグは NO</li> </ul> </li> </ol>
RESTART	<ol style="list-style-type: none"> <li>1. 出口 J2AUT を呼び出します (AUTOMATION グローバル・フラグのため)。</li> <li>2. 戻りコードにかかわらず、RESTART フラグは NO です。</li> </ol>

注: 通常、START フラグと RECOVERY フラグは、SA z/OS によって、マイナー・リソースに対してのみ検査され、サブシステムに対しては検査されません。

## パラメーター

パラメーターは、ブランクで区切って、順番に渡されます。

### フラグ

これは、調べるフラグの名前です。使用できる値は、AUTOMATION、INITSTART、START、RECOVERY、TERMINATE、または RESTART です。

### Time Setting

時間設定値は定数です。これは、以下のいずれかとすることができます。

#### AUTO

現在、自動化はオンになっています。

#### NOAUTO

現在、自動化はオフになっています。

値 NOAUTO が使用できるのは、パラメーター EXITS=FORCED を指定して AOCQRY が呼び出される場合だけです。

注: これは、確実に出口を呼び出しますが、出口に関する NOAUTO 期間の指定を変更することはできません。

### Resource Name

これは、フラグが要求されるリソースの名前です。マイナー・リソースの場合は、完全修飾マイナー・リソース名が入ります。TSO.USER.MAG1 のフラグ定義を指定せずに TSO.USER に対して使用可能にされた出口を指定すると、TSO.USER.MAG1 についての検査が行われた場合、出口に渡されるリソース名は TSO.USER.MAG1 になります。

### Resource Type

これは、フラグが要求されるリソースのタイプです。使用できる値は、DEFAULTS、SUBSYSTEM、または MVSESA (共通グローバル変数 AOFSYSTEM の値) です。

### Target Prefix

これは、AOCQRY を呼び出したときの TGPFX 値です。TGPFX を指定しないと、値 SUB が渡されます。

## タスク・グローバル変数

AOCQRY コマンドで設定されるタスク・グローバル変数は、フラグ出口で使用できます。

### 戻りコード

0

出口によって自動化が許可されています。

>0

自動化が許可されていません。

注：

1. フラグ出口は常に AOCQRY コマンドで呼び出されます。これは、リソースのタスク・グローバル変数が事前準備されており、使用可能になっているという意味です。通常、タスク・グローバル変数の名前は、接頭部に SUB が付きますが、パラメーター TGPFIX に異なる値を指定して AOCQRY を呼び出した場合、タスク・グローバル変数の名前は、接頭部にその値が付いた変数の中にあります。タスク・グローバル変数を見つけるために渡される TGPFIX パラメーターを使用する必要があります。
2. AOCQRY で設定されるタスク・グローバル変数のセットは、リソースと要求パラメーターの値によって異なります。ユーザー出口で使用するタスク・グローバル変数がセットアップされていることを確認してください。
3. マイナー・リソースを対象として出口を呼び出すと、タスク・グローバル変数は、そのマイナー・リソースと関連したメジャー・リソース用に設定されます。
4. ユーザー出口の内側から AOCQRY を呼び出す場合、渡された TGPFIX パラメーター値とは異なる TGPFIX 値を指定する必要があります。AOCQRY をネストする場合、ユーザーは TGPFIX がすべて固有であることを確認する必要があります。この作業は非常に複雑になる場合があるので、出口をネストするのは避けることをお勧めします。
5. ACFICMD、ACFREP、または CDEMATCH は、接頭部が SUB のタスク・グローバル変数 (ユーザーが処理したいアプリケーション用に設定されないことがある) を使用するため、これらへの呼び出しをコーディングしないでください。
6. AOCQRY で設定されたタスク・グローバル変数は、どれも変更しないでください。
7. フラグ出口は頻繁に呼び出されることがあるので、パフォーマンスが重要です。
8. AOCQRY を FORCE とともに指定し、フラグに複数の出口を定義すると、その出口は順に呼び出されません。

## コマンド出口

これらの出口は、特定のコマンドの処理中に呼び出されます。

サンプル出口が製品に用意されています。サンプル出口のコピーおよび更新については、「*IBM Z System Automation* 計画とインストール」を参照してください。

### AOFEXC00

AOC コマンド・ダイアログで選択項目 L を入力すると、AOFEXC00 出口ルーチンが呼び出されます。このルーチンにはパラメーターは渡されません。このルーチンの目的は、インストール・システム が提供するローカル機能の開始点としての役割を果たすことです。

### AOFEXC01

この出口は、定義されると、INGREQ の処理中の事前検査処理と検査処理の前に呼び出されます。

この出口では、渡されたパラメーターを変更できます。

出口に渡されるパラメーターおよび戻りコードについて詳しくは、サンプル出口を参照してください。

この出口は、INGREQ REQ=CANCEL 処理中には呼び出されません。代わりに出口 AOFEXC08 を使用できます。

**AOFEXC02**

この出口ルーチンが定義された場合、この出口は、INGSCHEd 処理で、スケジュール・オーバーライド・ファイルが更新される前に呼び出されます。パラメーターは定位置パラメーターで、コンマで区切られます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

**AOFEXC03**

この出口ルーチンが定義された場合、この出口は、サブシステムについてのユーザー提供の情報を検索するための DISPINFO コマンド・スレーブによって呼び出されます。ルーチンの入力にはサブシステム名です。出口によって返されたデータは、DISPINFO 出力の一部として表示されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

**AOFEXC04**

この出口ルーチンが定義された場合、DISPMTR、DISPSTAT、および INGLIST コマンドでコマンド・コード U がサポートされます。AOFEXC04 出口の入力は、リソース名 (DISPSTAT のサブシステム名) およびリソースの場所です。

場所は、リソースがローカル・シスプレックスのシステム・メンバーにある場合はシステム名で、リソースがローカル・シスプレックス外のシステムにある場合はドメイン ID です。シスプレックス APG リソース、REF リソースおよび DMN リソースの場合、場所は常に INGLIST コマンドが呼び出されたシステムになります。パラメーターはコンマで区切ります。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

**AOFEXC05**

この出口は、INGLIST コマンドの入力時に呼び出されます。この出口では、入力パラメーターを変更できます。変更された入力パラメーターは、以下の例のように、メッセージ (単一行メッセージまたは複数行メッセージ) をコンソールに送信することによって、INGLIST コマンドに戻されます。

```
OBSERVED=* DESIRED=*
```

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

**AOFEXC06**

この出口は、SET アクションを伴う INGSET コマンドの入力時に呼び出されます。この出口では、INGSET コマンドのリソースの許可検査を実行できます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

**AOFEXC07**

この出口は、INGIMS コマンドの入力時に呼び出されます。この出口では、INGIMS コマンドの対象である IMS サブシステムの許可検査を実行できます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

**AOFEXC08**

この出口は、INGVOTE コマンドの入力時に呼び出されます。この出口では、INGVOTE コマンドのリソースの許可検査を実行できます。INGSET CANCEL/KILL アクションは INGVOTE コマンドを使用するため、この出口は、このアクションを実行するときにも呼び出されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。



### AOFEXC09

この出口は、SETSTATE コマンドの入力時に呼び出されます。この出口では、SETSTATE コマンドのリソースの許可検査を実行できます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

### AOFEXC10

この出口は、INGEVENT コマンドの入力時に呼び出されます。この出口では、INGEVENT コマンドのリソースの許可検査を実行できます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

### AOFEXC11

この出口は、INGCICS コマンドの入力時に呼び出されます。この出口では、INGCICS コマンドのリソースの許可検査を実行できます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

### AOFEXC13

この出口は、INGGROUP コマンドおよび INGMOVE コマンドの入力時に呼び出されます。この出口では、コマンドを発行するユーザー ID の許可検査を実行できます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

### AOFEXC14

この出口は、PAM を停止した後、または PAM となる SAM を選択した後に、SA z/OS GDPS 終了ルーチン (INGRGDPS) によって呼び出されます。

戻りコードについては、サンプル出口を参照してください。

### AOFEXC15

この出口ルーチンは、定義されると、INGREQ の処理中の GO 確認の受信後に呼び出されます。

ユーザー出口は PIPE に呼び出されます。出口に渡されるパラメーターについては、サンプル出口を参照してください。

### AOFEXC16

この出口は、指定されたリソースのしきい値を更新または削除する前に、INGTHRES コマンドによって呼び出されます。この出口では、要求されたアクションの許可検査を実行できます。出口がゼロ以外の戻りコードを戻し、コンソールに追加データが書き込まれた場合、このデータはメッセージ・パネルに表示されます。出口に追加データが戻されない場合は、メッセージ AOF227I が発行されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

### AOFEXC17

この出口は、INGALERT コマンドによって呼び出されます。これにより、以下のことを行うことができます。

- イベント・テキストの変更
- IOM、EIF、TTT、および USR などのイベント通知ターゲットによる、通知リストの削減
- イベントの重大度、または、イベントを無視するかどうかなどの情報を使用して、一致するコード定義から返された値を変更

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## AOFEXC18

この出口は、INGLKUP コマンドによって呼び出されます。これは、指定されたアドレス・スペースの停止または取り消しの前に駆動されます。この出口では、要求されたアクションの許可検査を実行できます。出口がゼロ以外の戻りコードを返し、コンソールに追加データが書き込まれた場合、このデータはメッセージ・パネルに表示されます。出口に追加データが戻されない場合は、メッセージ AOF227I が発行されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## AOFEXC19

この出口は、INGAMS コマンドによって呼び出されます。以下の場合に、駆動されます。

- テークオーバー・ファイルへのアクセスを可能または不可に設定する場合
- システムを中断または再開する場合
- 構成をリフレッシュする場合
- 診断アクションを実行する場合 (記録の開始または停止、スナップショットの作成)
- 1次自動化マネージャーを切り替える場合

この出口では、INGAMS コマンドの許可検査を実行できます。出口がゼロ以外の戻りコードを返し、コンソールに追加データが書き込まれた場合、このデータはメッセージ・パネルに表示されます。出口に追加データが戻されない場合は、メッセージ AOF227I が発行されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## AOFEXC20

この出口は、TWS 要求インターフェースを介してコマンドが渡されたときに呼び出されます。この出口を使用することにより、インストール済み環境で許可検査を実行することができます。オプションで、この出口は最大2つのデータ行を戻すことによりコマンドと完了情報を変更できます。

- 行1は変更されるコマンドです (そのすべてのパラメーターを含む)。コマンドが変更されない場合は、ヌル・ストリングが戻されます。これはこの出口で完了情報を変更する場合に限り必要です。
- 行2は完了情報です。
  - 最大戻りコード
  - コーディング検査の完了

パラメーターはコンマで区切る必要があります。いずれかのパラメーターが間違っている場合は、エラー・コード U010 がポストされます。

インストール・システム出口は PIPE で呼び出されます。出口が誤った戻りコードを返し、コンソールに追加データが書き込まれた場合、このデータはネットログに書き込まれます。出口に追加データが渡されない場合は、メッセージ AOF227I が発行されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## AOFEXC21

この出口は、オプション REQ=MOD を指定した INGOPC コマンドによって呼び出されます。この出口では、要求されたアクションの許可検査を実行できます。出口がゼロ以外の戻りコードを返し、コンソールに追加データが書き込まれた場合、このデータはメッセージ・パネルに表示されます。出口に追加データが戻されない場合は、メッセージ AOF227I が発行されます。

出口に渡されるパラメーターおよび戻りコードについては、サンプル出口を参照してください。

## AOFEXC22

この出口は、INGALERT コマンドを使用してトラブル・チケットが作成されたときに呼び出されます。この出口により、詳細データ・セットに書き込まれるトラブル・チケットの詳細データの判別が可能になります。

## コマンド出口

出口に渡されるパラメーターおよび戻りコードについて詳しくは、サンプル出口を参照してください。

### AOFEXC23

この出口は、TWS インターフェースを介して要求が渡されたときに呼び出されます。この出口では、要求されたアクションの許可検査を実行できます。出口がゼロ以外の戻りコードを戻し、コンソールに追加データが書き込まれた場合、このデータはメッセージとみなされます。出口に追加データが戻されない場合は、メッセージ AOF227I が発行されます。

出口に渡されるパラメーターおよび戻りコードについて詳しくは、サンプル出口を参照してください。

### AOFEXC24

この出口は、定義されていると INGRUN の処理中に呼び出されます。この出口では、渡されたパラメーターを変更できます。

出口に渡されるパラメーターおよび戻りコードについて詳しくは、サンプル出口を参照してください。

### AOFEXC25

この出口は、INGAMS REFRESH 要求が処理されるときに呼び出されます。この出口により、新規、変更済み、削除済みの、サブシステム (APL)、アプリケーション・グループ (APG)、およびモニター・リソース (MTR) に関する詳細が提供されます。この出口は、NetView オペレーター AUTO2 によって駆動され、リフレッシュ処理中に性能低下を防ぎます。戻りコード 0 を返すことをお勧めします。ただし、戻りコードは現在では検査されません。

詳しくは、サンプル出口を参照してください。

### AOFEXC26

この出口は、定義されていれば INGSUSPD の処理中に検査処理の前に呼び出されます。この出口を使用すれば、渡されるパラメーターを検査できます。

出口に渡されるパラメーターおよび戻りコードについて詳しくは、サンプル出口を参照してください。デフォルト処理では、渡されたコメントが空かどうかを検査されます。コメントが指定されていない場合、サンプル出口で中断要求は投入できません。

この出口は、INGSUSPD REQ=RESUME 処理中には呼び出されません。

### AOFEXC27

AOFEXC27 ユーザー出口は、指定されたパラメーターを使用して INGAUTO コマンドの許可検査を実行します。INGAUTO コマンドおよび DISPFLGS コマンドによる自動化関連フラグの設定を許可したり禁止したりする柔軟性が得られます。

AOFEXC27 は、定義されている場合、INGAUTO パラメーターが渡された後で、INGAUTO 処理中に呼び出されます。続いて AOFEXC27 は、パラメーターに基づいて INGAUTO 実行を許可するか拒否するかを決定します。

SUSPEND 要求および RESUME 要求は、AOFEXC27 定義とは無関係に、常に実行されます。

出口に渡されるパラメーターと戻りコードについて詳しくは、SINGSAMP ライブラリーにあるサンプル出口を参照してください。

### AOFEXC28

この出口が定義されている場合、この出口は、INGDYN の処理時に、動的リソースが作成または削除される前に呼び出されます。

INGDYN CREATE の場合、この出口では、動的リソースの作成をリジェクトしたり、渡されたパラメーターを変更したりできます。

INGDYN DELETE の場合、この出口では、動的リソースの削除をリジェクトできます。

このユーザー出口は PIPE で呼び出されます。出口に渡されるパラメーターおよび戻りコードについて詳しくは、サンプル出口を参照してください。

## AOFEXC29

この出口が定義されている場合、この出口は、INGDYN の処理時に、動的リソースを作成または削除するために GO 確認が受信された後で呼び出されます。

例えば、新規動的リソースを作成後に再開する場合や、新規リソースを作成してホスティング・グループに追加した後でホスティング・グループのプロパティを適応させる場合、この出口を使用できます。

このユーザー出口は PIPE で呼び出されます。出口に渡されるパラメーターについて詳しくは、サンプル出口を参照してください。

## 疑似出口

この節では、SA z/OS がフラグ出口を特別な用途で使用する場所、あるいは SA z/OS がある種の出口のような性質の機能を持つ場所をいくつか説明します。

### 自動化制御ファイルの再ロード許可出口

オペレーターが ACF COLD コマンドを発行すると、SA z/OS は、マイナー・リソース MVSESA.RELOAD.CONFIRM のグローバル AUTOMATION フラグを調べます。

このフラグが NO に設定されていると、自動化制御ファイルの再ロードはできません。このフラグが YES に設定されていると、タスク・グローバル変数 AOFCONFIRM がチェックされます。AOFCONFIRM が非ヌル値に設定されていると、ユーザーは、自動化制御ファイルを再ロードしたいか確認するようにプロンプトが出されます。

注：

1. 出口は、このリソースのグローバル AUTOMATION フラグと関連付けられている可能性がある点に注意してください。
2. 自動化制御ファイルは、メジャー・リソース MVSESA のグローバル AUTOMATION フラグが N に設定されている場合は、ロードできません。マイナー・リソース MVSESA.RELOAD.CONFIRM のグローバル AUTOMATION フラグが Y に設定されている場合は、ACF の再ロードが許可されます。

### 自動化制御ファイルの再ロード・アクション出口

自動化制御ファイルの再ロード許可出口を調べた後で、SA z/OS が自動化制御ファイルを再ロードすることをコミットすると、SA z/OS はマイナー・リソース MVSESA.RELOAD.ACTION のグローバル AUTOMATION フラグを調べます。

このフラグの実際の設定値 (ON または OFF) は無視されますが、そのフラグに定義されている出口がすべて呼び出されます。出口はすべて、戻りコード 0 を戻すはずでです。

### 初期化コマンドにおけるサブシステム UP

SA z/OS が再同期状況を終了し、アプリケーションが稼働していることが検出された場合に発行するコマンドを指定することができます。これらのコマンドは、「カスタマイズ・ダイアログ」>「APL 項目タイプ」>「STARTUP ポリシー」>「REFRESHSTART フェーズ」で指定します。

これらのコマンドは、SA z/OS の上部にビルドされたローカルの自動化を同期する上で役立ちます。

## 出口のテスト

出口は、実動に移す前に、各種の入力パラメーターを使用して、十分にテストする必要があります。AOCQRY タスク・グローバル変数を必要とする出口の場合は、AOCQRY を呼び出して、フラグ出口を評価せずにグローバル変数を設定してから、その出口をテスト目的でそのまま呼び出すことができます。この方式は、出口を実行するたびに AOCQRY の呼び出しのオーバーヘッドのむだを省きます。



**重要：**

## 出口のテスト

出口で構文上にエラーがあったり、値を持たない条件があると、SA z/OS の一部が異常終了する原因になり、自動化に深刻な影響を与えることがあります。

## 第 17 章 自動化ソリューション

SA z/OS は、自動化処理のコンテキストで役立つツールである自動化プロシージャのほかに、z/OS コンポーネント、データ・セットと、ジョブ・スケジュール・システムの自動処理を使用可能にするソリューションも提供します。これらの既製の自動化プロシージャを使用することによって、相当する状態での処理を扱う独自のプロシージャを開発する時間を節約できます。

特に、これらの自動化ルーチンは以下に対する解決策を提供します。

- [167 ページの『LOGREC データ・セット処理』](#)
- [168 ページの『SMF データ・セット処理』](#)
- [168 ページの『SYSLOG 処理』](#)
- [168 ページの『システム・ログ障害のリカバリー』](#)
- [168 ページの『SVC ダンプ処理』](#)
- [169 ページの『処理済み WTOR の表示からの削除』](#)
- [169 ページの『AMRF バッファ不足の処理』](#)
- [44 ページの『JES2 スプールのモニター』](#)
- [169 ページの『JES2 シャットダウン前のドレイン処理』](#)
- [169 ページの『IMS トランザクション・リカバリー』](#)
- [170 ページの『INGWHY ユーザー・アクションの定義』](#)
- [172 ページの『ループ・アドレス・スペース抑制』](#)

これらの状態の自動処理に対するソリューションには、自動化構成ファイル内の定義と自動化プロシージャが含まれます。

対応する自動化フラグを AOCQRY コマンドで検査することによって、自動化が許可されているかどうかを自動化プロシージャが最初に判別することは、提供されるソリューションのすべてに共通しています。自動化フラグのタイプと設定値に関する詳細については、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。自動化フラグの現行の設定値を表示するか、一時的に変更するには、DISPFLGS コマンドを使用してください。

一部の自動化ルーチンは、自動化構成ファイルからコマンドを出すことによってメッセージに応答します。これらのほとんどの自動化ルーチンは、これらのメッセージの受信を記憶し、受信したメッセージの頻度を、低頻度 (infrequent)、高頻度 (frequent)、クリティカル (critical) レベルの事前に定義されたしきい値と比較します。そのような定義済みのしきい値を超えた場合は、構成ファイルの MESSAGES/USER DATA ポリシー項目のコマンド項目中の最初のフィールドに従って適切なコマンドを選択するためのオプションとして使用されます。しきい値を超えていない場合は、選択オプション ALWAYS 用に定義されたコマンドが実行されます。しきい値のセットアップについて詳しくは、「*IBM Z System Automation* ユーザーズ・ガイド」の『SA z/OS によるエラーしきい値の使用法』を参照してください。

本章では、SA z/OS によって提供される自動化機能の詳細について説明します。

### LOGREC データ・セット処理

logrec リカバリー機能は、logrec データ・セットがフルであるか、あるいはフルに近いことを示しているシステム・メッセージに応答します。このリカバリー機能は、logrec データ・セットをダンプしてクリアする事前定義のコマンドを発行します。リカバリー機能が進行中である間は、自動化処理の 2 回目の開始を停止します。

logrec リカバリー機能には、以下の項目が含まれます。

- [自動化ルーチン AOFRSA01 および AOFRSA02 \(177 ページの『AOFRSA01』および 178 ページの『AOFRSA02』を参照\)](#)
- システム・メッセージ IFB040I、IFB060E、IFB080E、IFB081I と、IFC001I 用の自動化テーブル項目
- MVS コンポーネント・マイナー・リソース LOGREC に対するエラーしきい値定義

- MESSAGES/USER DATA 自動化ポリシー項目の特別メッセージ ID LOGREC に対するコマンドの指定

## SMF データ・セット処理

提供された SMF リカバリー機能は、SMF データ・セットがフルになっているか、あるいは切り替えられたことを示しているシステム・メッセージにตอบสนองします。SMF データ・セットの内容をダンプしてクリアするために構成ファイルから事前に定義されたコマンドが選択されます。

選択されるコマンドは、受信メッセージのオカレンスに応じて定義できます。また、SA z/OS は初期化時に SMF データ・セットをダンプする必要があるかどうかを検査し、適切なリカバリー・アクションをトリガーします。SMF リカバリー機能には、以下の項目が含まれます。

- 自動化ルーチン AOFRSA03 (179 ページの『AOFRSA03』を参照)
- システム・メッセージ IEE362A、IEE362I、IEE391A と、IEE392I 用の自動化テーブル入力
- MVS コンポーネント・マイナー・リソース SMFDUMP に対するエラーしきい値定義
- MESSAGES/USER DATA 自動化ポリシー項目の特別メッセージ ID SMFDUMP に対するコマンドの指定

## SYSLOG 処理

提供された syslog 機能は、syslog のキューに入れられたメッセージにตอบสนองします。この機能は、キューに入れられた syslog を保管するために、外部書き出しプログラムを開始します。選択されるコマンドは、受信メッセージのオカレンスに応じて定義できます。

syslog 機能には、以下の項目が含まれます。

- 自動化ルーチン AOFRSA08 (181 ページの『AOFRSA08』を参照)
- システム・メッセージ IEE043I 用の自動化テーブル項目
- MVS コンポーネント・マイナー・リソース SYSLOG に対するエラーしきい値定義
- MESSAGES/USER DATA 自動化ポリシー項目の特別メッセージ ID SYSLOG に対するコマンドの指定

## システム・ログ障害のリカバリー

提供されたシステム・ログ障害のリカバリー機能は、システム・ログをリスタートすることによって、システム・ログの非アクティブ・メッセージにตอบสนองします。システム・ログをハードコピー・メディアとして使用可能にする必要がある場合、リカバリー機能は、システム・ログをハードコピー・メディアとして割り当てます。

リカバリー・コマンドが出されるのは、受け取るシステム・ログの非アクティブ・メッセージのオカレンスが定義されたクリティカル (critical) のしきい値を超えない場合のみです。

提供されたシステム・ログ障害のリカバリー機能には、以下の項目が含まれます。

- 自動化ルーチン INGRX740 (199 ページの『INGRX740』を参照)
- システム・メッセージ IEE037D、IEE041I、IEE533E、IEE769E、IEE043I 用の自動化テーブル項目
- MVS コンポーネント・マイナー・リソース LOG に対するリカバリー自動化フラグ
- MVS コンポーネント・マイナー・リソース LOG に対するエラーしきい値定義
- MESSAGES/USER DATA 自動化ポリシー項目の特別メッセージ ID LOG に対するコマンドの指定

## SVC ダンプ処理

提供された SVC ダンプ処理機能は、ダンプを処理するための事前定義コマンドを構成ファイルから発行することにより、SVC ダンプ採取メッセージにตอบสนองします。選択されるコマンドは、受信メッセージのオカレンスに応じて定義できます。

提供された SVC ダンプ処理機能には、以下の項目が含まれます。

- 自動化ルーチン AOFRSA0C (183 ページの『AOFRSA0C』を参照)
- システム・メッセージ IEA611I と IEA911E 用の自動化テーブル項目



- MVS コンポーネント・マイナー・リソース MVSDUMP に対するエラーしきい値定義
- MESSAGES/USER DATA 自動化ポリシー項目の、以下の特別メッセージ ID に対するコマンドの指定
  - MVSDUMP
  - MVSDUMPTAKEN
  - MVSDUMPRESET

## 処理済み WTOR の表示からの削除

提供された WTOR 処理機能は、WTOR が応答または取り消しされるとときに、WTOR を SA z/OS 表示機能から削除します。

WTOR 処理機能には、以下の項目が含まれます。

- 自動化ルーチン AOFRSA0E (186 ページの『AOFRSA0E』を参照)
- システム・メッセージ IEE400I と IEE600I 用の自動化テーブル項目

## AMRF バッファ不足の処理

提供された AMRF バッファ不足の処理機能は処置メッセージ保存機能 (AMRF) のバッファ不足を報告します。この機能は、バッファ不足の自動処理のために構成ファイルからコマンドを実行します。

提供された AMRF バッファ不足の処理機能には、以下の項目が含まれます。

- 自動化ルーチン AOFRSA0G (187 ページの『AOFRSA0G』を参照)
- システム・メッセージ IEA359E、IEA360A と、IEA361I 用の自動化テーブル項目
- MESSAGES/USER DATA 自動化ポリシー項目の、以下の特別メッセージ ID に対するコマンドの指定
  - AMRFSHORT
  - AMRFFULL
  - AMRFCLEAR

## JES2 シャットダウン前のドレーン処理

SA z/OS は、JES2 シャットダウン前のドレーン処理のための機能を提供します。

提供された JES2 ドレーン処理機能には、以下の項目が含まれます。

- 自動化ルーチン AOFRSD07、AOFRSD0F、AOFRSD0G。188 ページの『AOFRSD07』、190 ページの『AOFRSD0F』および 192 ページの『AOFRSD0G』を参照してください。
- システム・メッセージ HASP607 用の自動化テーブル項目。
- JES2 DRAIN 自動化ポリシー項目でのドレーンされる JES2 アプリケーションの指定、および JES2 シャットダウン前のドレーン方法の指定。

## IMS トランザクション・リカバリー

SA z/OS は、IMS トランザクション・リカバリー機能を提供します。この機能は、リカバリー目的のために構成ファイルから事前定義された応答またはコマンドを発行することによって、IMS アプリケーション・プログラムの異常終了メッセージに応答します。このプログラムがリカバリー処理から除外されている場合、あるいは着信メッセージのオカレンスが、事前定義されたクリティカル (critical) しきい値を超えている場合、リカバリー・アクションは実行されません。

SA z/OS が提供する IMS トランザクション・リカバリー機能には、以下の項目が含まれます。

- アプリケーション・プログラム異常終了メッセージ用の NetView 自動化テーブル項目、DFS554A
- 異常終了メッセージを発行するサブシステムは、以下の自動化ポリシー定義を持っています。
  - MINOR RESOURCE ポリシー項目内の、マイナー・リソース PROG.progid または TRAN.tran のエラーしきい値定義

- MESSAGES/USER DATA ポリシー項目のメッセージ・タイプ ABCODEPROG.*progid*、ABCODEPROG、ABCODETRAN.*tran*、または ABCODETRAN に対するコード定義
- MESSAGES/USER DATA ポリシー項目のメッセージ ID DFS554A に対する応答またはコマンドの指定

## INGWHY ユーザー・アクションの定義

---

INGWHY を使用すれば、リソースを使用可能/使用不可にすることが望ましい理由や、希望する状況に自動化を移行させることができない理由を見つけるために最初に状態を分析することができます。

自動化では、可能性のある理由が分析されるだけでなく、さらに詳しい分析に役立つ可能性のあるコマンドを一般的に指し示すアクションも処理されます。自動化管理者は、さまざまな理由により、そのアクションを適応させたりオーバーライドしたりしなければならない場合があります。

- アクションはインストール済み環境 (サンドボックスなのか実動システムなのか) によって異なります。
- アクションはリソース (タイム・クリティカル・リソースなのか非タイム・クリティカル・リソースなのか) によって異なります。
- アクションは時間帯 (勤務時間計画) によって異なります。
- アクションは、リソース固有のアクションを含む問題データベースが既に存在するかどうかによって異なります。

INGWHY は DSIPARM メンバー INGWHYSA および INGWHYU からアクション・テキストを読み取ります。

- メンバー INGWHYSA には、提案されたアクション (製品に付属します) が含まれています。このメンバーはユーザーが変更してはなりません。
- ユーザーはメンバー INGWHYU を使用して、INGWHYSA で定義されているアクションをオーバーライドできます。

INGWHY は最初に INGWHYU を処理します。INGWHY は、メンバー INGWHYU において一致するアクション ID を検出すると、メンバー INGWHYSA において再びそのアクション ID を検索することはしません。

注: INGWHYSA と INGWHYU にはロジック・モードの NetView Data REXX が含まれています。これらのファイルは、NetView で NOINCL オプションを指定したブラウザ・コマンドを使用して表示できます。

### サンプル・シナリオ

リソースが BROKEN 状態になっていることが判明しました。標準のアクションを拡張する必要があります。

1. SAPHOST\_CTL/APL/AOCA は既に INGWHY によって分析されています。

```

INGKYAN2                      SA z/OS - Command Dialogs          Line 1    of 14
Domain Id   : IPUFA             ----- INGWHY -----          Date     : 02/07/17
Operator Id : OPER1            Sysplex = TSAPLEX          Time     : 19:05:50

Analyzed Resource:  SAPHOST CTL/APL/TSA1                      Job Name: SAPHCTL
Status Compound:   INHIBITED      Desired: AVAILABLE      Observed:
SOFTDOWN

-----

SITUATION:
SAPHOST_CTL/APL/AOCA is not in its desired status and
automation is unable to proceed.

REASON 2 OF 2:
SAPHOST_CTL/APL/AOCA has a dependency on SAPHOST_EXE/APL/AOCA.
SAPHOST_EXE/APL/AOCA ..
..could not be started.
..is in a PROBLEM status and requires operator intervention.
..is in the agent status 'BROKEN'.
..may have received a non-recoverable error.
..is desired to be AVAILABLE.

ACTION 2 OF 2:                                     INGWHYSA(A0209600)
Refer to your company's rules in order to take the appropriate
action.
You may contact the owner that is responsible for SAPHOST_EXE/APL/AOCA.
Consider the following commands to apply to SAPHOST_EXE/APL/AOCA:
- EXPLAIN
- SETSTATE

Command
===>-----

F1=Help      F2=End      F3=Return   F4=Exit     F5=Delete
F6=Roll      F7=Home     F8=Forward  F9=Refresh  F10=Previous F11=Next    F12=Retrieve

```

図 32. シナリオ: 提案されたアクションをオーバーライドする

2. アクション 2 では、アクションを拡張する必要があることが提言されています。

このアクションはメンバー名 INGWHYSA およびアクション ID A0209600 で識別されます。この情報はアクション・タイトル行の末尾にあります。

```
ACTION 2 OF 2:                                     INGWHYSA(A0209600)
```

INGWHYSA は DSIPARM メンバーの名前です。INGWHY はそこからアクション・テキストを読み取ります。アクション・テキストは、このファイル内で A0209600 によって識別されます。

3. ISPF でメンバー INGWHYSA を表示し、アクション ID A0209600 を見つけます。

```

/*=====*/
/* Action   : A0209600                                     */
/*          : Resource is in agent state BROKEN.           */
/*          : The Desired state is AVAILABLE.              */
/*=====*/
IF #action = 'A0209600' THEN DO
'Refer to your company's rules to take the appropriate action.'
'You may contact the owner that is responsible for '#sub_res'.'
'Consider the following commands to apply to '#sub_res': '
'- EXPLAIN '
'- SETSTATE '
END

```

コメントおよび完全な 'IF ... THEN DO ... END' セクションをクリップボードにコピーします。

4. ISPF でメンバー INGWHYU を編集し、クリップボードの内容をファイルの末尾において以下の行間に貼り付けます。

```

/* ***** */
/* Start coding your action overrides here...          */
/* End of INGWHYU                                     */
/* ***** */

```

5. INGWHYU の初めで説明されているように INGWHYU を編集してテストします。

**重要:** INGWHYU は NetView によって Data REXX コードとして処理されます。そのため、INGWHYU は ロジック・モード の Data REXX 構文に従わなければなりません。

Data REXX デイレクティブ: /\*%LOGIC。『*Programming: REXX and the NetView Command List Language*』の「[Using Data REXX](#)」を参照してください。

## ループ・アドレス・スペース抑制

ループ・アドレス抑制は、特殊 APL として実装されます。提供されているサンプル自動化ポリシーからインポートできます。

インポート後、TEMS サーバーの定義を追加し、APL 用にいくつかのカスタマイズを行ってから、APG とともに APL が実行されるシステムに APL をリンクする必要があります。

SA z/OS は、APL を含む SOCNTL ファイルをロードする際に、ループ・アドレス・スペース抑制プロシージャを活動化します。

### 準備

プロシージャを実行するシステムに、OMEGAMON for z/OS をインストールし、稼働させておく必要があります。

OMEGAMON for z/OS と通信している TEMS サーバーを稼働中にしておく必要があります。

以下を表示できることを確認します。

- OMEGAMON for z/OS
- TEMS に接続された TEP からの OMEGAMON データ

以下が分かっている必要があります。

- TEMS へのアクセス権限を持つユーザー ID およびパスワード
- TEMS 上で稼働する SOAP サーバーの IP アドレスおよびポート番号

通信に HTTPS を使用する場合は、TCPIP 下で HTTPS を予めセットアップしておく必要があります (『*IBM Z System Automation* 計画とインストール」の『*従来の SA z/OS 構成*』で、Tivoli Enterprise Portal の構成手順を参照してください。)

### サンプル・ポリシーのコピー

\*ITM アドオン全体を PDB にコピーするだけで、新しい PDB を作成できます。

#### このタスクについて

\*ITM アドオン内の一部またはすべてのエレメントを含む既存の PDB がある場合は、さらに以下のような選択を行うことができます。

#### 手順

1. 通常 PDB のバックアップを取得し、カスタマイズ・ダイアログで開きます。
2. 「データ管理」(オプション 5) に移動し、「アドオンのインポート (Import for Add On)」(オプション 2) を選択します。
3. \*ITM アドオンに対して C (カスタマイズ) を入力し、コンポーネント選択パネルに移動します。
4. コンポーネントのリストが表示されます。M (reMove) および S (Select) を使用して、「モニタリング分析」コンポーネントのみが選択されるようにしてください。
5. PF3 を押し、1 を選択して、インポートするデータを表示します。オプション 3 を選択して、インポートを実行します。

## タスクの結果

通知メッセージがいくつか表示されてから、Import Successful が表示されます。その後で、メインメニューにアクセスするには、PF3 を 2 回押します。

## SOAP サーバーのカスタマイズ

### 手順

1. PDB を開いて NTW (ネットワーク) を選択します。
2. 新しい TEMSserver を入力します。TEMSserver は、OMEGAMON データを提供する TEMS サーバーを識別するための固有の名前です。
3. Copy を使用して、定義を SOAP\_SERVERS ネットワーク定義からコピーします。
4. SOAP SERVER ポリシーを選択して、HUBTEMS 項目を編集します。
5. Host 名およびプロトコルを入力します。  
HTTPS をプロトコルとして使用している場合は、ポート番号を 3661 に変更する必要があります (この番号とポート番号 1920 のみがデフォルト値ですが、ご使用の環境では変更されている可能性があります)。
6. SAFPW の User ID および Password を指定します。  
これにより、パスワードを提供するために NetView で INGPW ユーティリティーを使用することを System Automation に指示します。
7. PF3 を 2 回押してポリシー・パネルに戻り、WHERE USED を選択します。
8. この TEMS サーバーを経由して OMEGAMON データを受け取るシステムを選択します。  
各システムにリンクされている、HUBTEMS SOAP サーバーが定義された単一ネットワークが 1 つのみ存在することが重要です。
9. PF3 を押して項目選択パネルに戻ります。

## APL の構成

### サンプルのコピー

APL 項目タイプを選択します。

new LOOPSUPP\_identity を入力して、新しい APL を作成します。ID は TEMS 名またはシスプレックス名 (同じ TEMS に送信する複数のシスプレックスからのシステムがある場合) にすることができます。後者にした場合は、シスプレックスごとに異なるポリシーのセットを指定する必要があります。

この APL に LOOPSUPP という SUBSYSTEM 名を付け、Enter を押し、次に PF3 を押します。

COPY ポリシーを使用して、データを LOOPSUPP APL からコピーします。

### マイナー・リソース

MONITOR という名前の定義済みのマイナー・リソースがあります。これは、APL が使用可能な場合にモニターの実行を制御します。デフォルトでは、このリソースの RECOVERY フラグは LOG に設定されています。これは、中断を伴うコマンドは、実行されずに Netlog に書き込まれることを意味します。中断を伴わないリカバリー・アクション WARN および DIAG は、この場合も実行されます。

初めてのデプロイ時には、この値を LOG に設定しておいた方がよいと思われませんが、以降のデプロイでは、アクションを使用可能にしてこの値を YES に設定することもできます。

### MESSAGES/USER DATA

この下には、2 つのカテゴリー (INGCATEGORY および INGRECOVERY) が定義されています。

- INGCATEGORY: アドレス・スペースのカテゴリー化規則を追加するには、変更する必要があります。
- INGRECOVERY: 該当するアクションをカテゴリー化に使用します。

すべての LOOPSUPP モニターの定義を、C\_LOOPSUPP APL クラスを使用して継承することができます。

### カテゴリー規則の追加

MESSAGES/USER DATA ポリシーの下で、INGCATEGORY に対して K を入力し、Enter を押します。

一番下の行 (未変更のサンプルを使用している場合は SYSOTHER) に対して I を入力し、Enter を押します。

これにより、以下のコードをこのパネルで入力できるようになります。

コード	説明
CODE1	アドレス・スペース WLM サービス・クラス。
CODE2	アドレス・スペースのジョブ・タイプ (STC、BATCH、TSO)。この値には、APL として SA z/OS に定義されたアドレス・スペースを表す接尾部 <code>_SA</code> が付きます。
CODE3	アドレス・スペースの <code>jobname.stepname</code> の値。コード・フィールドは長さが 15 文字しかないため、ステップ名の最後の数文字が重要となる検査を行う場合は、ワイルドカードを使用しなければならないことに注意してください。

コード・マッチングの出力はリカバリー・カテゴリーであり、これは WLM サービス・クラスの名前にデフォルト設定されます。

サンプル・ポリシーによって、SA z/OS に認識されているすべてのリソース、およびすべての SYSOTHER リソースが WARN リカバリー・クラスに入れられ、すべての TSO ユーザーが TSO\_USER リカバリー・カテゴリーに入れられます。

完了したら PF3 を押して終了します。

### リカバリー・シーケンスの追加

リカバリー・シーケンスをすべてのループ・アドレス・スペース・モニターで汎用的に使用できるようにしたい場合は、リカバリー・シーケンスを C\_LOOPSUPP APL クラスに追加する必要があります。リカバリー・シーケンスを APL に追加すると、リカバリー・シーケンスと C\_LOOPSUPP INGRECOVERY データの間のリンクを切断することになるため、このクラスに対するこれ以降の変更はリカバリー・シーケンスでは取得されません。

コード・マッチングに渡されるコードは以下のとおりです。

コード	説明
CODE1	INGCATEGORY の結果である、リカバリー・カテゴリーの名前。
CODE2	アドレス・スペースの現行リカバリー・パス。アドレス・スペースがループとして報告されないモニター・サイクルがある場合、またはモニター APL が停止された場合、リカバリー・パスはリセットされます。通常は数値ですが、* を指定すると、すべてのパスで対応が発生します。
CODE3	<ブランク>

出力は、以下の 1 つ以上のリカバリー・アクションのリストです。

アクション	説明
NONE	何もありません。
WARN	警告メッセージ ING601E を発行します。メッセージはネットログに書き込まれ、また警告メッセージを受け取るクラス 40 および 44 の代わりに通知を受けるようにセットアップされた通知オペレーターに書き込まれます。

アクション	説明
	<p>警告メッセージは、MVSESA リソース下の SDF に送られます。ジョブが検出されたパスの数によって、以下のようにメッセージの重大度が決まります。</p> <p>&lt;10 -&gt; UNUSUAL &lt;20 -&gt; IMPORTANT 20 以上 -&gt; CRITICAL</p> <p>リソースがこれ以上検出されない場合、またはモニターが停止された場合、メッセージは消去されます。</p>
DIAG	<p>OMEGAMON 検査ツールを使用して、アドレス・スペースに関する診断を取得し、診断レポートをネットログに書き込みます。これは、コード内のどこでループが発生しているかを識別するために役立ちます。</p>
STOP	<p>アドレス・スペースに MVS STOP コマンドを発行します。あるいは、アドレス・スペースが TSO USER の場合は、MVS CANCEL USER コマンドが発行されます。SA z/OS に定義されているアドレス・スペースで使用される場合でも、STOP コマンドは発行されますが、INGREQ STOP 要求には変換されません。これにより、SHUT_ コマンドを試行後、直接シャットダウン・シーケンスをパス上にコード化できます。</p>
CANCEL	<p>アドレス・スペースに MVS CANCEL コマンドを発行します。あるいは、アドレス・スペースが TSO USER の場合は、MVS CANCEL USER コマンドが発行されます。SA z/OS に定義されているアドレス・スペースで使用される場合でも、CANCEL コマンドは発行されますが、INGREQ STOP 要求には変換されません。これにより、SHUT_ コマンドを試行後、直接シャットダウン・シーケンスをパス上にコード化できます。</p>
FORCE	<p>アドレス・スペースに MVS FORCE, ARM コマンドを発行します。あるいは、アドレス・スペースが TSO USER の場合は、MVS CANCEL USER コマンドが発行されます。SA z/OS に定義されているアドレス・スペースで使用される場合でも、FORCE コマンドは発行されますが、INGREQ STOP 要求には変換されません。これにより、SHUT_ コマンドを試行後、直接シャットダウン・シーケンスをパス上にコード化できます。</p>
SHUT_NORM	<p>SA z/OS に定義されたアドレス・スペースに、シャットダウン・タイプ NORM 用の INGREQ STOP を発行します。SA z/OS に定義されていないアドレス・スペースで使用される場合は、STOP 命令に変換されます。</p>
SHUT_IMMED	<p>SA z/OS に定義されたアドレス・スペースに、シャットダウン・タイプ IMMED 用の INGREQ STOP を発行します。SA z/OS に定義されていないアドレス・スペースで使用される場合は、CANCEL 命令に変換されます。</p>
SHUT_FORCE	<p>SA z/OS に定義されたアドレス・スペースに、シャットダウン・タイプ FORCE 用の INGREQ STOP を発行します。SA z/OS に定義されていないアドレス・スペースで使用される場合は、FORCE 命令に変換されます。</p>
SUSPEND	<p>アドレス・スペースに MVS RESET, QUIESCE を発行します。これにより、アドレス・スペースに対するこれ以降の CPU ディスパッチングは行われませんが、アドレス・スペースは停止されません。</p>
RESET_class	<p>アドレス・スペースに MVS RESET, SRVCLASS=class を発行します。これにより、アドレス・スペースの WLM ディスパッチング・クラスを、可能であれば、より CPU 使用量が少ないクラスに変更します。アドレス・スペースをクラス SYSOTHER に変更すると、そのディスパッチング優先順位は非常に低い値になります。</p>



アクション	説明
USER_prog	<p>prog パラメーターは、リカバリー・プロセス中に呼び出されるユーザー・プログラム (検出された場合) の名前です。</p> <p>ルーチンに渡されるパラメーターは次のとおりです。</p> <pre>asid jobname jobid jobstep cpuindex</pre>

サンプル・ポリシーには、以下の4つのリカバリー・カテゴリがあります。

リカバリー・カテゴリ	説明
IGNORE	モニターでジョブに何も行わず、ジョブに関して誰にも通知しません。既知の誤検出でを使用することを目的としています。
WARN	すべてのパスに関して警告メッセージを出力し、2番目のパスに関して診断を収集します。
SUSPEND	すべてのパスに関して警告メッセージを出力し、2番目のパスに関して診断を収集し、4番目のパスに関してアドレス・スペースに SUSPEND を発行します。アドレス・スペースの中断に関するリマインダーはありません。ログを確認する必要があります。
TSO_USER	すべてのパスに関する警告メッセージの発行のみを行います。

こちらも、完了したら PF3 を押して終了します。

## リンケージ

カテゴリ化およびリカバリー・ポリシーが定義されたため (後で戻って調整が可能)、APL を実行する1つ以上のシステムに APL をリンクする必要があります。

APL がリンクされるシステムには、HUBTEMS という名前の SOAP サーバーを定義しているネットワーク (NTW) がリンクされている必要があります、そうでないとモニターは機能しないことに注意してください。

これを行うには、自動化名がブランクの APG を作成し、LOOPSUPP\_identity APL をその APG のメンバーにし、モニターを実行するシステムにその APG をリンクします。必要な場合は、ING\_ANALYTIC APG で APG をモデル化できます。

## ビルド

SOCNTL ファイルを作成し、ターゲット・システムで活動化できるようにする必要があります。

## 資格情報の事前準備

ここでは、コマンドを発行して、モニターが TEMS サーバーへのアクセスに使用するパスワードをセットアップする必要があります。

モニターを実行する各システムで自動化エージェントにログオンし、次のコマンドを発行します。

```
NETVASIS INGPW userid SOAP,INIT=password
```

従って、ユーザー ID が AutoAgnt、パスワードが ABC1234d の場合は、次のように発行します。

```
NETVASIS INGPW AutoAgnt SOAP,INIT=ABC1234d
```

後でパスワードを変更する場合は、このコマンドを再発行して、System Automation に新規パスワードが何であるかを通知する必要があります。

## モニターの活動化

INGAMS REFRESH を発行して新しい SOCNTL ファイルを選択すると、モニターが実行を開始します。

## インストールの検証

5 秒待ってからネットログを確認してください。ING600I メッセージを探してください。

これは、プロシージャーで作成されるモニター・レポートです。モニター・レポートは、照会が正常に行われたことを示し、検出されたすべてのループ・アドレス・スペースの詳細情報が含まれている場合と含まれていない場合があります。

## AOFRSA01

### 目的

AOFRSA01 自動化ルーチンは、logrec データ・セットがフルに近い、またはフルであることを示す、システムからのメッセージに応答するために使用できます。応答するには、logrec データ・セットの内容をダンプおよびクリアするコマンドを構成ファイルから発行します。

AOFRSA01 は、これらの着信 logrec データ・セット・メッセージを記憶し、その発生と事前定義のしきい値の低頻度 (infrequent) レベル、高頻度 (frequent) レベルと、クリティカル (critical) レベルを比較します。しきい値を超えた場合は、構成ファイルの項目/タイプのペアである MVSESA/LOGREC のコマンド項目中の最初のフィールドに従って、適切なコマンドを選択するためのオプションとして使用します。しきい値を超えていない場合は、選択オプション ALWAYS 用に定義されたコマンドが実行されます。

AOFRSA01 は、NetView 自動化テーブルから呼び出す必要があります。

### 構文

▶ AOFRSA01 ◀

### 制約事項

- AOFRSA01 のアクションが実行されるのは、LOGREC の自動化フラグがオンになっている場合だけです。
- AOFRSA01 の処理が実行されるのは、予想通りのメッセージ IFB040I、IFB060E、IFB080E、または IFB081I のいずれかによってそれが NetView 自動化テーブルから呼び出された場合だけです。
- LOGREC データ・セットをダンプしてクリアするコマンドが自動化ポリシーから出されるのは、LOGREC リカバリー機能がもう進行中ではない場合だけです。

### 使用法

自動化ルーチン AOFRSA01 は、以下のメッセージに応答するためのものです。

```
IFB040I SYS1.LOGREC AREA IS FULL, hh.mm.ss
IFB060E SYS1.LOGREC NEAR FULL
IFB080E LOGREC DATA SET NEAR FULL, DSN=dsname
IFB081I LOGREC DATA SET IS FULL, hh.mm.ss, DSN=dsn
```

出すコマンドは、構成ファイル中の MVSESA/LOGREC 項目/タイプのペアのコマンド項目から選択されます。

期待したメッセージの 1 つを受信したときにしきい値に達しない場合は、選択オプションなしの項目に対するコマンド、および選択オプション ALWAYS に対するコマンドがすべて選択されます。低頻度 (infrequent) レベルのしきい値を超えた場合は、選択指定オプションなしの項目に対するコマンド、および選択オプション INFR に対するコマンドがすべて選択されます。同様に、高頻度 (frequent) レベルは選択オプション FREQ と対応し、クリティカル (critical) レベルは選択オプション CRIT と対応しています。

自動化ルーチン AOFRSA02 は、LOGREC リカバリー機能の完了を示すために、NetView 自動化テーブルからメッセージ IFC001I によって出されていることを確認してください。

## グローバル変数

### &EHKVAR1

LOGREC データ・セットの内容をダンプしてクリアするコマンドの構成ファイルでの定義時に、変数 &EHKVAR1 を LOGREC データ・セットの名前に使用できます。この変数は、LOGREC データ・セット名の完全なデータ・セット名で置換されます。

## AOFRSA02

### 目的

AOFRSA02 自動化ルーチンは、LOGREC リカバリー機能が進行中であることを示すフラグをリセットするための LOGREC データ・セットの初期化メッセージに応答するために使用できます。

AOFRSA02 は、NetView 自動化テーブルから呼び出す必要があります。

### 構文

▶▶ AOFRSA02 ◀◀

### 制約事項

- AOFRSA02 のアクションが実行されるのは、LOGREC の自動化フラグがオンになっている場合だけです。
- AOFRSA02 の処理が実行されるのは、それが NetView 自動化テーブルから呼び出された場合だけです。

### 使用法

自動化ルーチン AOFRSA02 は、次のメッセージに応答するためのものです。

```
IFC001I D=devtyp N=x F=track1* L=track2* S=recd** DIP COMPLETE
```

これは、LOGREC データ・セットの初期化時に生成され、データ・セットの限界を示しています。

(LOGREC リカバリー機能が進行中であることを示している) フラグは自動化ルーチン AOFRSA01 によって使用されます。

### 例

この例は、LOGREC データ・セット処理のサンプル・シナリオを示しています。

NetView 自動化テーブルの次の項目は、期待されるメッセージの 1 つを受信したときに適切な自動化ルーチンを実行するために、自動的に作成されます。

```
IF MSGID = 'IFB040I' | MSGID = 'IFB060E' |
   MSGID = 'IFB080I' | MSGID = 'IFB081I'
THEN
EXEC(CMD('AOFRSA01')ROUTE(ONE %AOFOPRECOPER%));

IF MSGID = 'IFC001I'
THEN
EXEC(CMD('AOFRSA02')ROUTE(ONE %AOFOPRECOPER%));
```

COMMANDS HELP	
-----	
Thresholds Definition	
Command ==> _____	
Entry Type : MVS Component	PolicyDB Name : DATABASE_NAME
Entry Name : MVS_COMPONENTS	Enterprise Name : YOUR_ENTERPRISE
Resource : MVSESA.LOGREC	
Critical Number . . . . 3	(1 to 50)
Critical Interval . . . . 00:05	(hh:mm or hhmm, 00:01 to 24:00)
Frequent Number . . . . 3	(1 to 50)
Frequent Interval . . . . 00:30	(hh:mm or hhmm, 00:01 to 24:00)
Infrequent Number . . . . 3	(1 to 50)
Infrequent Interval . . . . 24:00	(hh:mm or hhmm, 00:01 to 24:00)

図 33. MVS コンポーネント LOGREC に対するしきい値定義

Cmd PS/Select AutoFn/*	Command Text
_____	MVS S CLRLOG,DSN=&EHKVAR1
_____	_____

図 34. 項目/タイプのペア MVSESA/LOGREC の MESSAGES/USER DATA ポリシー項目

ある日、次のメッセージが初めて送られてきたとします。

```
IFB080E LOGREC DATA SET NEAR FULL, DSN=SYS1.AOC1.MAN3
```

定義されたどのしきい値も超えていないので、自動化ルーチン AOFRSA01 は、実行される選択オプションなしの定義されたコマンドおよび選択オプション ALWAYS に対する定義されたコマンドを検索します。前記の制御ファイルによって、コマンド MVS S CLRLOG,DSN=&EHKVAR1 が選択されます。このコマンドを出す前に、変数 &EHKVAR1 は受信したメッセージのデータ・セット名に置き換わり、MVS S CLRLOG,DSN=SYS1.AOC1.MAN3 となります。

メッセージ IFB080E が送信され続け、期待するメッセージの頻度が低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値を超えると、自動化ルーチン AOFRSA01 は実行する選択オプションなしの定義されたコマンドおよび選択オプション INFR、FREQ、または CRIT に対する定義されたコマンドを探します。

任意のオプションを指定したコマンドは定義されていないので、前述のケースと同様に、選択オプションなしの定義されたコマンドが選択されて出されます。

メッセージ AOF589I、AOF588I、または AOF587I は、低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値を超えた場合に出されます。これらのメッセージは、低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値のアクションが処理されたことを示します。

期待したエラー・メッセージの受信時に、LOGREC データ・セットのリカバリー処理がまだ進行中である場合は、次のメッセージが出されます。

```
AOF585I 15:45 : RECOVERY OF LOGREC IS ALREADY IN PROGRESS -
```

LOGREC データ・セットが初期化されてしまったことを示しているメッセージ IFC001I を受信すると、リカバリー処理は終了しているとみなされます。

## AOFRSA03

### 目的

AOFRSA03 自動化ルーチンは、フル SMF データ・セットまたはシステムからのスイッチ・メッセージに回答するために使用できます。AOFRSA03 は、SMF データ・セットの内容をダンプしてクリアするコマンドを構成ファイルから出します。

AOFRSA03 は、着信 SMF データ・セット・メッセージをトラッキングし、そのオカレンスと低頻度 (infrequent)、高頻度 (frequent)、およびクリティカル (critical) レベルの事前に定義されたしきい値を比較します。しきい値を超えた場合は、構成ファイルの項目/タイプのペアである MVSESA/SMFDUMP のコマンド項目中の最初のフィールドに従って、適切なコマンドを選択するためのオプションとして使用します。しきい値を超えていない場合は、選択オプション ALWAYS 用に定義されたコマンドが実行されます。

AOFRSA03 は、NetView 自動化テーブルから呼び出す必要があります。

このような自動化に加えて、SA z/OS は SA z/OS がアクティブでなかったときに満杯になったフル SMF データ・セットも検査します。選択オプションが 'ALWAYS' に設定されている場合は、ダンプが必要なデータ・セットごとにコマンドが発行されます。&EHKVAR には完全修飾の SMF データ・セット名が含まれます。

## 構文

▶▶ AOFRSA03 ◀◀

## 制約事項

- AOFRSA03 の処理が実行されるのは、期待されるメッセージ IEE362A、IEE262I、IEE391A または IEE392I のいずれかによって NetView 自動化テーブルから AOFRSA03 が呼び出された場合です。
- 上記のいずれかのメッセージによって AOFRSA03 がトリガーされた場合、アクションが実行されるのは、SMFDUMP のリカバリー自動化フラグがオンになっている場合だけです。
- AOFRSA03 のアクションが実行されるのは、SMFDUMP のリカバリー自動化フラグがオンになっている場合だけです。
- AOFRSA03 の処理が実行されるのは、AOFRSA03 が、期待されるメッセージ IEE362A、IEE262I、IEE391A または IEE392I のいずれかによって NetView 自動化テーブルから呼び出された場合だけです。

## 使用法

自動化ルーチン AOFRSA03 は、以下のメッセージに応答するためのものです。

```
IEE362A SMF ENTER DUMP FOR dsname ON ser
IEE362I SMF ENTER DUMP FOR dsname ON ser
IEE391A SMF ENTER DUMP FOR DATA SET ON VOLSER ser, DSN=dsname
IEE392I SMF ENTER DUMP FOR DATA SET ON VOLSER ser, DSN=dsname
```

上記は、SMF データ・セットのダンプの準備ができていることを示しています。

## グローバル変数

### &EHKVAR1

SMF データ・セットの内容をダンプしてクリアするコマンドの構成ファイルでの定義時に、変数 &EHKVAR1 を SMF データ・セットの名前に使用できます。メッセージ IEE391A または IEE392I の受信時に、この変数は AOFRSA03 によって完全なデータ・セット名で置換されます。メッセージ IEE362A または IEE362I の場合は、この変数は MANn (SMF データ・セット名の 2 番目の部分) で置換されることとなります。

### &EHKVAR2

SMF データ・セットの内容をダンプしてクリアするコマンドを構成ファイルで定義する際に、変数 &EHKVAR2 を SMF データ・セットの名前に使用できます。メッセージ IEE391A、IEE392I、IEE362A、IEE362I の受信時に、この変数は AOFRSA03 によって完全なデータ・セット名で置換されます。

## 例

この例は、AOFRSA03 が自動化テーブルから呼び出された場合の SMF データ・セットの処理を示します。

NetView 自動化テーブルの次の項目は、期待されるメッセージの 1 つを受信したときに適切な自動化ルーチンを実行するために、自動的に作成されます。

```
IF (MSGID = 'IEE362I' | MSGID = 'IEE362A' |
    MSGID = 'IEE391A' | MSGID = 'IEE392I')
THEN
EXEC(CMD('AOFRSA03')ROUTE(ONE %AOFOPRECOPER%));
```

COMMANDS HELP

```
-----
                                Thresholds Definition
Command ==> _____
Entry Type : MVS Component          PolicyDB Name : DATABASE_NAME
Entry Name : MVS_COMPONENTS        Enterprise Name : YOUR_ENTERPRISE

Resource   : MVSESA.SMFDUMP

Critical Number . . . . 3_          (1 to 50)
Critical Interval . . . 00:05       (hh:mm or hhmm, 00:01 to 24:00)

Frequent Number . . . . 3_          (1 to 50)
Frequent Interval . . . 00:30       (hh:mm or hhmm, 00:01 to 24:00)

Infrequent Number . . . 3_          (1 to 50)
Infrequent Interval . . 24:00       (hh:mm or hhmm, 00:01 to 24:00)
```

図 35. MVS コンポーネント SMFDUMP に対するしきい値定義

```
Cmd PS/Select AutoFn/* Command Text
_____ MVS S SMFDUMP1,DA='&EHKVAR2'
```

図 36. 項目/タイプのペア MVSESA/SMFDUMP の MESSAGES/USER DATA ポリシー項目

ある日、次のメッセージが初めて送られてきたとします。

```
IEE391A SMF ENTER DUMP FOR DATASET ON VOLSER 123, DSN=SYS1.AOC1.MAN3
```

定義済みしきい値はどれも超えていないので、AOFRSA03 自動化ルーチンは、実行する、選択オプションなしで定義されたコマンドまたは選択オプション ALWAYS で定義されたコマンドを検索します。前記の制御ファイルによって、コマンド MVS S SMFDUMP1,DA='&EHKVAR2' が選択されます。このコマンドを実行する前に、変数 &EHKVAR2 は受信したメッセージのデータ・セット名に置き換えられ、MVS S SMFDUMP1,DA='SYS1.AOC1.MAN3' となります。

メッセージ IEE391A が継続して受信され、期待するメッセージのオカレンスが低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値を超えると、AOFRSA03 自動化ルーチンは、実行する、選択オプションなしで定義されたコマンドまたは選択オプション INFR、FREQ または CRIT で定義されたコマンドを検索します。

選択オプションを指定したコマンドは定義されていないので、前述の場合と同様に、選択オプションなしで定義されたコマンドのみが選択され、実行されます。

メッセージ AOF589I、AOF588I または AOF587I は、低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値を超えた場合に出されます。これらのメッセージは、低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値のアクションが処理されたことを示します。

## AOFRSA08

### 目的

AOFRSA08 自動化ルーチンは、キューに入れられた syslog を保管するために、外部書き出しプログラムを開始することによって、メッセージがキューに入っている syslog に応答するために使用できます。

AOFRSA08 は、受信した syslog 待機メッセージを記憶し、そのオカレンスと低頻度 (infrequent)、高頻度 (frequent)、およびクリティカル (critical) レベルの事前に定義されたしきい値を比較します。しきい値を

超えた場合は、構成ファイルの項目/タイプのペアである MVSESA/SYSLOG のコマンド項目中の最初のフィールドに従って、適切なコマンドを選択するためのオプションとして使用します。しきい値を超えていない場合は、選択オプション ALWAYS 用に定義されたコマンドが実行されます。

AOFRSA08 は、NetView 自動化テーブルから呼び出す必要があります。

## 構文

▶▶ AOFRSA08 ◀◀

## 制約事項

- AOFRSA08 の処理が実行されるのは、それが予想通りのメッセージ IEE043I によって NetView 自動化テーブルから呼び出された場合だけです。
- AOFRSA08 のアクションが実行されるのは、SYSLOG の自動化フラグがオンになっている場合と、JES の状況が UP または HALTED になっている場合だけです。

## 使用法

自動化ルーチン AOFRSA08 は、次のメッセージに応答するためのものです。

```
IEE043I A SYSTEM LOG DATA SET HAS BEEN QUEUED TO SYSOUT CLASS class
```

これは、システムがシステム・ログ (SYSLOG) データ・セットをクローズし、データ・セットを SYSOUT クラスのキューに入れたことを示しています。

出すコマンドは、構成ファイル中の MVSESA/SYSLOG 項目/タイプのペアのコマンド項目から選択されます。

期待したメッセージの 1 つを受信したときにしきい値に達しない場合は、選択オプションなしの項目に対して定義されたコマンド、および選択オプション ALWAYS に対するコマンドがすべて選択されます。

低頻度 (infrequent) レベルのしきい値を超えた場合は、選択指定オプションなしの項目に対して定義されたコマンド、および選択オプション INFR 付きの項目に対して定義されたコマンドがすべて選択されます。

同様に、高頻度 (frequent) レベルは選択オプション FREQ と対応し、クリティカル (critical) レベルは選択オプション CRIT と対応しています。

## 例

この例は、SYSLOG 処理のサンプル・シナリオを示しています。

NetView 自動化テーブルの次の項目は、着信 IEE043I メッセージへの応答として AOFRSA08 を発行するために、自動的に作成されます。

```
IF MSGID = 'IEE043I'
THEN
EXEC (CMD('AOFRSA08'))ROUTE(ONE %AOFOPRECOPER%);
```



COMMANDS HELP	
-----	
Thresholds Definition	
Command ==>	_____
Entry Type : MVS Component	PolicyDB Name : DATABASE_NAME
Entry Name : MVS_COMPONENTS	Enterprise Name : YOUR_ENTERPRISE
Resource : MVSESA.SYSLOG	
Critical Number . . . . 3	(1 to 50)
Critical Interval . . . . 00:05	(hh:mm or hhmm, 00:01 to 24:00)
Frequent Number . . . . 3	(1 to 50)
Frequent Interval . . . . 00:30	(hh:mm or hhmm, 00:01 to 24:00)
Infrequent Number . . . . 3	(1 to 50)
Infrequent Interval . . . . 24:00	(hh:mm or hhmm, 00:01 to 24:00)

図 37. MVS コンポーネント SYSLOG に対するしきい値定義

Cmd PS/Select AutoFn/*	Command Text
_____	MVS S SAVELOG
_____	_____

図 38. 項目/タイプのペア MVSESA/SYSLOG の MESSAGES/USER DATA ポリシー項目

ある日、次のメッセージが初めて送られてきたとします。

```
IEE043I A SYSTEM LOG DATA SET HAS BEEN QUEUED TO SYSOUT CLASS A
```

定義されたどのしきい値も超えていないので、自動化ルーチン AOFRSA08 は、実行される選択オプションなしの定義されたコマンドおよび選択オプション ALWAYS に対する定義されたコマンドを検索します。前記の制御ファイルによって、コマンド MVS S SAVELOG が選択されます。

メッセージ IEE043I が送信され続け、期待するメッセージの頻度が低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値を超えると、自動化ルーチン AOFRSA08 は選択オプションなしの定義されたコマンドおよび選択オプション INFR、FREQ、または CRIT に対する定義されたコマンドを探します。

任意のオプションを指定したコマンドは定義されていないので、前述のケースと同様に、選択オプションなしの定義されたコマンドが選択されて出されます。

メッセージ AOF589I、AOF588I、または AOF587I は、低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値を超えた場合に出されます。これらのメッセージは、低頻度 (infrequent)、高頻度 (frequent)、またはクリティカル (critical) のしきい値のアクションが処理されたことを示します。

## AOFRSA0C

### 目的

AOFRSA0C 自動化ルーチンは、ダンプをフォーマット設定するため、ダンプ・データ・セットをクリアするため、またはそれ以上のダンプを防ぐためのコマンドを構成ファイルから出すことによって、ダンプ・データ・セット・メッセージに対して行われた SVC ダンプに応答するために使用できます。出すコマンドは、MVSESA/MVSDUMP と MVSESA/MVSDUMPTAKEN 項目/タイプのペアから取得され、受信メッセージの頻度と自動化ポリシーで定義されたしきい値に従って選択されます。コマンド項目の最初のフィールドからは、構成ファイルから適切なコマンドを選択するための詳細基準が得られます。

AOFRSA0C は、NetView 自動化テーブルから呼び出す必要があります。

### 構文

▶▶ AOFRSA0C ◀◀

## 制約事項

- AOFRSA0C のアクションが実行されるのは、MVSDUMP のリカバリー自動化フラグがオンになっている場合だけです。
- AOFRSA0C の処理が実行されるのは、期待するメッセージ IEA611I または IEA911E のいずれかによって NetView 自動化テーブルから呼び出された場合だけです。

## 使用法

自動化ルーチン AOFRSA0C は、以下のメッセージに応答するためのものです。

```
IEA611I {COMPLETE|PARTIAL} DUMP ON dsname
DUMPID=dumpid REQUESTED BY JOB (jobname)
FOR ASIDS(id,id,...)
...

IEA911E {COMPLETE|PARTIAL} DUMP ON SYS1.DUMPnn
DUMPID=dumpid REQUESTED BY JOB (jobname)
FOR ASIDS(id,id,...)
...
```

これらは、システムが完全または部分的な SVC ダンプを、直接アクセス記憶装置またはテープ・ボリューム上の自動的または事前に割り振られたダンプ・データ・セットに書き込んだことを示しています。

AOFRSA0C は、これらのメッセージの受信に関してトラッキングし、受信メッセージの頻度を MVS コンポーネント MVSDUMP に対するしきい値が考慮される低頻度 (infrequent)、高頻度 (frequent)、およびクリティカル (critical) のレベルの事前に定義されたしきい値と比較します。出すコマンドは、受信メッセージの頻度に従って選択されます。

しきい値に達しない場合は、選択オプションなしの項目に対するコマンド、および選択オプション ALWAYS に対するコマンドがすべて選択されます。低頻度 (infrequent) レベルのしきい値を超えた場合は、選択オプションなしの項目に対するコマンド、および選択オプション INFR に対するコマンドがすべて選択されます。同様に、高頻度 (frequent) レベルは選択オプション FREQ と対応し、クリティカル (critical) レベルは選択オプション CRIT と対応しています。

出すコマンドは、これらのメッセージの受信頻度に関して構成ファイルの MVSESA/MVSDUMP 項目/タイプのパアから取得されます。

AOFRSA0C がメッセージ IEA911E の受信時にトリガーされた場合は、クリティカル (critical) のしきい値を超えていない限り、構成ファイルの項目/タイプのパア MVSESA/MVSDUMPTAKEN からすべてのコマンドも選択されて発行されます。

ダンプ処理の完了後に、AOFRSA0C はメッセージ IEF611I と IEF911E の頻度を 15 分間隔でさらにモニターします。頻度が低頻度 (infrequent) のしきい値を下回るとすぐに、MVSESA/MVSDUMPRESET 項目/タイプのパアのすべてのコマンドが出されます。

## グローバル変数

SVC ダンプ・データ・セットを処理するコマンドの構成ファイルでの定義時に、変数 &EHKVAR1 から &EHKVAR6 までを使用して IEA611I または IEA911E の変数の内容で置換できます。変数 &EHKVAR1 から &EHKVAR6 までは、タイプ MVSDUMPRESET のコマンド項目では使用できません。これらの変数は次のように置換されます。

### &EHKVAR1

IEA611I の DSNNAME または IEA911E 内の SYS1.DUMPnn のサフィックス

### &EHKVAR2

データ・セット名

### &EHKVAR3

ダンプ ID

### &EHKVAR4

ジョブ名

### &EHKVAR5

アドレス・スペースの ID

**&EHKVAR6**

ダンプ・タイプ (PARTIAL または COMPLETE)

**例**

この例は、サンプル・コンテキストにおける自動化ルーチン AOFRSA0C の使用を示しています。

NetView 自動化テーブルの項目は、期待するメッセージの 1 つを受信したときに、AOFRSA0C を出すために使用されます。

```
IF MSGID = 'IEA611I' | MSGID = 'IEA911E'
THEN
EXEC(CMD('AOFRSA0C ')ROUTE(ONE %AOFOPRECOPE%));
```

自動化ポリシーで定義されている MVS コンポーネント MVSDUMP に対する 3 レベルのしきい値:

```
AOFKAASR          SA z/OS - Command Dialogs
Domain ID   = IPSNO  ----- INGTHRES -----      Date = 08/28/03
Operator ID = SAUSER                               Time = 09:38:02
```

Specify thresholds and resource changes:

```
Resource   =>  MVSESA.MVSDUMP   Group or specific resource
System     =>  KEY3             System name, domain ID, sysplex name or *all

Critical   =>  6      errors in 00:30   Time (HH:MM)
Frequent   =>  4      errors in 00:20   Time (HH:MM)
Infrequent =>  2      errors in 00:20   Time (HH:MM)
```

Pressing ENTER will set the THRESHOLD values

```
Command ==>
PF1=Help      PF2=End      PF3=Return      PF6=Roll
PF12=Retrieve
```

図 39. MVSDUMP しきい値

項目/タイプのペア MVSESA/MVSDUMP の MESSAGES/USER DATA 自動化ポリシー項目には、異なるレベルの選択オプションがある、メッセージ ID MVSDUMP に対する次のコマンド項目が入っています。

Ps/Select	コマンド・テキスト
FREQ	'MVS DD ALLOC=INACTIVE'
INFR	'MVS DD ALLOC=ACTIVE'
CRIT	'MVS DD ALLOC=INACTIVE'

項目/タイプのペア MVSESA/MVSDUMPTAKEN の MESSAGES/USER DATA 自動化ポリシー項目には、選択オプションがない次のコマンド項目が入っています。

```
'MVS DD CLEAR,DSN=&EHKVAR1'
```

項目/タイプのペア MVSESA/MVSDUMPRESET の MESSAGES/USER DATA 自動化ポリシー項目には、選択オプションがない次のコマンド項目が入っています。

```
'MVS DD ALLOC=ACTIVE'
```

メッセージ IEA611I と IEA911E のいずれかの受信時にどのしきい値も超えていなければ、アクションは実行されません。

ダンプが、低頻度 (infrequent) のしきい値の定義よりも多い頻度で行われた場合は、項目タイプ MVSDUMP で指定されたコマンド MVS DD ALLOC=ACTIVATE が出力されます。これにより、自動ダンプ・データ・セ

ット割り振りが使用可能になります。ダンプが事前に割り振られた SYS1.DUMP ダンプ・データ・セットに書き込まれた場合、さらにそのデータ・セットは、項目タイプ MVSDUMPTAKEN で指定されたコマンド MVS DD CLEAR, DSN=&EHKVAR1 を使用してクリアされます。変数 &EHKVAR1 は SYS1.DUMP ダンプ・データの数字のサフィックスで置換されます。

受信ダンプ・データ・セット・メッセージが低頻度 (frequent) レベルを超えている場合には、同じ処理が実行されます。

クリティカル (critical) のしきい値を超えるとすぐに、自動化ルーチンは事前に割り振られた SYS1.DUMP ダンプ・データ・セットのクリアを停止します。

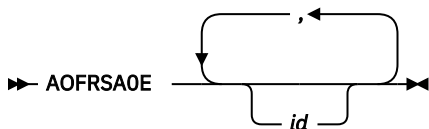
ダンプ・データ・セットの自動処理によってコマンドが出された後、自動化ルーチン AOFRSA0C は低頻度 (infrequent) のしきい値が再び満たされているかどうかを 15 分ごとに検査します。この状態に達するとすぐに、自動ダンプ・データ・セット割り振りは、項目タイプ MVSDUMPRESET の定義に従って、コマンド MVS DD ALLOC=ACTIVE によって再び使用可能になります。

## AOFRSA0E

### 目的

自動化ルーチン AOFRSA0E は、WTOR が応答または取り消しされるときに、WTOR を SA z/OS 表示機能から削除します。

### 構文



### パラメーター

#### *id*

取り消されたメッセージの応答 ID。

### 制約事項

AOFRSA0E の処理が実行されるのは、メッセージ IEE400I または IEE600I によってそれが NetView 自動化テーブルから呼び出された場合、あるいはこれらのメッセージのいずれかがパラメーターによって渡された場合だけです。

### 使用法

自動化ルーチン AOFRSA0E は、以下のメッセージに応答するためのものです。

```
IEE400I THESE MESSAGES CANCELED- id,id,id
IEE600I REPLY TO id IS; text
```

メッセージ IEE400I は、タスクの実行が終了したか、あるいは明確にメッセージの取り消しを要求したので、システムがメッセージを取り消したことを示しています。メッセージ IEE600I は、システムがこのメッセージに対する応答を受け入れたということを、そのメッセージを受信したすべてのコンソールに通知します。

また、AOFRSA0E はメッセージの ID を抽出して、渡されたパラメーターから削除することもできます。

## 例

次の例は、NetView 自動化テーブルから AOFRSA0E を出す方法を示しています。

```
IF MSGID = 'IEE400I' | MSGID = 'IEE600I'
THEN
EXEC(CMD('AOFRSA0E ')ROUTE(ONE %AOFOPWTOURS%));
```

## AOFRSA0G

### 目的

AOFRSA0G 自動化ルーチンは、メッセージに応答し、バッファ不足の自動処理のために構成ファイルからコマンドを実行して処置メッセージ保存機能 (AMRF) のバッファ不足を報告するために使用できます。

バッファ不足メッセージを受信した場合、実行するコマンドは、選択オプション PASS1 で MVSESA/AMRFSHORT 項目/タイプのペアから取得され、パス・カウントを増加させて 1 分間隔で再度実行されます。

バッファ・フル・メッセージの場合は、実行するコマンドは MVSESA/AMRFFULL 項目/タイプのペアから取得されます。バッファ不足の解消が報告された場合は、項目/タイプのペア MVSESA/AMRFCLEAR に定義されたコマンドが選択されます。

AOFRSA0G は、NetView 自動化テーブルから呼び出す必要があります。

### 構文

▶▶ AOFRSA0G ◀◀

### 制約事項

- AOFRSA0G のアクションが実行されるのは、AMRF のリカバリー自動化フラグがオンになっている場合だけです。
- AOFRSA0G のシステム・メッセージの処理が実行されるのは、メッセージ IEA359E、IEA360A または IEA361I によって NetView 自動化テーブルから呼び出された場合だけです。

### 使用法

自動化ルーチン AOFRSA0G は、次のメッセージに応答するためのものです。

```
IEA359E BUFFER SHORTAGE FOR RETAINED ACTION MESSAGES - 80% FULL
IEA360A SEVERE BUFFER SHORTAGE FOR RETAINED ACTION MESSAGES - 100% FULL
IEA361I BUFFER SHORTAGE RELIEVED FOR RETAINED ACTION MESSAGES
```

IEA359E と IEA360A は、即時アクション・メッセージ、非クリティカルおよびクリティカルな最終アクション・メッセージ、および WTOR メッセージ用のバッファ領域の不足を報告します。IEA361I は、保存されたアクション・メッセージの数が減少し、バッファが現在 75% フルを下回っていることを示しています。

AOFRSA0G がメッセージ IEA359E の受信時にトリガーされた場合、実行するコマンドは、選択オプション PASS1 から始まり、増加した選択オプションで継続し、メッセージ IEA361I がバッファ不足の解消を報告するまで 1 分間隔で項目/タイプのペア MVSESA/AMRFSHORT から取得されます。最大に達すると、定義されたコマンド処理に使用された選択オプションは、PASS1 の選択オプションから再開されます。

AOFRSA0G がメッセージ IEA360A の受信時にトリガーされた場合は、項目/タイプのペア MVSESA/AMRFFULL からすべてのコマンドが実行されます。

AOFRSA0G がメッセージ IEA361I の受信時にトリガーされた場合は、項目/タイプのペア MVSESA/AMRFCLEAR からすべてのコマンドが実行されます。

## 例

次の例は、AMRF 不足の処理のサンプル・シナリオを示しています。

NetView 自動化テーブルの項目は、メッセージ IEA359E、IEA360E、または IEA361I の受信時に AOFRSA0G を実行するために使用されます。

```
IF MSGID = 'IEA359E'
THEN
EXEC(CMD('AOFRSA0G')ROUTE(ONE %AOFOPRECOPER%));
IF MSGID = 'IEA360A'
THEN
EXEC(CMD('AOFRSA0G')ROUTE(ONE %AOFOPRECOPER%));
IF MSGID = 'IEA361I'
THEN
EXEC(CMD('AOFRSA0G')ROUTE(ONE %AOFOPRECOPER%));
```

メッセージ IEA359E と IEA361I に応答する方法を指定するために、次のコマンド定義が項目/タイプのペア MVSESA/AMRFFULL と MVSESA/AMRFCLEAR の下で自動化ポリシーに基づいて作成されます。

```
Command = ACF ENTRY=MVSESA,TYPE=AMRF*,REQ=DISP
SYSTEM = AOC1 AUTOMATION CONFIGURATION DISPLAY - ENTRY= MVSESA
-----
AUTOMATION CONFIGURATION DISPLAY - ENTRY= MVSESA
TYPE IS AMRFCLEAR
CMD = (,, 'MVS CONTROL M,AMRF=Y')
TYPE IS AMRFFULL
CMD = (,, 'MVS CONTROL M,AMRF=N')
END OF MULTI-LINE MESSAGE GROUP
```

#### 図 40. MVSESA AMRF コマンド定義

例えば、次のメッセージ

```
IEA360A SEVERE BUFFER SHORTAGE FOR RETAINED ACTION MESSAGES - 100% FULL
```

を受信すると、AOFRSA0G が NetView 自動化テーブルに示されたステートメントによって実行され、これにより AMRF バッファをクリアするためにコマンド CONTROL M,AMRF=N が実行されます。

AMRF バッファ不足が解消すると、受信メッセージ

```
IEA361I BUFFER SHORTAGE RELIEVED FOR RETAINED ACTION MESSAGES
```

によって、AMRF を再活動化するためにコマンド CONTROL M,AMRF=Y が実行されます。

## AOFRSD07

### 目的

AOFRSD07 自動化ルーチンは、ドレーンされていないリソースに対してコマンドを実行することによって、JES2 シャットダウン時に JES2 の非休止メッセージにตอบสนองするために使用できます。

実行するコマンドは、アプリケーション JES2 の自動化ポリシー項目 JES2 DRAIN から取得されます。

さらに AOFRSD07 は、すべてのアクティブ・ジョブと開始済みタスクのリストと、まだドレーンされていないすべてのリソースのリストを出力する AOFRSD0F を呼び出します。

AOFRSD07 は、NetView 自動化テーブルから呼び出す必要があります。

### 構文

▶ AOFRSD07 ◀

### 制約事項

AOFRSD07 の処理が実行されるのは、以下の場合だけです。

- JES2 メッセージ HASP607 によって NetView 自動化テーブルから呼び出された
- JES2 の終了自動化フラグがオンになっている

- JES2 のシャットダウンが進行中である

## 使用法

自動化ルーチン AOFRSD07 は、次のメッセージに応答するためのものです。

```
HASP607 JES2 NOT DORMANT -- MEMBER DRAINING, RC=rc text
```

これは、システムから JES2 を取り除くために P JES2 コマンドが入力された場合に、JES2 の機能のすべてが完了したわけではないことを示しています。

ドレーンされていないすべてのリソースを判別するために、JES2 コマンド DU,STA に対する応答が処理されます。状況 DRAINING の各リソースに対して、ドレーンを強制するため、このリソース・タイプの自動化ポリシー項目 JES2 DRAIN から対応するコマンドが出されます。ドレーンを強制するために自動化ポリシー項目からコマンドが出される前に、状況 ACTIVE のリソースが JES2 コマンド P resource によって最初に停止されます。JES2 コマンド P resource によってのみ、状況 INACTIVE のリソースが停止されます。

まだドレーンされていないリソース上で自動化によってアクションが実行できない場合は、JES2 は状況 STUCK に設定され、オペレーターのアクションが必要であることを通知するメッセージが出されます。これらの状況は、リソースをドレーンするためのコマンドが JES2 の自動化ポリシー項目 JES2 DRAINED 内に指定されていない場合や、まだドレーンされていないリソースの状況が不明な場合に起こります。

## AOFRSD09

### 目的

自動化ルーチン AOFRSD09 は JES2 スプール・リカバリーに対して使用されます。これは、JES2 のスプール使用状況をモニターするため、およびポリシー項目 JES2 SPOOLSHORT または JES2 SPOOLFULL のリカバリー・コマンドを連続的に実行するために再試行間隔ごとにタイマーによって AOFRSD01 が呼び出します。

この目的のために、AOFRSD09 は以下のステップを行います。

- AOFRSD09 は JES2 コマンド D SPOOL を出して現行スプール使用量を把握する。
- AOFRSD09 は、リカバリー処理のターゲットを実際の TG の警告しきい値と構成ファイルからのバッファ値を基にして再評価する。
- リカバリー・ターゲットがまだ達成されず、現行の JES2 サブシステムがスプールのリカバリーを行う場合は、AOFRSD09 はパス・カウントを増加させ、構成ファイルから適切なコマンドを実行します。すべての JES2 サブシステムがスプール不足メッセージのコピーを受信する共用 JES2 環境で、AOFRSD09 は、スプール・リカバリーを行う適切な JES2 サブシステムを判別します。これを行うために、AOFRSD09 は、構成ファイルに定義されている cpuid のリストを、JES2 コマンド D MEMBER,STATUS=ACTIVE に対する応答と比較します。このリストの最初のアクティブ cpuid が、スプール・リカバリーを行う適切な JES2 サブシステムであるとみなされます。
- スプール不足の問題が既に解消されている場合は、AOFRSD09 はリカバリー処理を停止し、リセット間隔後にリカバリー・コマンドのパス・カウントがリセットされるようにタイマーを設定します。

バッファ値、リセット間隔、および cpuid のような、JES2 リカバリー処理用のリカバリー・コマンドおよび構成パラメーターは、スプール不足のリカバリー処理に対する自動化ポリシー項目 JES2 SPOOLSHORT と、スプール・フル・リカバリー処理に対する JES2 SPOOLFULL によって定義します。

JES2 SPOOLSHORT および JES2 SPOOLFULL 自動化ポリシー項目の詳細については、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。

### 構文

```
▶▶ AOFRSD09 — subsystem — recovery type ◀◀
```



## パラメーター

### subsystem

JES2 のサブシステム名。このパラメーターは必須です。

### recovery type

このパラメーターは、JES2 スプール不足と JES2 スプール・フルの状態を区別するために使用されます。このパラメーターは必須です。

### SHORT

JES2 スプール不足の状態からの自動リカバリーが処理されます。

### FULL

JES2 スプール・フルの状態からの自動リカバリーが処理されます。

## 制約事項

- AOFRSD09 のリカバリー・コマンドの処理が実行されるのは、JES2 のリカバリー自動化フラグがオンになっている場合だけです。そうでない場合は、リカバリー処理は延期され、構成ファイルからの選択リカバリー・コマンドのパス・カウントは増加しません。
- 自動化ルーチン AOFRSD09 は、JESOPER によって処理する必要があります。別のタスクから呼び出されると、JESOPER に経路指定し直されます。
- AOFRSD09 の処理が実行されるのは、指定されたタイプのスプール・リカバリー処理が自動化ルーチン AOFRSD01 によって開始された場合だけです。
- SPOOLFULL リカバリー状態の間は、SPOOLSHORT リカバリーの処理は延期されます。

## 使用法

実行するコマンドは、ポリシー項目 JES2 SPOOLSHORT または JES2 SPOOLFULL のコマンド項目から選択されます。パス・カウントは選択オプションとして使用され、自動化ルーチン AOFRSD09 の連続処理ごとに増加します。リカバリー処理の初期化時に、パス・カウントは自動化ルーチン AOFRSD01 によって値 PASS1 に設定されます。

スプール不足の状態が解決される前に、受け渡し処理が定義済みリカバリー・コマンドを使い尽くした場合は、AOFRSD09 はリカバリー・シーケンスを再実行します。この動作は、System Automation の始動時に適切な拡張自動化オプションを設定することによって変更できます。AOFSPPOOLSHORTCMD 変数 (SPOOLSHORT 状態の場合) と AOFSPPOOLFULLCMD 変数 (SPOOLFULL 状態の場合) は、すべてのコマンドが実行された時にリカバリーの試行を停止すること、およびスプールの状態を解決するには手動介入が必要であるということをオペレーターに通知するためのメッセージ AOF294I を出すことを自動化ルーチン AOFRSD09 に指示するために使用できます。拡張自動化オプションについての詳細は、[204 ページの『読み取り/書き込み変数』](#)を参照してください。

## グローバル変数

リカバリーを処理する構成ファイルの SPOOLFULL または SPOOLSHORT 処理パネルでのコマンドの定義時に、変数 &EHKVAR1 と &EHKVAR2 を使用して変数の内容で置換できます。変数 &EHKVAR1 は現行スプール使用状況で置換され、&EHKVAR2 にはリカバリー・ターゲットが入ります。

## AOFRSD0F

### 目的

自動化ルーチン AOFRSD0F は、JES2 のシャットダウン前のドレーン処理のために AOFRSD07 によって使用されます。シャットダウン遅延間隔ごとに、AOFRSD0F はまだドレーンされていない JES2 リソースをすべて表示します。この目的のために、タスクを実行するための JES2 コマンド DA,S への応答、ジョブを実行するための JES2 コマンド DA,J と、開始済みデバイスまたはまだドレーンされていない回線に対する JES2 コマンド DU,STA への応答をスキャンし、結果をメッセージに表示します。

## 構文

▶▶ AOFRSD0F — *subsystem* ▶▶

## パラメーター

### *subsystem*

JES2 のサブシステム名。

## 制約事項

AOFRSD0F での処理が実行されるのは、以下の条件が満たされる場合のみです。

- サブシステムが JES2 タイプになっている
- JES2 のシャットダウンが進行中である
- 終了自動化フラグがオンになっている

## 使用法

この自動化ルーチンは、SHUTDOWN 処理の一部として実行されます。

## 例

この例は、JES2 シャットダウン前の JES2 ドレイン処理のサンプル・シナリオを示しています。

次のステートメントは、AOFRSD07 が JES2 メッセージによって NetView 自動化テーブルからどのように実行されるかを示しています。

```
$HASP607: IF MSGID(2) = 'HASP607'
THEN
EXEC(CMD('AOFRSD07')ROUTE(ONE %AOFOPJESOPER%));
```

自動化ポリシー項目に、JES2 DRAIN の次のドレイン処理の仕様があるものと想定します。

```
COMMANDS  HELP
-----
JES2 DRAIN Specification                               Line 00000001
Command ==> _____ Scroll ==> PAGE

Entry Type : Application           PolicyDB Name  : DATABASE_NAME
Entry Name  : JES2                 Enterprise Name: YOUR_ENTERPRISE

Enter YES or NO for initial drain to bring down JES2 facilities:

Drain lines . . . . . YES          (YES NO)
Drain VTAM interface . . . . . YES  (YES NO)
Drain spool offloaders . . . . . NO  (YES NO)
Drain printers . . . . . YES        (YES NO)
Drain readers . . . . . YES         (YES NO)
Drain punches . . . . . YES         (YES NO)

Enter NO or a command for force drain if normal drain fails:
Force drain lines . . . . . $E _____ (NO, Command)
Force drain VTAM interface . . . $E _____ (NO, Command)
Force drain spool offloaders . . NO _____ (NO, Command)
Force drain printers . . . . . $I _____ (NO, Command)
Force drain readers . . . . . $C _____ (NO, Command)
Force drain punches . . . . . $E _____ (NO, Command)
```

図 41. JES2 DRAIN の詳細パネル

JES2 リソースのドレインを強制するコマンドのリストは構成ファイルの JES2/FORCEDRAIN 項目/タイプのペアに渡され、DISPACF コマンドで表示できます。

```

Command = ACF ENTRY=JES2,TYPE=FORCEDRAIN,REQ=DISP
SYSTEM = KEY3      AUTOMATION CONFIGURATION DISPLAY - ENTRY= JES2
-----
AUTOMATION CONFIGURATION DISPLAY - ENTRY= JES2
TYPE IS FORCEDRAIN
LIN           = "$E"
LOG           = "$E"
OFF           = "NO"
PRT           = "$I"
RDR           = "$C"
PUN           = "$E"
END OF MULTI-LINE MESSAGE GROUP

```

#### 図 42. DISPACF パネル

JES2 のシャットダウン時にメッセージ \$HASP607 を受信し、そのメッセージが JES2 の機能のすべてが完了したわけではないこと、およびコマンド \$DU,STATUS に対する JES2 の応答が次のとおりであることを示しているとします。

```

$HASP636 13.53.22 $DU,STA
LINE1     UNIT=0FF3,STATUS=ACTIVE/BOEVM9,DISCON=NO

```

自動化ルーチン AOFRSD07 は最初に JES2 コマンド \$PLINE1 を出して回線を停止してから、項目/タイプのペア JES2/FORCEDRAIN に対するポリシーの指定に従って、JES2 コマンド \$E を実行します。

次に、ドレーンされていないすべての JES2 リソースをリストするために、シャットダウン遅延間隔ごとに自動化ルーチン AOFRSD0F が実行されます。

## AOFRSD0G

### 目的

AOFRSD0G 自動化ルーチンは、JES2 シャットダウン前に JES2 リソースをドレーンするために使用できます。AOFRSD0G は、コマンドを実行して、アプリケーション JES2 の自動化ポリシー項目 JES2 DRAIN にリストされて使用可能にされているリソースに応じて、イニシエーター、オフローダー・タスク、回線、印刷装置、パンチ、およびリーダーをドレーンします。

AOFRSD0G は DRAINJES コマンドによって使用されます。

### 構文

▶▶ AOFRSD0G — *subsystem* ▶▶

### パラメーター

#### *subsystem*

JES2 のサブシステム名。

### 制約事項

AOFRSD0G の処理が実行されるのは、サブシステムが JES2 タイプになっている場合だけです。

### 使用法

アプリケーション JES2 の自動化ポリシー項目 JES2 DRAIN の初期ドレーンに対して使用可能になっているすべてのリソースに対して、JES2 コマンド P が実行されます。

### 例

初期ドレーンの JES2 DRAIN で使用可能になっているすべてのリソースを停止するには AOFRSD0G JES2 を呼び出します。

これらのリソースは、コマンド DISPACF JES2 INITDRAIN でリストできます。

```

Command = ACF ENTRY=JES2,TYPE=INITDRAIN,REQ=DISP
SYSTEM = AOC1      AUTOMATION CONFIGURATION DISPLAY - ENTRY= JES2
-----
AUTOMATION CONFIGURATION DISPLAY - ENTRY= JES2
TYPE IS INITDRAIN
LIN           = "YES"
LOG           = "YES"
OFF           = "NO"
PRT           = "YES"
RDR           = "YES"
PUN           = "YES"
END OF MULTI-LINE MESSAGE GROUP

```

図 43. 「DISPACF JES2 INITDRAIN」 パネル

## AOFRSD0H

### 目的

AOFRSD0H 自動化ルーチンは JES2 スプール・リカバリーに対して使用されます。これは、リセット間隔後にタイマー・コマンドによって AOFRSD09 が呼び出し、構成ファイルのリカバリー・コマンドの受け渡し処理用のパス・カウンターをクリーンアップします。

### 構文

▶▶ AOFRSD0H — *subsystem* — *recovery type* →◀

### パラメーター

#### **subsystem**

JES2 のサブシステム名。このパラメーターは必須です。

#### **recovery type**

このパラメーターは、JES2 スプール不足と JES2 スプール・フルの状態を区別するために使用されます。このパラメーターは必須です。

#### **SHORT**

スプール不足の状態からのリカバリー処理用のパス・カウンターはリセットされます。

#### **FULL**

スプール・フルの状態からのリカバリー処理用のパス・カウンターはリセットされます。

### 制約事項

- AOFRSD0H 自動化ルーチンは、JESOPER によって処理する必要があります。別のタスクから呼び出されると、JESOPER に経路指定し直されます。
- リセット間隔中の各リカバリー・アクション
- AOFRSD0H がリセット間隔後にスケジュールされるのは、対応する SHORT または FULL のタイプの新規リカバリー・アクションがこの時間中に実行されなかった場合だけです。
- スプール・フル状態からのリカバリー処理用のパス・カウンターは、スプール不足の処理がまだ進行中であっても、リセット間隔後に AOFRSD0H によってリセットされます。

### 例

次の例は、JES2 スプール・リカバリー処理のサンプル・シナリオを示しています。

NetView 自動化テーブルの次の項目は、期待するメッセージのいずれかの受信時に NetView 自動化テーブルから AOFRSD01 自動化ルーチンを実行するために使用されます。

```

IF MSGID(2) = 'HASP050' & TEXT = '.'TGS'.
THEN
EXEC(CMD('AOFRSD01')ROUTE(ONE %AOFOPJESOPER%));
IF MSGID(2) = 'HASP355'

```

```
THEN
EXEC(CMD('AOFRSD01')ROUTE(ONE %AOFOPJESOPER%));
```

SPOOLSHORT リカバリーは、[194 ページの図 44](#) に示されているように、自動化ポリシー項目 JES2 SPOOLSHORT を使用して構成されます。

```

                                COMMANDS
HELP
-----
                                JES2 SPOOLSHORT Specification                Line 00000001
Command ==> _____ Scroll ==> PAGE

Entry Type : Application          PolicyDB Name : DATABASE_NAME
Entry Name  : JES2                Enterprise Name : YOUR_ENTERPRISE

Spool Shortage Retry Time . . . . . 00:02:00 (hh:mm:ss)
Spool Shortage Buffer . . . . . 5 (0 to 50)
Spool Shortage Reset Time . . . . . 00:15:00 (hh:mm:ss)

Enter priority of systems for spool recovery:
Spool Shortage Recovery SMFID 1 . . SYS1
Spool Shortage Recovery SMFID 2 . . SYS2
Spool Shortage Recovery SMFID 3 . . SYS3
Spool Shortage Recovery SMFID 4 . . ____
Spool Shortage Recovery SMFID 5 . . ____
Spool Shortage Recovery SMFID 6 . . ____
Spool Shortage Recovery SMFID 7 . . ____
Spool Shortage Recovery SMFID 8 . . ____
Spool Shortage Recovery SMFID 9 . . ____

```

図 44. JES2 SPOOLSHORT リカバリー定義

smfids が定義されていないため、JES2 サブシステムが JES2 スプール・リカバリー処理を行います。JES2 SPOOLSHORT CMDS ポリシーを編集すると、コマンド DISPACF JES2 に対する応答パネルに示されているように定義された受け渡しリカバリー・コマンドを入力できます。

```

Command = ACF ENTRY=JES2,TYPE=*,REQ=DISP
SYSTEM = KEY3      AUTOMATION CONFIGURATION DISPLAY - ENTRY= JES2
-----
TYPE IS SPOOLSHORT
CMD      = (PASS1,, 'MVS $PQ,Q=N,A=3')
CMD      = (PASS1,, 'MVS $0Q,Q=N,A=3,CANCEL')
CMD      = (PASS1,, 'MVS $PQ,Q=V,A=3')
CMD      = (PASS1,, 'MVS $0Q,Q=V,A=3,CANCEL')
CMD      = (PASS2,, 'MVS $PQ,ALL,A=4')
CMD      = (PASS2,, 'MVS $0Q,ALL,A=4,CANCEL')
CMD      = (PASS3,, 'MVS $PQ,ALL,A=3')
CMD      = (PASS3,, 'MVS $0Q,ALL,A=3,CANCEL')
CMD      = (PASS4,, 'AORRSPLS RANGE=JOB1-5000,NAME=T*')
CMD      = (PASS4,, 'AORRSPLS RANGE=JOB5000-10000,NAME=T*')
CMD      = (PASS4,, 'AORRSPLS RANGE=JOB10000-15000,NAME=T*')
CMD      = (PASS4,, 'AORRSPLS RANGE=JOB15000-20000,NAME=T*')

```

図 45. DISPACF コマンド応答パネル

JES2 スプール不足の問題が、次のメッセージによって報告されると仮定します。

```
$HASP050 JES RESOURCE SHORTAGE OF TGS - 80% UTILIZATION REACHED
```

これは、適切な NetView 自動化テーブル項目によって AOFRSD01 自動化ルーチンを実行します。AOFRSD01 は JES2 SPOOLSHORT リカバリー処理を開始し、カスタマイズ・ダイアログで定義されているように、5 分ごとに AOFRSD09 JES2 SHORT を出すことによって受け渡し処理ルーチン呼び出すように、すべてのタイマー を設定します。SPOOLSHORT 処理については、[194 ページの図 44](#) を参照してください。

AOFRSD09 は、実際のスプール使用量を再判別し、それを定義済みの 80% の TGWARN と比較して、リカバリーのターゲットを TGWARN とバッファ値の差として計算します。この結果、75 という値が得られます。実際のスプール使用量がこの値を超過すると、SPOOLSHORT リカバリー・タイプに対する構成ファ

イルの PASS1 選択オプションがあるリカバリー・コマンドがすべて実行されます。5 分の再試行間隔後、AOFRSD09 はタイマーによって再度実行されます。

ここで、AOFRSD09 が JES2 スプール不足の問題は解消されたと判断した場合は、リカバリー処理を停止し、15 分のリセット間隔後 AOFRSD0H JES2 SHORT を出すようにタイマーを設定します。

期待する JES2 メッセージのいずれもリセット間隔の終わりまでに受信しない場合、AOFRSD0H 自動化ルーチンはパス・カウントを 1 にリセットします。これにより、次の SPOOLSHORT リカバリー処理が、PASS1 選択オプションから再び始まっているリカバリー・コマンドを実行するようになります。

## HASP099

### 制約事項

JES2 メッセージ HASP099 のシャットダウン処理は、次の場合にのみ実行されます。

- JES2 のシャットダウン自動化がオンになっている
- JES2 のシャットダウンが進行中である

### 使用法

ISSUEACT コマンドは、次のメッセージに応答します。

```
HASP099 ALL AVAILABLE FUNCTIONS COMPLETE
```

これは、すべての JES2 ジョブ・プロセッサが休止状態になり、どの JES2 RJE 回線もアクティブではないことを示しています。

## INGRMJSP

### 目的

INGRMJSP 自動化ルーチンは、JES2 スプール・ファイル使用量をモニターする場合に使用します。このルーチンは、現在のスプール使用量および警告レベルを取得するために、スプール使用量を照会します。必要であれば、このルーチンは JES2 スプール・リカバリー処理に対して INGRJSP 自動化ルーチン呼び出します。

INGRMJSP コマンドは、呼び出されるたびに、状況表示機能 (SDF) の SPOOL 項目の更新も行います。

### 構文

```
▶▶ INGRMJSP ◀◀
```

### 制約事項

INGRMJSP によるモニタリングは、カスタマイズ・ダイアログで適切なモニター・リソース用のモニター・コマンドとして INGRMJSP が定義されている場合にのみ行われます。

### 使用法

INGRMJSP モニター・ルーチンは、D SPOOLDEF,TGSPACE コマンドを発行してスプール使用量を照会し、JES2 システム・プログラマーによって次のようにセットアップされている現在のスプール使用量および警告レベルを取得します。

- スプール・ファイルがフルの場合、INGRMJSP はヘルス状況を「CRITICAL」に設定し、INGRCJSP を呼び出します。
- スプール使用量が警告レベルを超えている場合、INGRMJSP はヘルス状況を「WARNING」に設定し、INGRCJSP を呼び出します。

スプール・フルの割合および警告レベルに従って、次のいずれかの戻りコードが設定されます。

戻りコード	意味
1	重大エラーが起きました。 <ul style="list-style-type: none"> <li>• モニターにジョブ名がありません</li> <li>• モニター対象オブジェクトは SPOOL でないか、または JES2 と関連付けられていません</li> <li>• 指定されたジョブ名が JES2 リソースを参照しません</li> <li>• JES2 のコマンド接頭部が見つかりませんでした</li> </ul>
2	次のモニター・コマンドが失敗しました。 <ul style="list-style-type: none"> <li>• D SPOOLDEF コマンドが失敗しました</li> </ul>
3	OK: スプール使用量は警告レベルより下です。
4	警告: スプール使用量は警告レベルより上です。
6	クリティカル: スプール・ファイルがフルです。

### 例

カスタマイズ・ダイアログでスプール使用量モニターを作成するには、以下の項目を定義する必要があります。

1. INGRMJSP をモニター・コマンドとするモニター・リソース (MTR)。例えば、「JES2 スプール・モニター」と簡略説明を付けた JES2SPOOL というモニター・リソースを作成する場合、MONITOR INFO ポリシー項目の中に以下の情報を指定します。

MONITOR INFO 項目	項目
モニター対象オブジェクト	SPOOL
モニター対象ジョブ名	JES2
活動化コマンド	
非活動化コマンド	
モニター・コマンド	INGRMJSP
モニター間隔	00:15
取り込みメッセージ限度	20
使用可能を要求 (Desired Available)	
通知リスト	SDF
所有者	
情報リンク	

2. RELATIONSHIPS ポリシー項目を使用して、以下の JES2 アプリケーションとの関係を定義します。

関係タイプ	サポートするリソース	条件
HasParent	JES2/APL/=	
ForceDown	JES2/APL/=	WhenObservedDown

モニターは、対応する JES2 リソースとの HasParent 関係を持ちます。これは、JES2 がアクティブの場合にのみ、スプール使用量をモニターする意味があるからです。



3. 以下のリカバリー・アクションを HEALTHSTATE ポリシー項目の中に定義します。

状態	コマンド
WARNING	INGRCJSP
CRITICAL	INGRCJSP

## INGRCJSP (AOFRSD01)

### 目的

INGRCJSP 自動化ルーチンは JES2 スプール・リカバリー処理に対して使用できます。このルーチンは、JES2 スプール不足に対してリカバリー処理を開始することによって JES2 スプール不足メッセージに対応します。また、スプールの超過使用の問題をダウングレードするために JES2 スプール・フルに対するリカバリー処理を開始することによって JES2 スプール・フル・メッセージに対応します。

INGRCJSP ルーチンは以下の処理を行います。

- 実際の値と履歴の値を基にしてスプール使用量の線形の 1 次予測を行う。
- スプール状況を状況表示機能 (SDF) に通知する。
- リカバリー処理のターゲットを実際のトラック・グループの警告しきい値と構成ファイルからのバッファ値の間の差として判別する。リカバリー処理がこのターゲットを実行すると、スプール不足の状態が軽減されたとみなされます。
- 受け渡し処理を開始して、JES2 SPOOLSHORT または JES2 SPOOLFULL ポリシー項目によって定義された構成ファイルのリカバリー・コマンドを実行する。受け渡し処理そのものは、再試行間隔ごとに出される AOFRSD09 自動化ルーチンによって実行されます。再試行間隔は構成ファイルから得られます。

バッファ値および再試行間隔のような、JES2 リカバリー処理用のリカバリー・コマンドおよび構成パラメーターは、スプール不足のリカバリー処理に対する自動化ポリシー項目 JES2 SPOOLSHORT と、スプール・フル・リカバリー処理に対する JES2 SPOOLFULL によって定義します。

JES2 SPOOLSHORT および JES2 SPOOLFULL 自動化ポリシー項目の詳細については、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。

INGRCJSP NetView 自動化テーブルから呼び出す必要があります。

### 構文

▶▶ INGRJSP ◀◀

### 制約事項

- INGRJSP の処理が実行されるのは、それが JES2 メッセージ HASP050 または HASP355 によって NetView 自動化テーブルから呼び出された場合だけです。
- メッセージ HASP355 が処理されるのは、トラック・グループ (TG) の不足が報告された場合だけです。

### 使用法

INGRCJSP 自動化ルーチンは、次のメッセージに応答するためのものです。

```
HASP050 JES2 RESOURCE SHORTAGE OF TGs - nnn% UTILIZATION REACHED
HASP355 SPOOL VOLUMES ARE FULL
```

HASP050 は、JES2 にはトラック・グループの不足があり、スプールの現在の使用状況がこの JES における現行 TGNWARN 値を超えていることを示しています。TGNWARN は JES 初期化メンバーの SPOOLDEF ステートメントで定義されていて、動的に変更できます。

HASP355 は、使用可能なスペースがすべて JES2 機能に割り振られているか、あるいはスプール・ボリュームが使用できないので、JES2 直接アクセス・スプール・スペースに対する要求を処理できないことを示

しています。したがって、この場合、リカバリー・ターゲットは 100% のスプール使用状況の数値が基になっています。

SPOOLSHORT リカバリーが SPOOLFULL 状態になる前に開始されるように、TGWARN を JES 初期化メンバー中の SPOOLDEF ステートメントにコーディングする必要があります。これを実行しないと、リカバリー処理は予測不能になることがあります。

SPOOLFULL 状態の後にリセットすると、この問題は SPOOLSHORT 状態にダウングレードします。SA z/OS では、以前に実行した SPOOLSHORT リカバリーが活動化してこの問題を OK にダウングレードします。事前の SPOOLSHORT リカバリーが行われないと、正常に SPOOLFULL リカバリーが行われた後も、スプール状況は SPOOLSHORT のままになります。

JES2 メッセージの NetView 自動化テーブル項目では、JES2 メッセージのメッセージ ID の前にある 1 文字のプレフィックスが重要になります。これにより、送信側の JES を識別することができます。

スプール状況は汎用 SPOOL 下の SDF に、固有名としてのサブシステムの名前と共に通知されます。これらを SDF パネルに表示するには、xxxx.SPOOL (エレメント 1 から n) の状況フィールドが必要です (n はそのスプールを使用する異なるサブシステムの数)。

## INGRTAPE

### 目的

INGRTAPE は SDF のテープ状況に関する詳細を保持します。SA z/OS は未処理のテープ・マウント要求を検出すると、関連メッセージを SDF に送ります。警告の間隔が満了する前に要求が満たされない場合は、状況が警告に変更されます。

アラートの遅延後にもテープ・マウント要求が満たされない場合は、状況がアラートに変更されます。

テープ・マウント要求は、関連したテープがマウントされるか、要求ジョブが取り消されると SDF から動的に削除されます。

INGRTAPE 自動化ルーチンは、SDF 内で保留中のテープ・マウント要求を視覚化するために使用されます。この動作は、「Tape Attendance」ポリシー・エントリーの定義に基づきます。テープ・アテンダンス (在席か不在) の活動化とカスタマイズについては、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。

### 構文

▶▶ INGRTAPE ◀◀

### 使用法

自動化ルーチン INGRTAPE は、次のメッセージに応答することを目的としています: IEC501E、IEC501A、IEC502E、IEC503I、IEC507D、IEC509A、IEC510D、IEC512I、IEC513D、IEC514D、IEC701D、IEC702I、IEC703I、IEC704A、IEC706I、IEC707I、IEC708I、IEC708D、IEC709I、IEC710I、IEC711I、IEC712I、IEC713I、IEC714I、IEC715I、IEF233A、IEF233D、IEF234E、IEF455D、IAT5210、TMS001、TMS002、TMS0012

### 制約事項

テープ・マウントのモニターは、カスタマイズ・ダイアログで活動化された場合にのみ使用可能になります。

INGRTAPE でアクションが実行されるのは、問題のメッセージについてリカバリー自動化フラグがオンになっている場合のみです。このフラグ検査はマイナー・リソース MVSESA.messageID に対して行われます。

## INGRX711

---

### 目的

自動化ルーチン INGRX711 は、システムがディレクトリー不足を報告したときはいつでも、DSEXTENT パラメーターを大きくして 1 次または代替の LOGR CDS を再フォーマットすることができます。

### 構文

▶▶ INGRX711 ◀◀

### 使用法

自動化ルーチン INGRX711 は、メッセージ IXG261E および IXG257I に応答することを目的としています。

### 制約事項

マイナー・リソース名 LOGGER および CDS に対して自動化を許可する必要があります。

## INGRX740

---

### 目的

INGRX740 自動化ルーチンを使用し、自動化制御ファイルから定義済みのリカバリー・アクションを発行して `syslog` をリスタートする、または `syslog` をハードコピー・メディアとして割り当てることにより、一部の `syslog` 関連のシステム・メッセージに応答できます。

INGRX740 は、着信 IEE037D `syslog` 非アクティブ・メッセージをトラッキングして、そのオカレンスを MVS コンポーネント・マイナー・リソース LOG に事前定義されたしきい値と比較します。クリティカル (critical) しきい値レベルを超えない限り、前に受け取ったシステム・メッセージに関連したリカバリー・アクションが実行されます。

現在処理中の IEE037D メッセージの前にメッセージ IEE043I、IEE533E または IEE769E の 1 つを受け取っている場合は、構成ファイルの `MVSESA/msgid` 項目/タイプのペア内の IEE043I、IEE533E または IEE769E 用に定義されているコマンドが実行されます。現在処理中の IEE037D メッセージの前に、上述のいずれのメッセージも受け取っていない場合は、コマンド **MVS WRITELOG START** が実行されます。

着信 IEE041I メッセージが、`SYSLOG` データ・セットはハードコピー・ログとして使用できることを示している場合は、リカバリー・ルーチン INGRX740 はその IEE041I メッセージにも応答します。構成ファイルの `MVSESA/IEE041I` 項目/タイプのペア内に定義されたメッセージ IEE041I に応答して、コマンドが実行されます。このケースに該当するコマンドは、**MVS VARY SYSLOG,HARDCPY** であり、このコマンドは `SYSLOG` がハードコピー・ログを受け取るようにします。

INGRX740 は、NetView 自動化テーブルから呼び出す必要があります。

### 構文

▶▶ INGRX740 ◀◀

### 制約事項および制限事項

ルーチン INGRX740 での処理が実行されるのは、以下の条件が満たされる場合のみです。

- LOG のリカバリー自動化フラグがオンになっている。
- ルーチンが、自動化タスク上で実行されている。
- ルーチンは、以下の期待するメッセージの 1 つによって NetView 自動化テーブルから呼び出された。
  - IEE037D
  - IEE041I

- IEE533E
- IEE769E
- IEE043I

メッセージ IEE037D に応答するアクションが INGRX740 で実行されるのは、ジョブ入力サブシステムが稼働中の場合のみです。

**使用法**

自動化ルーチン INGRX740 は、以下のメッセージに応答します。

```
IEE037D LOG NOT ACTIVE
IEE041I THE SYSTEM LOG IS NOW ACTIVE[-MAY BE VARIED AS HARDCOPY LOG]
IEE043I A SYSTEM LOG DATA SET HAS BEEN QUEUED TO SYSOUT CLASS class
IEE533E SYSTEM LOG INITIALIZATION HAS FAILED
IEE769E SYSTEM ERROR IN SYSTEM LOG
```

**例**

この例は、システム・ログ障害のリカバリーのサンプル・シナリオを示しています。

NetView 自動化テーブルの次の項目は、SA z/OS によって提供され、着信メッセージ IEE043I および IEE037D に応答して INGRX740 を実行します。

```
IF MSGID = 'IEE037D' THEN
EXEC(CMD('INGRX740')ROUTE(ONE %AOFOPRECOPER%));
IF MSGID = 'IEE043I' THEN
EXEC(CMD('INGRX740')ROUTE(ONE %AOFOPRECOPER%));
```

MVS コンポーネント・マイナー・リソース LOG の自動化ポリシーに、以下のしきい値レベルが定義されているとします。

COMMANDS	HELP
-----	
Thresholds Definition	
Command ==> _____	
Entry Type : MVS Component	PolicyDB Name : DATABASE_NAME
Entry Name : MVS_COMPONENTS	Enterprise Name : YOUR_ENTERPRISE
Resource : MVSESA.LOG	
Critical Number . . . . 3	(1 to 50)
Critical Interval . . . 00:05	(hh:mm or hhmm, 00:01 to 24:00)
Frequent Number . . . . 3	(1 to 50)
Frequent Interval . . . 00:30	(hh:mm or hhmm, 00:01 to 24:00)
Infrequent Number . . . 3	(1 to 50)
Infrequent Interval . . 24:00	(hh:mm or hhmm, 00:01 to 24:00)

図 46. MVS コンポーネント LOG に対するしきい値定義

MVS コンポーネントの自動化ポリシー項目 MESSAGES/USER DATA 内のメッセージ IEE043I に対して、以下の図に示すようにコマンドが定義されているとします。

```

COMMANDS  HELP
-----
Command ==> _____ CMD Processing _____ Row 1 to 4 of 20
                                           SCROLL==> PAGE
Entry Name : MVS_COMPONENTS      Message ID : IEE043I

Enter commands to be executed when resource issues the selected message.
or define this message as status message.

Status . . . _____ ('?' for selection list)

Pass/Selection Automated Function/'*'
Command Text
-----
MVS WRITELOG START _____
-----

```

図 47. 項目/タイプのペア MVSESA/LOG の MESSAGES/USER DATA ポリシー項目

ある日、ジョブ入力サブシステムが稼働中であり、MVS コンポーネント・マイナー・リソース LOG のリカバリー自動化フラグがオフに切り替えられていないときに、次のメッセージが初めて送られてきたとします。

```

IEE043I A SYSTEM LOG DATA SET HAS BEEN QUEUED TO SYSOUT CLASS 1
IEE037D LOG NOT ACTIVE

```

メッセージ IEE037D の前に IEE043I を受け取っており、メッセージ IEE037D 用に定義されたクリティカル (critical) しきい値を超えていないので、メッセージ IEE037D に応答してメッセージ IEE043I 用に定義されたコマンドが実行されます。



## 付録 A グローバル変数

ユーザーが作成するグローバル変数の名前はどれも、SA z/OS の外部グローバル変数の名前または内部グローバル変数の名前と同じでないことを確認する必要があります。ユーザー独自のグローバル変数を作成する前に、以下の表を確認してください。

### 読み取り専用変数

これらの変数の値は、SA z/OS で内部的に使用されます。これらは読み取り可能ですが、ユーザー REXX ルーチンで変更してはなりません。

変数名	説明	参照
AOF.clist.ODEBUG	Y またはブランクのいずれかが入ります。 Y の場合、SA z/OS 自動化プロシージャがサポートするデバッグの中間レベルはオンになります。	
AOF.clist.OTRACE	自動化プロシージャ <i>clist</i> によって使用される REXX トレース設定値が入ります。	
AOF_PRODVL	SA z/OS のリリース・レベルが入ります。 値は次のとおりです。 <b>SA z/OS 4.2</b> SA z/OS, V4R2M0 <b>SA z/OS 4.1</b> SA z/OS, V4R1M0 <b>SA z/OS 3.5</b> SA z/OS, V3R5M0	
AOFAOCCLONEx	x は、存在しない (AOFAOCCLONE) か、1 から 9 または A から Z のいずれかの値です。 AOFAOCCLONEx グローバル変数には、このシステムの &AOCCLONEx 変数に指定された値が入ります。	「IBM Z System Automation 自動化ポリシーの定義」のシステム・ポリシー・オブジェクトの説明を参照してください。
AOFBFP	バックアップ・フォーカル・ポイントが入ります。	
AOFCFP	現行フォーカル・ポイントのドメイン ID が入ります。	
AOFCOMPL	初期化が完了すると、YES が入ります。	
AOFDEBUG	グローバルに使用する REXX トレース設定値が入ります。	
AOFINITIALSTARTTYP	SA z/OS が IPL 後初めて開始されたか、NetView の再生後初めて開始されたかによって、「IPL」または「RECYCLE」の値が入ります。	
AOFJESPREFX	1 次スケジューリング・サブシステムのコマンド接頭部。	
AOFPPF	1 次フォーカル・ポイントが入ります。	



表 25. 外部定義の共通グローバル変数 (続き)		
変数名	説明	参照
AOFUFP	1 次フォーカル・ポイントとバックアップ・フォーカル・ポイントが並行して更新される場合は、値「B」が入ります。そうでない場合、この変数は存在しません。	
AOFSUBSYS	1 次スケジューリング・サブシステムのサブシステム名。	
AOFSYSNAME	システムの名前が入ります。	「IBM Z System Automation プログラマーズ・リファレンス」の『AOCUPDT』を参照してください。
AOFSYSTEM	システム・タイプが入ります。常に MVSESA に設定されます。	
AOFSYSTEMSHUTDOWN	GDPS または INGREQ ALL を介してシステム・シャットダウンが呼び出された場合は YES が入ります。  注: NetView/SA が終了するか、システムが XCF グループを離れるまで、この値は YES のままです。	
AOFSYSPLEX	物理シスプレックスの名前が入ります。	INGAMS パネルの「Sysplex」列を参照してください。
AOFSYSPLEXGROUP	カスタマイズ・ダイアログで定義された、グループ・タイプが SYSPLEX の GRP ポリシー項目の名前が入ります。	カスタマイズ・ダイアログの「GROUP INFO」パネル。INGAMS パネルの「SAplex」列を参照してください。

## 読み取り/書き込み変数

204 ページの表 26 には、ユーザーが定義できる共通グローバル変数が記載されています。ユーザー始動出口でそれらを設定して、SA z/OS の動作方法を変更できます。これらの変数は、SA z/OS システムに対して一度だけ設定してください。拡張自動化オプション (AAO) は、NetView スタイルシートでグローバル変数の設定値を変更することによって、使用可能または使用不可にすることができます。例:

```
*****
* System Automation AAO CGlobals
*****
COMMON.AOFCNMASK = 290C0D0E0F101518
COMMON.INGREQ_ORIGINATOR = 1
COMMON.AOFRESTARTALWAYS = 0
COMMON.AOFSMARTMAT = 0
```

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS)		
変数	値	効果
AOF_AAO_AAREQUEST_MAX_WAIT	1 から 999 秒	最大待機時間を秒数で定義します。ワークロードに MAXWAIT パラメーターが指定されていない場合に使用します。有効な範囲については、MAXWAIT パラメーターを参照してください。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_AAO_AOCUPDT_LOGGING	YES   NO	LOG パラメーターを指定せずに AOCUPDT が呼び出された場合に、トレース・メッセージを NetView ログに書き込むかどうかを指定します。デフォルト値は YES です。
	FORCE_YES   FORCE_NO	トレース・メッセージを NetView ログに書き込み、AOCUPDT 呼び出しの LOG パラメーターで指定された値を無効にするかどうかを指定します。
AOF_AAO_AOCUPDT_PRESERVE_CASE	YES	AOCUPDT をパラメーター MSG、INFO、または USER とともに使用すると、これらのパラメーター値の大/小文字は、SDF での表示用に保持されます。
	未定義またはその他の値	AOCUPDT パラメーター値はすべて大文字に変換されます。これはデフォルトの動作です。
AOF_AAO_GWPING_RETRY	1-99	ゲートウェイ・モニターは、時間がかかる可能性のある RMTCMD をリモート・ターゲットに対して試行する前に、PING コマンドを発行できます。この AAO により、PING コマンドが正常に行われなかった場合の再試行回数を指定することができます。  AAO が設定されていない場合、または指定された値が無効である場合、PING コマンドは発行されません。
AOF_AAO_HW_VALIDATION (このオプションは SA z/OS にのみ適用されます)	YES	このオプションは、ハードウェア検証の実行を可能にします。詳しくは、章『シスプレックス自動化の有効化』およびセクション『ハードウェア関連の自動化を使用可能にする』を参照してください。YES がデフォルトです。
	NO	このオプションは、ハードウェア検証の続行を禁止します。検証に依存する機能は正しく機能しない可能性があります。対象となった検証ルーチンは、戻りコード 2 で終了します。
AOF_AAO_INJECT_NOFORCE_REQ	任意の値	SA z/OS は、INGREQ REQ=STOP RESTART=YES 要求を処理する際に、ソース *RECYCLE に対して Priority Force を持つ STOP 投票を投入しません。代わりに、通常の停止要求が自動化マネージャーに渡され、リソースがダウンすると自動的に削除されます。この要求は、同じソースによって作成されたリソースに対する前の要求もすべて削除します。
AOF_AAO_IPL_COMPLETE_MSG	COND	期待される時間制限内にすべてのリソースが AVAILABLE 状況に達した場合にのみメッセージ ING313I を出し、そうでない場合は ING314I を出すことを指定します。
	ALWAYS	ING314I が出され、リソースが後で (任意の時点で) AVAILABLE になる場合に、ING315I を追加で出すことを指定します。サブスクリプションは、リソースが AVAILABLE 状況に達するまで有効のままになります。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_AAO_ISQ_DYNTGT	ユーザー定義	<p>プロセッサ操作開始時の動的ターゲット・システム名の作成方法のパターンが含まれています。</p> <p>このパターンは、以下の3つのビルディング・ブロックから構成できます。</p> <ul style="list-style-type: none"> <li>• Lx: 1 から x までの LPAR 名を使用します。</li> <li>• Cy: 1 から y までの PROCESSOR 名を使用します。</li> <li>• 特殊文字「#\$@」(許可されるのは1つのみです)。</li> </ul> <p>パターンについて詳しくは、「IBM Z System Automation ユーザーズ・ガイド」の『プロセッサ操作 - 動的ターゲット・システム名のセットアップ』を参照してください。</p>
AOF_AAO_ISQ_KALIST	ip_addr   ip_addr/i	<p>SE または HMC のドット 10 進アドレスまたはホスト名を指定します。</p> <p>/ - 分離文字</p> <p>i - アイドル時間 (分)</p> <p>1 つ以上の空白で区切って、複数の定義を行うことができます。ip_addr はそれぞれ、PDB からのプロセッサ・アドレス定義に一致しなければならないことに注意してください。</p>
AOF_AAO_JLM_RECYCLE	RESET	NetView リサイクル後に前のモニター・タスクをすべて無視するよう、ジョブ・ログ・モニターに指示します。
	RESUME	NetView リサイクル後にモニターを続行するよう、ジョブ・ログ・モニターに指示します。
AOF_AAO_MSG_EHKVAR	<b>YES</b>	<p>これは、コマンドの呼び出し時に、起動されるメッセージのトークンが、パラメーター EHKVAR で指定されていない場合、変数 EHKVAR0 から EHKVAR9 および EHKVART に保管されることを指示します。</p> <p>YES がデフォルトです。</p>
	NO	これは、起動されるメッセージのトークンが、パラメーター EHKVAR で指定されていない場合、EHKVAR 変数に保管されないことを指示します。
AOF_AAO_MVSTAPEMON	>0	LATE 警告に達した後に、INGRTAPE が MVS コマンドを使用したモニターを継続する反復回数を示すように、この値を設定します。また、ゼロ以外の入力は、LATE 警告の前に MVS コマンドをすべてのテープ・マウントのモニターに使用することを示します。
	0	INGRTAPE は、DOMMED メッセージの受け取りに頼って、すべての未解決の警告を満たすこととなります。
AOF_AAO_OMVS_SHUTDOWN	NOWAIT	OMVS の完全な終了に対する待機をスキップします。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_AAO_RDS_TSO_DSN	ユーザー定義	TSO で RDS テーブル・エディターを使用するために設定する必要があります。RDS 作業データ・セットの名前を指定します。この値が変更された場合、この値を有効にするために NetView はリサイクルを必要とします。
AOF_AAO_RDS_TSO_RACFHLQ	ユーザー定義	TSO で RDS テーブル・エディターの RACF 保護を使用するために設定する必要があります。最大 26 バイト長の高位修飾子を指定します(ドットを含む場合あり)。この値が変更された場合、この値を有効にするために NetView はリサイクルを必要とします。
AOF_AAO_RETENTIONPERIOD	0 から 1440	SA z/OS に、TWS から受け取るコマンド要求をトラッキングするために使用する CGLOBALS を保持させる時間(分単位)を定義します。デフォルトは 60 分です。
AOF_AAO_SDF_MSGDEL_OPERID	taskname	ターゲット・システム上の SDF フォーカル・ポイントから収集された例外メッセージを削除するために、RMTCMD で使用されるタスクの名前。
	<OPER>	RMTCMD でターゲット上のメッセージを削除するために、フォーカル・ポイントからのオリジナル・オペレーター ID を使用します。
	ブランク	RMTCMD でターゲット上の重大メッセージを削除するためにレポーター ID タスクを使用します。これがデフォルトです。
AOF_AAO_SDFBFP_ROOT.*	ユーザー定義	バックアップ・フォーカル・ポイントの静的ルート名を定義します。詳しくは、「 <i>IBM Z System Automation</i> プログラマーズ・リファレンス」にあるコマンド SDFTREE の『制約事項および制限事項』を参照してください。  使用する場合、変数 AOF_AAO_SDFBFP_ROOT.0 が存在する必要があります。隣接グローバル数を定義します。  隣接グローバルの値には、単一のルート名、またはブランク文字で区切られたルート名のリストを指定できます。
AOF_AAO_SDFFPF_ROOT.*	ユーザー定義	1 次フォーカル・ポイントの静的ルート名を定義します。詳しくは、「 <i>IBM Z System Automation</i> プログラマーズ・リファレンス」にある SDFTREE コマンドの『制約事項および制限事項』を参照してください。  この変数を使用する場合は、AOF_AAO_SDFFPF_ROOT.0 変数が存在する必要があります。隣接グローバル数を定義します。  隣接グローバルの値には、単一のルート名、またはブランク文字で区切られたルート名のリストを指定できます。  1 次フォーカル・ポイントとバックアップ・フォーカル・ポイントが並行して更新される場合にのみ、グローバルが評価されます。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)

変数	値	効果
<p>AOF_AAO_SDFROOT_LIST AOF_AAO_SDFROOT_LISTn (n は 1 から 4)</p>	<p>ユーザー定義</p>	<p><b>(非推奨、AOF_AAO_SDFROOT.* を使用)</b></p> <p>SA z/OS で用意されているサンプル SDF パネルのルート名として使用される &amp;SDFROOT 変数の値を定義します。</p> <p>可能な値は、単一システムの名前か、または空白文字で区切られたシステム名のリストです。リストは、SDF フォーカル・ポイントで使用して、SA z/OS がそのリストの全システムに必要なパネル定義を生成できます。</p> <p>AOF_AAO_SDFROOT_LISTn 変数は、NetView 環境内の共通グローバル変数の最大長を受け入れる場合に使用します。</p> <p>システム名には 1 つ以上のメンバー名をスラッシュで区切って追加することができます。最初の名前はパネル定義を示し、2 番目の名前はツリー定義を示します (例: SYS1 SYS2/MYPNLS SYS3&amp;SLASH./MYTREE SYS4/MYPNLS/MYTREE)。これらの定義の解釈については、「<i>IBM Z System Automation プログラマーズ・リファレンス</i>」の『状況コンポーネント・パネル定義』を参照してください。</p>
<p>AOF_AAO_SDFROOT.*</p>	<p>ユーザー定義</p>	<p>SA z/OS で用意されているサンプル SDF パネルのルート名として使用される &amp;SDFROOT 変数の値を定義します。</p> <p>使用する場合、変数 AOF_AAO_SDFROOT.0 が存在する必要があり、隣接グローバル数を定義します。</p> <p>隣接グローバルの値には、単一システムの名前、または空白文字で区切られたシステム名のリストを指定できます。リストは、SDF フォーカル・ポイントで使用して、SA z/OS がそのリストの全システムに必要なパネル定義を生成できます。</p> <p>システム名には 1 つ以上のメンバー名をスラッシュで区切って追加することができます。最初の名前はパネル定義を示し、2 番目の名前はツリー定義を示します (例: SYS1 SYS2/MYPNLS SYS3&amp;SLASH./MYTREE SYS4/MYPNLS/MYTREE)。これらの定義の解釈については、「<i>IBM Z System Automation プログラマーズ・リファレンス</i>」の『状況コンポーネント・パネル定義』を参照してください。デフォルトの名前は AOFPNLS および AOFTREE です。</p> <p><b>注:</b> NetView は、2 つの連続したスラッシュを行コメントの始まりと解釈します。この理由で、上の例では、スラッシュ文字の記号が使用され、その後にスラッシュ文字自体が続いています。</p>

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_AAO_SDFTGT_COMP.*	ユーザー定義	<p>コンポーネントが属するルートツリーの全体を削除もリフレッシュもすることなく、RESYNC 要求で SDF フォーカル・ポイントに転送する、特定のコンポーネント名を定義します。詳しくは、「<i>IBM Z System Automation</i> プログラマーズ・リファレンス」にある SDFTREE コマンドの『制約事項および制限事項』を参照してください。</p> <p>変数 AOF_AAO_SDFTGT_COMP.0 は、使用した場合、隣接グローバル数を定義します。</p> <p>末尾のワイルド・キャラクターであるアスタリスク (*) が、コンポーネント名、代替コンポーネント名、およびメジャー・コンポーネント名でサポートされていますが、ルート・コンポーネント名ではサポートされていません。例えば、「ROOT.H*(IPL*)」という指定は、先頭が「H」のすべてのコンポーネント名と、先頭が「IPL」のすべての代替コンポーネント名に一致します。さらに、システム記号もサポートされています。</p>
AOF_AAO_SEC_PPIAUTH	FAIL	<p>コマンド受信側、Web アダプター、および E2E アダプターが受信する着信コマンドのサード・パーティー検査で使用されます。対応する SAF プロファイルまたはクラスが定義されていない場合、着信コマンドの実行はリジェクトされます。</p>
	未定義またはその他の値	<p>コマンド受信側、Web アダプター、および E2E アダプターが受信する着信コマンドのサード・パーティー検査で使用されます。対応する SAF プロファイルまたはクラスが定義されていない場合、着信コマンドの実行はパススルーされます。</p>
AOF_AAO_SDFCxxx.*	ユーザー定義	<p>SA z/OS で提供されるサンプル SDF ツリーの状況コンポーネント名として使用される &amp;SDFCxxx 変数の値を定義します。xxx は、共通グローバルの NetView 要件を満たす、任意の文字のシーケンスです。</p> <p>使用する場合、変数 AOF_AAO_SDFCxxx.0 が存在する必要があり、隣接グローバル数を定義します。</p> <p>隣接グローバルの値には、状況コンポーネントの名前、または空白文字で区切られた状況コンポーネント名のリストを指定できます。隣接グローバルはすべてリストに結合されます。これにより、SA z/OS は、リストにあるすべてのコンポーネント名について、必要なツリー定義を複製します。</p> <p>これらの定義の解釈については、「<i>IBM Z System Automation</i> プログラマーズ・リファレンス」の『AOCUPDT』および『AOFREE』を参照してください。</p>
AOF_AAO_SDFCsaplex.n		<p>エンタープライズ全体の System Automation サブプレックスに命名します。</p>

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_AAO_SHUTDOWN_BOOST	<b>YES</b>   未定義またはその他の値	<p>この変数は、GDPS を使用して z/OS システムがシャットダウンされたときに、シャットダウン・ブーストを活動化できるかどうかを制御します。</p> <p>指定しない場合は、値 YES が想定されます。YES に設定すると、シャットダウン・ブーストの活動化が可能になります。その他の値は、シャットダウン・ブーストの活動化が試行されないことを意味します。</p> <p><b>注:</b> このオプションは、z/OS システムが通常は System Recovery Boost を活用できるが、インストール済み環境で特定の理由でオプトアウトしたいという状況でのみ使用するよう意図されています。</p>
AOF_AAO_SHUTDOWN_STOPAPPL	ユーザー定義	<p>シャットダウンの完了を GDPS に報告するために、システム・シャットダウン中に SA z/OS がモニターするリソースの名前 (name/type/system) を指定します。GDPS ソリューションの停止アプリケーションを定義する方法については、GDPS の資料を参照してください。</p>
AOF_AAO_SHUTSYS_OLD	YES	<p>GDPS タワーがアクティブな場合に、SA z/OS が INGREQ ALL REQ=STOP コマンドを GDPS STOPAPPL リソースにリダイレクトしないように指定します。</p>
AOF_AAO_TRANRERUN	YES	<p>そのジョブの自動化ポリシーで別途指定されていない場合、一時ジョブは、特定の z/OS のライフ・サイクル内で再実行可能であることを示します。</p>
	<b>NO</b>	<p>そのジョブの自動化ポリシーで別途指定されていない場合、一時ジョブは、特定の z/OS のライフ・サイクル内で 1 回のみ実行されることを示します。NO がデフォルト値です。</p>
AOF_AAO_TWS_CHK_CONDDPEP	YES	<p>操作が正常に行われた場合に条件付き依存関係処理のための完了検査ルーチンから返される戻りコードを保存できる、新しい動作を活用するには、この AAO を YES に設定します。</p>
AOF_AAO_TWS_CHK_OUTPUT_NETLOG	YES	<p>完了検査ルーチンの出力をネットログに入れるには、この AAO を YES に設定します。</p>
AOF_AAO_TWS_CMD_OUTPUT_NETLOG	<b>YES</b>  NO	<p>コマンド実行の出力をネットログに入れるには、この AAO を YES に設定します。</p>
AOF_AAO_TWS_ERRMSG		<p>この AAO を使用して ERRMSG パラメーターを禁止することができます。NON BLANK に設定すると、ERRMSG パラメーターの内容が削除されます。</p>
AOF_AAO_TWS_MAX_WAIT_TIME		<p>INGREQ コマンドおよび INGMOVE コマンドのインストール済み環境での最大待ち時間のデフォルトを定義します。完了情報パラメーターに待ち時間が指定されていない場合、デフォルトが使用されます。</p>



表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_AAO_TWS_RESYSLEX	YES NO	エンタープライズ全体の命名規則を簡便にするために、この AAO を使用して、TWS 特殊リソース名が SYSPLEX でなく SA z/OS シスプレックス名を使用できるようにします。 デフォルト: SYSPLEX に対して NO
AOF_AAO_VPCEINIT	0	SA z/OS は、GDPS 初期化出口 VPCEINIT を起動しません。
AOF_ASSIGN_JOBNAME	1	これは、SA z/OS が「ASSIGN BY MESSAGE ID」機能より高い優先順位で NetView「ASSIGN BY JOBNAME」機能を利用することを示します (優先順位 3)。 これはデフォルト設定です。
	0	SA z/OS は、「ASSIGN BY MESSAGE ID」機能より低い優先順位で NetView「ASSIGN BY JOBNAME」機能を利用します (優先順位 4)。
AOF_E2E_EAS_PPI	ユーザー定義	イベントをエンドツーエンド自動化アダプターに転送するのに使用されるイベント自動化サービスの PPI 受信側 ID。
AOF_E2E_EVT_RETRY	1 から n	これは、イベントを PPI TECROUTE 経由でイベント自動化サービスのメッセージ・アダプターへ転送するのに使用する再試行の回数を 1 秒間隔で指定します。これで、イベントはエンドツーエンド自動化アダプターに転送されます。
AOF_E2E_EXREQ_NETLOG	1	エンドツーエンド自動化アダプターから受信して 1 次自動化エージェントが発行する要求に対する出力は、NetView ログに記録されます。
	0	これらの要求に対する出力は、NetView ログに記録されません。 0 がデフォルトの設定値です。
AOF_E2E_TKOVV_TIMEOUT	hh:mm:ss	この変数で指定した値よりも自動化マネージャーのホット・リスタートに時間がかかる場合、エンドツーエンド自動化マネージャーは障害を通知され、第 1 レベル自動化で再同期化する必要があります。
AOF_EMCS_AUTOTASK_ASSIGNMENT	1	SA z/OS は、コンソール状況が MASTER または ACTIVE である 拡張 MCS コンソールに自動タスクを割り当てます。
	0	SA z/OS は、コンソール状況が MASTER または ACTIVE である 拡張 MCS コンソールに自動タスクを割り当てません。 0 がデフォルトです。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_EMCS_CN_ASSIGNMENT	1	SA z/OS は、オペレーター端末タスク (OST) の固有名称を持つ拡張 MCS コンソールを取得します。MVS コンソールが既に OST 用に前に取得されていると、そのコンソールは解放されます。 1 がデフォルトの設定値です。
	0	SA z/OS は OST の固有名称を持つ拡張 MCS コンソールを取得しません。コマンド AOCGETCN は使用できません。
AOF_FDBK_CS_HONOR (OA59461 で導入)	0	INGREQ/INGMOVE/INGGROUP FDBK コマンド処理は、リソースの予期される監視状況に達したときに、操作を「成功」と通知します。 0 がデフォルトの設定値です。
	1	INGREQ/INGMOVE/INGGROUP FDBK コマンド処理は、複合状況が SATISFACTORY で、監視状況が予期される状況と一致したときに、操作を「成功」と通知します。 また、CSONLY フラグ付き APG については、操作の複合状況が SATISFACTORY で、その監視状況が予期される状況で終わった、STARTING (開始アクションの場合) で終わった、または STOPPING/DEGRADED (どちらも停止アクションの場合) で終わったときに、操作は「完了」と通知されます。
AOF_PRESERVE_EXECUTION_CON TEXT	1	SA z/OS が RMTCMD インフラストラクチャーを使用している場合、発信側ユーザー ID は、ターゲット NetView 環境にログオンします。ログオンした後、そのユーザー ID は、目的のアクションを実行するために自動機能ユーザー ID に切り替わります。
	0	SA z/OS が RMTCMD インフラストラクチャーを使用している場合、発信側ユーザー ID は、RMTCMD 処理の間に自動機能ユーザー ID に切り替わります。 0 がデフォルトの設定値です。 この変数のセキュリティ考慮事項については、「TWS Automation Programmer's and Operator's Reference Guide」で『RMTCMD Security Considerations』を参照してください。
AOFACFINIT	1	これは、自動化制御ファイルの処理中に、SA z/OS がエラー・メッセージを無視して初期化を続けるよう試みることを示します。 1 がデフォルトの設定値です。
	0	SA z/OS は、このようなエラーの発生時に初期化プロセスを停止します。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOFARMQUERYRETRYS	ユーザー定義の数値	UNKNOWN 状況が戻された後に、AOFARMQ がエレメントの ARM 状況を照会するために呼び出される回数。最大再試行回数に達するまでに ARM 状況が別の状況に変わらない場合、SA z/OS は処理を続行し、エレメントが ARM 対応ではないと見なします。 デフォルトは 10 です。
AOFARMQUERYWAIT	ユーザー定義の数値	上記 AOFARMQUERYRETRYS 値で指定した次の再試行までの待機秒数。 デフォルトは 15 です。

変数	値	効果
AOFCNMASK	ユーザー定義	<p>固有のコンソール名の判別で使用する文字は、共通グローバル変数 AOFCNMASK を更新することによって調整できます。コマンド AOCGETCN によって固有のコンソール名を生成するときに、このグローバル変数を 16 進マスクとして使用して、以下のストリングから文字を取り出します。</p> <pre data-bbox="834 457 1446 569">left(opid(),8)  right(opid(),8),   left(aofsysname,4)  right(aofsysname,4),   left(applid(),8)  right(applid(),8),   'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789\$Y#@_!?',   aofcnmask_extended</pre> <p>ここで、</p> <ul style="list-style-type: none"> <li>• opid() は、OST タスク名を戻す関数です。</li> <li>• aofsysname は、システム名を保管する共通グローバルです。</li> <li>• applid() は、VTAM LU 名を戻す関数です。</li> <li>• aofcnmask_extended はオプションの共通グローバルであり、追加のユーザー定義 ID を保管できます (AOFCNMASK_EXTENDED を参照)。</li> </ul> <p>AOFCNMASK のデフォルトは 290C0D0E0F101718 です。X'29' は 41 桁目の文字 A を選択し、X'0C' から X'10' は 12 から 16 桁目の opid の下位 5 文字を選択し、X'17' および X'18' は 23 および 24 桁目の sysname の下位 2 文字を選択します。</p> <p>AOFCNMASK がヌルになっていると、AOCGETCN は、1 分の間隔の後、次に 2 分の間隔をおいて、以下同様に最大 5 パス (コマンドの初期呼び出しから 15 分経過) の後に、固有の拡張 MCS コンソールの取得を試みます。</p> <p>AOFCNMASK を調整するには、偶数の文字数で構成するようにならなければなりません。2 つの文字がそれぞれ 1 つの 16 進数を表すためです。そうでないと、固有コンソール名を生成できません。</p> <p>以下は一例です。</p> <pre data-bbox="834 1472 1179 1493">AOFCNMASK: 2A01020304055455</pre> <p>X'2A' は 42 桁目の文字 B を選択し、X'01' から X'05' は 1 から 5 桁目の opid の上位 5 文字を選択し、X'54' および X'56' は 84 および 85 桁目の sysname の下位 2 文字を選択します。</p> <p><b>注:</b> ここで指定する値は、NetView ConsMask スタイル・シート・パラメーターの指定よりも優先されます。</p>

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOFCNMASK_EXTENDED	ユーザー定義	固有コンソール名の生成用の追加 ID。 この変数は、固有コンソール名の生成に追加のユーザー定義 ID が必要な場合 (例えば、前のコンソール名が固有でなかった場合) に、NetView スタイルシートで個々に設定できます。 この場合、AOFCNMASK_EXTENDED に値を割り当てる必要があり、必要な文字を AOFCNMASK_EXTENDED から抽出するために共通グローバル変数 AOFCNMASK を調整する必要があります (AOFCNMASK を参照)。 固有コンソール名の生成に追加のユーザー定義 ID が不要な場合、この共通グローバルを使用する必要はなく、コンソール名は通常の方法で生成されます。 最大長は 172 文字です。
AOFDEFAULT_TARGET	ユーザー定義	TARGET パラメーターが使用されるすべてのコマンドについて、このパラメーターにデフォルトを設定します。
AOFDESCA	010000100000 1000	処置メッセージの記述子コード
AOFDESCD	010000100000 1000	判断メッセージの記述子コード
AOFDESCE	001000100000 1000	結果としての処置メッセージの記述子コード
AOFDESCI	000001100000 1000	通知メッセージの記述子コード
AOFDESCW	100000100000 1000	待機メッセージの記述子コード
AOFEXPLAIN_USER	ユーザー定義	EXPLAIN コマンドは、この変数を受け入れ、カスタマー・インストール提供の用語にヘルプ・サポートを組み込みます。1 つ以上の用語/ヘルプ・パネル 指定の対を、ブランクで区切って持つことができます。 EXPLAIN コマンドで指定された状況が有効な SA z/OS の状況でない場合、コマンド・ルーチンは、それがインストール・システムで定義された用語かどうか検査します。インストール・システムで定義された用語であれば、関連したヘルプ・パネルが表示されます。
AOFINITREPLY	hh:mm:ss	初期応答 AOF603D が発行され、hh:mm:ss 後に自動的に応答されます。 <b>00:02:00</b> (2 分) がデフォルト設定です。
	0	初期応答 AOF603D は出されず、オペレーターに要求しないまま、デフォルトの始動で自動化が継続します。
AOF_INIT_MCSFLAG	ユーザー定義の有効値	この変数は、初期化時に SA z/OS が出す WTO および WTOR に使用される MCSFLAG を含みます。 デフォルトは「00000000」です。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOF_INIT_ROUTCDE	ユーザー定義の有効値	この変数は、初期化時に SA z/OS が出す WTO および WTOR に使用される ROUTCDE (宛先コード) を含みません。 デフォルトは「01000000」です。
AOF_INIT_SYSCONID	ユーザー定義の有効値	この変数は、初期化時に SA z/OS が出す WTO および WTOR に使用される SYSCONID を含みます。 デフォルトはブランクです。
AOFLOCALHOLD	0	INGNTFY および SA z/OS 初期化で、通知オペレーターに対して SETHOLD AUTO コマンドを実行します。 0 がデフォルトの設定値です。
	1	SETHOLD は、手動で呼び出す必要があります。
AOFMATLISTING	0	この変数を 0 に設定することは、NetView 自動化テーブルのロード時に、NetView 自動化テーブル・リストが DSILIST データ・セットに入れられないことを意味します。
	未定義、または 0 以外のすべての値	この変数を設定しないか、0 以外の値に設定すると、NetView 自動化テーブル・リストが DSILIST データ AOFMATLISTING セットに入れられます。
AOFOPCCMDMSG	0	OPCAMOD は、INGOPC によって生成されるメッセージの出力のみを行います。 0 がデフォルトの設定値です。
	1	OPCAMOD は、EVJ011I、EVJ412I、EVJ420I、および EVJ423I メッセージを作成します。
AOFRESTARTALWAYS	1	SA z/OS の制御外で、RESTARTOPT=ALWAYS を指定して、正常にシャットダウンされたアプリケーションは、そのクリティカル・エラーしきい値に達したかどうかとは無関係に再始動されることとなります。
	0	RESTARTOPT=ALWAYS が指定されていて、SA z/OS の制御外で正常にシャットダウンされたアプリケーションは、そのクリティカル・エラーしきい値に達した場合、再始動されないこととなります。 0 がデフォルトの設定値です。
AOFRMTCMDWAIT	NetView の RMTCMD を参照	RMTCMD が通信用に使用された場合に、インストールの待ち時間が入ります。 RMTCMD の場合のデフォルトの設定値は <b>60 秒</b> です。
AOFRPCWAIT	0 から n	これは、SA z/OS がシスプレックス内の他のシステムからのコマンド応答を待つ秒数です。 <b>10</b> がデフォルトの設定値です。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOFSENDALERT	<b>Yes</b> または <b>No</b>	これは、フォーカル・ポイントにデータを転送するのに、NetView アラート転送が使用される (YES) のか、コマンド・ハンドラーが使用される (NO) のかを決定します。 <b>Yes</b> がデフォルトの設定値です。
AOFSEXINT	<b>1</b>	出口 AOFEXINT は、初期化プロセスのものと BASEOPER 自動化オペレーターで処理されます。これがデフォルトです。
	0	出口 AOFEXINT の実行は、初期化プロセス内でシリアルライズされます。
AOFSHUTDELAY	0 から 59	これは、SA z/OS が、シャットダウン・プロセスを続行する前に、終了メッセージを待つ分数です。この範囲外の値はすべて 0 として扱われます。0 に設定すると、メッセージ AOF745E は出されません。 <b>0</b> がデフォルトの設定値です。



表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOFSMARTMAT	0	SA z/OS エージェントは AT と MRT を最新表示しません。SA z/OS が最初に INGMMSG01 をロードするときに組み込まれる AT メンバー INGMMSG02 をユーザーが指定する必要があります。
	1	INGAMS REFRESH が出されたときに、SA z/OS エージェントは AT を最新表示できます。カスタマイズ・ダイアログで構築された AT フラグメントはロードされず、代わりに使用するメンバー INGMMSG02 をユーザーが指定する必要があります。  テスト・ロードが成功すると、AT がロードされます。これにより、エージェントは AT のロードの問題を AM に通知できるようになります。エージェントは AM に AT ロード障害を知らせ、構成の最新表示を停止します。SA z/OS エージェントは MRT をロードしません。
	2	INGAMS REFRESH が出されたときに、SA z/OS エージェントはカスタマイズ・ダイアログで生成された AT をロードし、AT を最新表示できます。カスタマイズ・ダイアログによって構築された AT は、INGMSG02 フラグメントとして動的にストレージにロードされます。  テスト・ロードが成功すると、AT がロードされます。これにより、エージェントは AT のロードの問題を AM に通知できるようになります。エージェントは AM に AT ロード障害を知らせ、構成の最新表示を停止します。SA z/OS エージェントは MRT をロードしません。  これはデフォルト値です。
	3	INGAMS REFRESH が出されたときに、SA z/OS 自動化エージェントはカスタマイズ・ダイアログで生成された MRT をロードし、MRT を最新表示できます。カスタマイズ・ダイアログで構築された AT は、INGMSG02 フラグメントとして動的にストレージにロードされます。カスタマイズ・ダイアログで構築された MRT は、INGMRT01 として動的にロードされます。テスト・ロードが成功すると、AT と MRT がロードされます。MRT でエラーが検出されると、エージェントの初期化が中断されず、MRT がロードされません。構成の最新表示が要求されたときにエラーが検出されると、最新表示は実行されません。
AOFSPoolFULLCMD	1	SA z/OS は、スプール・リカバリー・パスを一度しか実行しません。SPOOLFULL の状態が続くと、メッセージ AOF2941I が出されます。
	0	SA z/OS は、スプール・リカバリー・コマンドを再実行します。  0 がデフォルトの設定値です。

表 26. 高機能の自動化を可能にするグローバル変数 (CGLOBALS) (続き)		
変数	値	効果
AOFSPoolSHORTCMD	1	SA z/OS は、スプール・リカバリー・パスを一度しか実行しません。SPOOLSHORT の状態が続くと、メッセージ AOF2941I が出されます。
	0	SA z/OS は、スプール・リカバリー・コマンドを再実行します。 0 がデフォルトの設定値です。
AOFSTATUSCMDSEL	0	指定された選択値にかかわらず、新規状況と関連するすべての状況コマンドまたは応答を発行します。マイナー・リソース <i>subsystem.status</i> に対する、選択基準を派生させるための、あるいはクリティカル (critical) のしきい値を超えた場合にコマンドまたは応答を発行しないようにするための、しきい値の検査は行われません。  AOFSTATUSCMDSEL が設定されない場合、または 0 以外の値に設定される場合、starttype または stoptype などの特定の選択基準が指定されたコマンドまたは応答のみが発行されます。
AOFUSSWAIT	1 から <i>n</i>	これは、タイムアウトを受信するまでに、SA z/OS がユーザー定義 z/OS UNIX モニター・ルーチンの完了を待つ秒数です (「z/OS UNIX 制御の指定 (z/OS UNIX Control Specification)」パネルで指定)。タイムアウトが発生した場合、SA z/OS はモニター・ルーチンからの応答を待たずに、SIGKILL をモニター・ルーチンに送信します。  10 がデフォルトの設定値です。
INGCICS_CORRWAIT	ユーザー定義の数値	INGCICS が CICS トランザクションからの出力を待つ秒数です。指定されていない場合、INGCICS はデフォルトの CORRWAIT (CCDEF) 値を使用します。
INGIMS_CORRWAIT	ユーザー定義の数値	INGIMS が IMS コマンドからの出力を待つ秒数です。指定されていない場合、INGIMS はデフォルトの CORRWAIT (CCDEF) 値を使用します。
INGOPC_MULTIPLIER	1 から <i>n</i>	これは、終了までの待機時間を判別するため、AOFRTMCDWAIT および AOFRPCWAIT とともに使用されます。
INGPAC_SHOWNOLIMIT	1	並行性制限 NOLIMIT を持つペーシング・ゲートが常に INGPAC 表示に示されます。これがデフォルトです。
	0	並行性制限 NOLIMIT を持つペーシング・ゲートは INGPAC 表示に示されません。
INGRAITF_WAIT	ユーザー定義の数値	INGRAITF ルーチンが待機する秒数です。

変数	値	効果
INGREQ_ORIGINATOR	1	INGREQ コマンドを実行する各オペレーターの個別の発信元 ID を SA z/OS が割り当てることを指示します。
	0	すべてのオペレーターが発信元 ID OPERATOR の下にグループ化されます。 0 がデフォルトの設定値です。

## コマンドのパラメーター・デフォルト

変数名	説明	参照 <sup>1</sup>
AOFSETSTATESCOPE	SETSTATE コマンドの SCOPE パラメーターの事前定義デフォルトをオーバーライドすることができます。	SETSTATE
DISPEVT_WAIT	DISPEVT コマンドの WAIT パラメーターを指定値に設定します。	DISPEVT
DISPEVTS_WAIT	DISPEVTS コマンドの WAIT パラメーターを指定値に設定します。	DISPEVTS
DISPGW_COLUMNS	DISPGW コマンドのパネル列レイアウトを指定します。 これはステムであり、意味は次のとおりです。 <ul style="list-style-type: none"> <li>DISPGW_COLUMNS.0 にはエレメント数が含まれています</li> <li>DISPGW_COLUMNS.n には n 番目の列の定義が含まれています</li> </ul> 列定義はそれぞれ、ブランクで区切られた、次の順序のトークンのリストです。 <ul style="list-style-type: none"> <li><i>prefix</i> は以下のような列の接頭部標識です。               <ul style="list-style-type: none"> <li>I 列が「移動不可」である場合 (「事前固定」であることを示す)</li> <li>P 列が水平スクロールのための「事前固定」である場合</li> <li>F 列が「浮動」である (つまりスクロールされる) 場合</li> <li>- 列が非表示である場合</li> </ul> </li> <li><i>sortorder</i> は以下のような列のソート順です。               <ul style="list-style-type: none"> <li>A 列が昇順でソートされている場合</li> <li>D 列が降順でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>sortkey</i> は以下のような列のソート・キーです。               <ul style="list-style-type: none"> <li>1-n 列が n 番目の位置でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>columnname</i> は列の名前です。これらは複数の語にできます。この名前は列の参照に使用されるため、列名を変更することはできません。</li> </ul> この名前の TGLOBALS はユーザー固有の設定であり、インストール・デフォルトを上書きします。レイアウトを TGLOBALS に保存し、必要に応じて CGLOBALS を派生させることをお勧めします。無効な定義は通知されることなく無視されます。	DISPGW

表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
DISPMTR_COLUMNS	<p>DISPMTR コマンドのパネル列レイアウトを指定します。 これはステムであり、意味は次のとおりです。</p> <ul style="list-style-type: none"> <li>DISPMTR_COLUMNS.0 にはエレメント数が含まれています</li> <li>DISPMTR_COLUMNS.n には n 番目の列の定義が含まれています</li> </ul> <p>列定義はそれぞれ、ブランクで区切られた、次の順序のトークンのリストです。</p> <ul style="list-style-type: none"> <li><i>prefix</i> は以下のような列の接頭部標識です。 <ul style="list-style-type: none"> <li>I 列が「移動不可」である場合 (「事前固定」であることを示す)</li> <li>P 列が水平スクロールのための「事前固定」である場合</li> <li>F 列が「浮動」である (つまりスクロールされる) 場合</li> <li>- 列が非表示である場合</li> </ul> </li> <li><i>sortorder</i> は以下のような列のソート順です。 <ul style="list-style-type: none"> <li>A 列が昇順でソートされている場合</li> <li>D 列が降順でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>sortkey</i> は以下のような列のソート・キーです。 <ul style="list-style-type: none"> <li>1-n 列が n 番目の位置でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>columnname</i> は列の名前です。これらは複数の語にできます。この名前は列の参照に使用されるため、列名を変更することはできません。</li> </ul> <p>この名前の TGLOBALS はユーザー固有の設定であり、インストール・デフォルトを上書きします。レイアウトを TGLOBALS に保存し、必要に応じて CGLOBALS を派生させることをお勧めします。無効な定義は通知されることなく無視されます。</p>	DISPMTR

表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
DISPSTAT_COLUMNS	<p>DISPSTAT コマンドのパネル列レイアウトを指定します。</p> <p>これはステムであり、意味は次のとおりです。</p> <ul style="list-style-type: none"> <li>DISPSTAT_COLUMNS.0 にはエレメント数が含まれています</li> <li>DISPSTAT_COLUMNS.n には n 番目の列の定義が含まれています</li> </ul> <p>列定義はそれぞれ、ブランクで区切られた、次の順序のトークンのリストです。</p> <ul style="list-style-type: none"> <li><i>prefix</i> は以下のような列の接頭部標識です。 <ul style="list-style-type: none"> <li>I 列が「移動不可」である場合 (「事前固定」であることを示す)</li> <li>P 列が水平スクロールのための「事前固定」である場合</li> <li>F 列が「浮動」である (つまりスクロールされる) 場合</li> <li>- 列が非表示である場合</li> </ul> </li> <li><i>sortorder</i> は以下のような列のソート順です。 <ul style="list-style-type: none"> <li>A 列が昇順でソートされている場合</li> <li>D 列が降順でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>sortkey</i> は以下のような列のソート・キーです。 <ul style="list-style-type: none"> <li>1-n 列が n 番目の位置でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>columnname</i> は列の名前です。これらは複数の語にできます。この名前は列の参照に使用されるため、列名を変更することはできません。</li> </ul> <p>この名前の TGLOBALS はユーザー固有の設定であり、インストール・デフォルトを上書きします。レイアウトを TGLOBALS に保存し、必要に応じて CGLOBALS を派生させることをお勧めします。無効な定義は通知されることなく無視されます。</p>	DISPSTAT
DISPTRG_WAIT	DISPTRG コマンドの WAIT パラメーターを指定値に設定します。	DISPTRG

表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
EVIRYDPO_COLUMNS	<p>INGIMS REQ=DEP コマンドのパネル列レイアウトを指定します。これはステムであり、意味は次のとおりです。</p> <ul style="list-style-type: none"> <li>• EVIRYDPO_COLUMNS.0 にはエレメント数が含まれています</li> <li>• EVIRYDPO_COLUMNS.n には n 番目の列の定義が含まれています</li> </ul> <p>列定義はそれぞれ、ブランクで区切られた、次の順序のトークンのリストです。</p> <ul style="list-style-type: none"> <li>• <i>prefix</i> は以下のような列の接頭部標識です。 <ul style="list-style-type: none"> <li>I 列が「移動不可」である場合 (「事前固定」であることを示す)</li> <li>P 列が水平スクロールのための「事前固定」である場合</li> <li>F 列が「浮動」である (つまりスクロールされる) 場合</li> <li>- 列が非表示である場合</li> </ul> </li> <li>• <i>sortorder</i> は以下のような列のソート順です。 <ul style="list-style-type: none"> <li>A 列が昇順でソートされている場合</li> <li>D 列が降順でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li>• <i>sortkey</i> は以下のような列のソート・キーです。 <ul style="list-style-type: none"> <li>1-n 列が n 番目の位置でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li>• <i>columnname</i> は列の名前です。これらは複数の語にできます。この名前は列の参照に使用されるため、列名を変更することはできません。</li> </ul> <p>この名前の TGLOBALS はユーザー固有の設定であり、インストール・デフォルトを上書きします。レイアウトを TGLOBALS に保存し、必要に応じて CGLOBALS を派生させることをお勧めします。無効な定義は通知されることなく無視されます。</p>	INGIMS

表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
INGAMS_COLUMNS	<p>INGAMS コマンドのパネル列レイアウトを指定します。</p> <p>これはステムであり、意味は次のとおりです。</p> <ul style="list-style-type: none"> <li>INGAMS_COLUMNS.0 にはエレメント数が含まれています</li> <li>INGAMS_COLUMNS.n には n 番目の列の定義が含まれています</li> </ul> <p>列定義はそれぞれ、ブランクで区切られた、次の順序のトークンのリストです。</p> <ul style="list-style-type: none"> <li><i>prefix</i> は以下のような列の接頭部標識です。 <ul style="list-style-type: none"> <li>I 列が「移動不可」である場合 (「事前固定」であることを示す)</li> <li>P 列が水平スクロールのための「事前固定」である場合</li> <li>F 列が「浮動」である (つまりスクロールされる) 場合</li> <li>- 列が非表示である場合</li> </ul> </li> <li><i>sortorder</i> は以下のような列のソート順です。 <ul style="list-style-type: none"> <li>A 列が昇順でソートされている場合</li> <li>D 列が降順でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>sortkey</i> は以下のような列のソート・キーです。 <ul style="list-style-type: none"> <li>1-n 列が n 番目の位置でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>columnname</i> は列の名前です。これらは複数の語にできます。この名前は列の参照に使用されるため、列名を変更することはできません。</li> </ul> <p>この名前の TGLOBALS はユーザー固有の設定であり、インストール・デフォルトを上書きします。レイアウトを TGLOBALS に保存し、必要に応じて CGLOBALS を派生させることをお勧めします。無効な定義は通知されることなく無視されます。</p>	INGAMS
INGAUTO_INTERVAL	INGAUTO コマンドの INTERVAL パラメーターのデフォルトを設定します。	INGAUTO
INGEVENT_WAIT	INGEVENT コマンドの WAIT パラメーターを指定値に設定します。このパラメーターは、要求が完了するまで待機するかどうかを指定します。	INGEVENT
INGEXEC_RESP	INGEXEC コマンドの RESP パラメーターを指定値に設定します。	INGEXEC
INGEXEC_SELECT	INGEXEC コマンドの SELECT パラメーターを指定値に設定します。	INGEXEC
INGEXEC_TIMEOUT	INGEXEC コマンドの TIMEOUT パラメーターを指定値に設定します。	INGEXEC
INGEXEC_WAIT	INGEXEC コマンドの WAIT パラメーターを指定値に設定します。	INGEXEC
INGGROUP_WAIT	INGGROUP コマンドの WAIT パラメーターを指定値に設定します。このパラメーターは、要求が完了するまで待機するかどうかを指定します。	INGGROUP
INGHIST_MAX	INGHIST コマンドの MAX パラメーターを指定値に設定します。	INGHIST
INGHIST_WIMAX	INGHIST コマンドの WIMAX パラメーターを指定値に設定します。	INGHIST
INGIMS_CMDWAIT	INGIMS コマンドの CMDWAIT パラメーター (コマンドが完了するまでの最大待ち時間) を指定します。	INGIMS



表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
INGIMS_REQ	INGIMS コマンドの REQ パラメーター (IMS サブシステムに発行される要求) を指定します。	INGIMS
INGINFO_WAIT	INGINFO コマンドの WAIT パラメーターを指定値に設定します。	INGINFO
INGINFO_RSTAT	INGINFO コマンドの RSTAT パラメーターを指定値に設定します。	INGINFO
INGLKUP_TIMEOUT	INGLKUP コマンドの TIMEOUT パラメーターを指定値に設定します。	INGLKUP
INGLKUP_WAIT	INGLKUP コマンドの WAIT パラメーターを指定値に設定します。	INGLKUP
INGLIST_COLUMNS	<p>INGLIST コマンドのパネル列レイアウトを指定します。</p> <p>これはステムであり、意味は次のとおりです。</p> <ul style="list-style-type: none"> <li>INGLIST_COLUMNS.0 にはエレメント数が含まれています</li> <li>INGLIST_COLUMNS.n には n 番目の列の定義が含まれています</li> </ul> <p>列定義はそれぞれ、ブランクで区切られた、次の順序のトークンのリストです。</p> <ul style="list-style-type: none"> <li><i>prefix</i> は以下のような列の接頭部標識です。 <ul style="list-style-type: none"> <li>I 列が「移動不可」である場合 (「事前固定」であることを示す)</li> <li>P 列が水平スクロールのための「事前固定」である場合</li> <li>F 列が「浮動」である (つまりスクロールされる) 場合</li> <li>- 列が非表示である場合</li> </ul> </li> <li><i>sortorder</i> は以下のような列のソート順です。 <ul style="list-style-type: none"> <li>A 列が昇順でソートされている場合</li> <li>D 列が降順でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>sortkey</i> は以下のような列のソート・キーです。 <ul style="list-style-type: none"> <li>1-n 列が n 番目の位置でソートされている場合</li> <li>- 列がソートされていないか、非表示である場合</li> </ul> </li> <li><i>columnname</i> は列の名前です。これらは複数の語にできます。この名前は列の参照に使用されるため、列名を変更することはできません。</li> </ul> <p>この名前の TGLOBALS はユーザー固有の設定であり、インストール・デフォルトを上書きします。レイアウトを TGLOBALS に保存し、必要に応じて CGLOBALS を派生させることをお勧めします。無効な定義は通知されることなく無視されます。</p>	INGLIST
INGLIST_WAIT	INGLIST コマンドの WAIT パラメーターを指定値に設定します。	INGLIST
INGMON_WAIT	INGMON コマンドの WAIT パラメーターを指定値に設定します。	INGMON
INGMOVE_WAIT	INGMOVE コマンドの WAIT パラメーターを指定値に設定します。	INGMOVE
INGPAC_WAIT	INGPAC コマンドの WAIT パラメーターを指定値に設定します。	INGPAC
INGRELS_SHOW	INGRELS コマンドの SHOW パラメーターを指定値に設定します。	INGRELS
INGRELS_WAIT	INGRELS コマンドの WAIT パラメーターを指定値に設定します。	INGRELS
INGREQ_BOOST	INGREQ コマンドのデフォルトの BOOST パラメーターを指定値に設定します。	INGREQ

表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
INGREQ_EXPIRE	INGREQ コマンドのデフォルトの EXPIRE パラメーターを指定値に設定します。 相対時刻 (+4:00 など) にのみ設定でき、絶対有効期限日時には設定できません。	INGREQ
INGREQ_INTERRUPT	INGREQ コマンドのデフォルトの INTERRUPT パラメーターを指定値に設定します。 このパラメーターは、リソースが UP 状態に達するまで、自動化マネージャーが待機するかどうかを指定しますが、より高い優先順位の停止要求が出された場合は、リソースは始動フェーズのままになります。	INGREQ
INGREQ_OVERRIDE	INGREQ コマンドのデフォルトの OVERRIDE パラメーターを指定値に設定します。	INGREQ
INGREQ_PRECHECK	INGREQ コマンドのデフォルトの PRECHECK パラメーターを指定値に設定します。	INGREQ
INGREQ_PRI	INGREQ コマンドのデフォルトの優先順位 (PRI パラメーター) を指定値に設定します。	INGREQ
INGREQ_PRI.E2EMGR	エンドツーエンド自動化マネージャーからの着信要求を実行する優先順位を指定します。 デフォルトは、LOW です。	INGREQ
INGREQ_REMOVE	INGREQ コマンドの REMOVE パラメーターのデフォルト値を設定します。 リソースが指定された状況 (状態) に達すると、この要求は自動的に除去されます。	INGREQ
INGREQ_REMOVE.START	INGREQ START コマンドの REMOVE パラメーターのデフォルト値を設定します。 指定されない場合、INGREQ_REMOVE によって設定される値が使用されます。	INGREQ
INGREQ_REMOVE.STOP	INGREQ STOP コマンドの REMOVE パラメーターのデフォルト値を設定します。 指定されない場合、INGREQ_REMOVE によって設定される値が使用されます。	INGREQ
INGREQ_RESTART	リソースのシャットダウン時に、INGREQ コマンドの RESTART パラメーターのデフォルトを設定します。	INGREQ
INGREQ_SCOPE	INGREQ コマンドの SCOPE パラメーターを指定値に設定します。	INGREQ
INGREQ_SOURCE	INGREQ コマンドのデフォルトの SOURCE パラメーターを指定値に設定します。 このパラメーターは、要求の発信元を指定します。	INGREQ
INGREQ_TIMEOUT	要求が正常に完了したかどうかを確認し、さらに、その時以降に要求が満たされない場合にメッセージを送信するのがあるいはその要求を取り消すのを確認するために使用する INGREQ コマンドについて、確認の間隔 (分) を設定します。	INGREQ
INGREQ_TYPE	INGREQ コマンドのデフォルトの始動/シャットダウンのタイプ (TYPE パラメーター) を指定値に設定します。	INGREQ
INGREQ_VERIFY	INGREQ コマンドのデフォルトの VERIFY パラメーターを指定値に設定します。	INGREQ
INGREQ_WAIT	INGREQ コマンドの WAIT パラメーターを指定値に設定します。	INGREQ
INGRPT_WAIT	INGRPT コマンドの WAIT パラメーターを指定値に設定します。	INGRPT
INGRUN_CMT	INGRUN コマンドの CMT パラメーターを指定値に設定します。	INGRUN
INGRUN_MULT	INGRUN コマンドの MULT パラメーターを指定値に設定します。	INGRUN

表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
INGRUN_OVERRIDE	INGRUN コマンドの OVERRIDE パラメーターを指定値に設定します。	INGRUN
INGRUN_PERSISTENT	INGRUN コマンドの PERSISTENT パラメーターを指定値に設定します。	INGRUN
INGRUN_PRI	INGRUN コマンドの PRI パラメーターを指定値に設定します。	INGRUN
INGRUN_REQ	INGRUN コマンドの REQ パラメーターを指定値に設定します。	INGRUN
INGRUN_RUNMODE	INGRUN コマンドの RUNMODE パラメーターを指定値に設定します。	INGRUN
INGRUN_RUNRES	INGRUN コマンドの RUNRES パラメーターを指定値に設定します。	INGRUN
INGRUN_SYSTEM	INGRUN コマンドの SYSTEM パラメーターを指定値に設定します。	INGRUN
INGRUN_TARGET	INGRUN コマンドの TARGET パラメーターを指定値に設定します。	INGRUN
INGRUN_TYPE	INGRUN コマンドの TYPE パラメーターを指定値に設定します。	INGRUN
INGRUN_VERIFY	INGRUN コマンドの VERIFY パラメーターを指定値に設定します。	INGRUN
INGRUN_WAIT	INGRUN コマンドの WAIT パラメーターを指定値に設定します。	INGRUN
INGSCHED_WAIT	INGSCHED コマンドの WAIT パラメーターを指定値に設定します。このパラメーターは、要求が完了するまで待機するかどうかを指定します。	INGSCHED
INGSET_VERIFY	INGSET コマンドのデフォルトの VERIFY パラメーターを指定値に設定します。	INGSET
INGSET_WAIT	INGSET コマンドの WAIT パラメーターを指定値に設定します。このパラメーターは、要求が完了するまで待機するかどうかを指定します。	INGSET
INGSTX_WAIT	INGSTX コマンドの WAIT パラメーターを指定値に設定します。	INGSTX
INGSUSPD_EXPIRE	これは、INGSUSPD コマンドのデフォルト EXPIRE パラメーターを指定値に設定します。相対時刻 (+4:00 など) にのみ設定でき、絶対有効期限日時には設定できません。	INGSUSPD
INGSUSPD_REMOVE	INGSUSPD コマンドの REMOVE パラメーターのデフォルト値を設定します。リソースが指定された状況 (状態) に達すると、この要求は自動的に除去されます。	INGSUSPD
INGSUSPD_SCOPE	これは、INGSUSPD コマンドの SCOPE パラメーターを指定値に設定します。	INGSUSPD
INGSUSPD_TIMEOUT	これは、要求が正常に完了したかどうかを INGSUSPD コマンドが検査する間隔 (分) を設定します。また、これは、その時間が経過した後で要求が満たされていない場合にメッセージを送信するか要求を取り消すのかを設定します。	INGSUSPD
INGSUSPD_VERIFY	これは、INGSUSPD コマンドのデフォルト VERIFY パラメーターを指定値に設定します。	INGSUSPD
INGSUSPD_WAIT	これは、INGSUSPD コマンドの WAIT パラメーターを指定値に設定します。	INGSUSPD
INGTRIG_WAIT	INGTRIG コマンドの WAIT パラメーターを指定値に設定します。	INGTRIG

表 27. 特定コマンドのインストール済み環境デフォルトを定義するグローバル変数 (続き)		
変数名	説明	参照 <sup>1</sup>
INGVOTE_EXCLUDE	INGVOTE コマンドの EXCLUDE パラメーターを指定値に設定します。このパラメーターは、すべての要求を表示する場合に除外するリソース・タイプ (例えば、SVP や GRP など) を指定します。そのタイプのリソースはフィルタリングされます。	INGVOTE
INGVOTE_SHOW	INGVOTE コマンドの SHOW パラメーターを指定値に設定します。	INGVOTE
INGVOTE_SOURCE	INGVOTE コマンドのデフォルトの SOURCE パラメーターを指定値に設定します。	INGVOTE
INGVOTE_STATUS	INGVOTE コマンドの STATUS パラメーターを指定値に設定します。このパラメーターは、表示する必要がある要求 (採用、不採用、またはすべて) を指定します。	INGVOTE
INGVOTE_WAIT	INGVOTE コマンドの WAIT パラメーターを指定値に設定します。	INGVOTE
INGWHY_TIMEOUT	INGWHY コマンドの TIMEOUT パラメーターを指定値に設定します。	INGWHY
INGWHY_WAIT	INGWHY コマンドの WAIT パラメーターを指定値に設定します。	INGWHY
1. 「IBM Z System Automation オペレーター・コマンド」に記載されているコマンドを参照してください。		

## 付録 B 状況表示機能のカスタマイズ

### 状況表示機能の概要

この情報では、状況表示機能 (SDF) パネル、記述子、および操作をカスタマイズする方法について説明します。

### 状況表示機能の動作方法

SA z/OS の状況表示機能 (SDF) は、カラーと強調表示を使用して、サブシステム・リソースの状態を表します。

一般に、SDF の状況パネル上で、緑で示されるサブシステムは、UP になっていることを示しているのに対し、赤は STOPPED または問題のある状態のサブシステムを示します。SDF は、システム・コンポーネントの状況を階層表現で表すように調整することができます。

### SDF パネルのタイプ

230 ページの図 48 は、システム CHI01 用のいくつかの SDF 画面を示しています。この図は、SDF で使用する主なパネルのタイプを示します。

- ルート・コンポーネント
- 状況コンポーネント
- 詳細状況表示

これらのパネルの他に、ユーザーのシステム要件およびモニターするアプリケーションに従って、他のタイプのパネルを作成することができます。

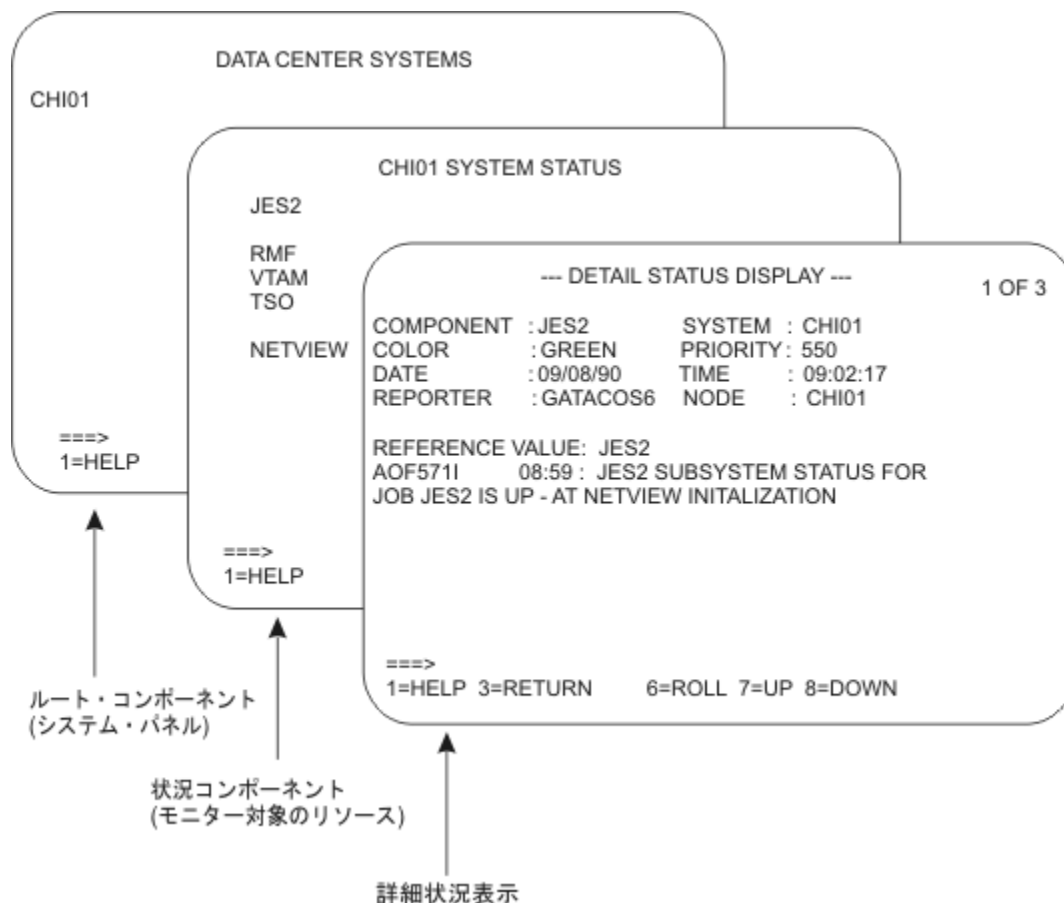


図 48. SDF パネルの例

### ルート・コンポーネント

一般に、ルート・コンポーネントとは、SDF を始動すると最初に表示される画面に表われる エレメントのことです。230 ページの図 48 では、CHI01 システムがルート・コンポーネントです。

### 状況コンポーネント

SDF によってモニターされるリソースは、状況コンポーネントと呼ばれます。230 ページの図 48 では、CHI01 システム 状況表示パネルに示されているように、システム CHI01 には JES2、RMF、VTAM、TSO、および NetView の状況コンポーネントがあります。状況コンポーネント・パネルには、システム内でモニターされるリソースすべてが表示されます。モニターされる各リソースは、その現行状況のカラーで表示されます。例えば、JES2 は、UP になっていると、緑で表示されます。

### 詳細状況表示

詳細状況ディスプレイは、情報記述子 (230 ページの『状況記述子』を参照) の情報から構築されます。このパネルは、状況コンポーネント・パネル上の該当リソースにタブ移動し、「詳細」PF キーを押すことによって表示されます。各状況コンポーネントは、状況コンポーネントと関連した、1 つ以上の状況記述子または明細レコードを持つことができます。

230 ページの図 48 は、JES2 状況記述子の詳細状況ディスプレイの例を示しています。パネル上の「1 of 3」は、JES2 には現在 3 つの状況記述子があり、したがって、JES2 に関連した 3 つの詳細状況ディスプレイがあることを示しています。

### 状況記述子

状況記述子は、リソース情報についての情報の明細レコードです。その未加工の形式では、状況記述子は、以下のような情報を含む複数行の SA z/OS のメッセージです。

- 状況記述子の適用先のルート・コンポーネントおよび状況コンポーネント。

- ・ 状況記述子と関連した優先順位、カラー、および強調表示 (詳細については、[232 ページの『状況記述子が SDF に及ぼす影響』](#)を参照してください)。
- ・ 状況記述子が生成された日時。
- ・ 実際のリソース状況情報。例えば、リソースが UP になっていることを示す SA z/OS メッセージ。

SDF は、状況記述子の情報を使用して、詳細状況ディスプレイを生成します ([230 ページの『詳細状況表示』](#)を参照してください)。通常、状況記述子を直接見ることはありませんが、詳細状況ディスプレイでその一部を見ることになります。例えば、[230 ページの図 48](#)では、詳細状況ディスプレイには、状況コンポーネント JES2 の状況記述子からの情報が表示されています。パネル上の「1 of 3」は、現在 JES2 には、関連した 3 つの状況記述子があることを示しています。

SDF は、状況記述子の生成、表示、および削除を行います。

## SDF ツリー構造

SDF は、ツリー構造を使用して、SDF 状況パネルに表示される モニター・リソースの階層を設定します。SDF ツリー構造は通常、ルート・ノードとしてシステム名で始まり、これがレベル番号 1 になります。ルート・ノードに従属するツリー構造レベルがモニター・リソースです。これらのリソースのレベル番号は、相互の依存要素を反映します。

パフォーマンス上の理由から、ツリー構造全体がデータ・スペースに保持されます。データ・スペースのサイズは、実際には SDF の使用状況によって異なります。データ・スペースのサイズの計算および定義については、「*IBM Z System Automation プログラマーズ・リファレンス*」の MAXTREEDSPSZ パラメーターの説明を参照してください。

SDF ツリー構造は、NetView DSIPARM データ・セット・メンバー AOFTREE に定義します。

[231 ページの図 49](#) は、SDF ツリー構造例を示しています。[230 ページの図 48](#) は、これらのステートメントがどのようにしてツリー構造になるかを示しています。

```

1 SY1
  2 SYSTEM
    3 WTOR
    3 APPLIC
      4 AOFAPPL
        5 AOFSSI
      4 JES
      4 VTAM
    3 TSO
    3 RMF
  2 GATEWAY
  2 MONITOR
  2 APG
  3 GROUP

```

図 49. SDF ツリー構造の例

SA z/OS は、SA z/OS サンプル・ライブラリーに、サンプル SDF ツリー構造を提供します。このツリー構造は、NetView DSIPARM データ・セットのメンバー AOFTREE の %INCLUDE ステートメントによって参照されます。このサンプル・ツリー構造を、ユーザー要件を満たすようにカスタマイズすることができます。依存要素のこの順序は、SA z/OS を使用したシステムの始動またはシャットダウンに使用するものと同じである必要はありません。ツリー構造では、システム・シンボルを使用できます。これにより、カスタマイズ作業とエラーの両方を削減することができます。

例えば、[231 ページの図 49](#) のツリー構造を使用して、TSO に問題があった場合、VTAM にはなんら問題がないので、VTAM 状況カラーも変更されるのは望ましくありません。これに対し、SA z/OS の始動プロシージャおよびシャットダウン・プロシージャでは、TSO は VTAM に依存します。

SDF ツリー構造定義の詳細については、[239 ページの『ステップ 1: SDF 階層の定義』](#)に説明があります。



## 状況記述子が SDF に及ぼす影響

状況記述子は、SDF が使用する情報の主要単位です。状況記述子の情報によって、SDF 状況ディスプレイが任意の時点でどのように見えるかが判別されます。この項では、SDF が状況記述子の使用方法を説明します。

### 優先順位とカラーの割り当て

状況記述子は、優先順位番号とカラーの両方に割り当てられます。これらのカラーと優先順位の割り当てによって、状況コンポーネントを表示するカラーが決まります。SDF では、低い番号は高い優先順位を表しています。状況記述子は、優先順位の昇順で、状況コンポーネントに結合されています。

状況記述子に必要なカラーと優先順位の割り当ては、以下の 2 か所で定義されます。

- NetView DSIPARM データ・セットの AOFINIT メンバー中の PRIORITY パラメーターの中。このパラメーターは、状況記述子に使用する初期の優先順位とカラーの割り当てを定義します。優先順位およびカラーの割り当てへのそれ以上のカスタマイズを行わないと、AOFINIT に定義された値が使用されます。AOFINIT で使用するデフォルトの優先順位の範囲とカラーは、以下のとおりです。

優先順位範囲	カラー
001 から 199	赤色
200 から 299	ピンク色
300 から 399	黄色
400 から 499	青緑色
500 から 599	緑色
600 から 699	青色

白は、状況記述子のデフォルトのカラー（「*IBM Z System Automation* プログラマーズ・リファレンス」に記載された AOFINIT メンバーの DCOLOR パラメーター）として、およびツリー構造項目を持っていない状況コンポーネントのデフォルトのカラー（「*IBM Z System Automation* プログラマーズ・リファレンス」に記載された AOFINIT メンバーの ERRCOLOR パラメーター）として使用されます。PRIORITY パラメーターについて詳しくは、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照してください。

- 状況詳細ポリシー・オブジェクトの SDF 定義の中。これらの項目は、特定のリソース状況に使用するカラー、強調表示、および優先順位を定義します。カスタマイズ・ダイアログで定義するカラーと優先順位の割り当ては、AOFINIT メンバーの割り当てをオーバーライドするために使用することができます。

**注：**SDF ディスプレイに表示されるリソース状況の一部は、自動化状況ファイルで使用されるリソース状況と直接は対応していません。

「*IBM Z System Automation* プログラマーズ・リファレンス」は、SA z/OS によって提供されるデフォルト・リソース状況タイプ、カラー、強調表示、および優先順位を示します。これらの設定値は、SA z/OS に対して、状況記述子を SDF に定義するときに使用されるパラメーターを定義します。

SDF 状況詳細定義について詳しくは、244 ページの『[ステップ 4: \(オプション\) カスタマイズ・ダイアログでの SDF の定義](#)』を参照してください。

### 状況記述子の状況コンポーネントへのチェーニング

リソース状況変化によって、状況記述子が生成されます。SDF は、この状況記述子を状況記述子のチェーンに追加します。チェーニングされた状況記述子は、状況コンポーネントの状況とカラーを判別します。チェーンの中で最高優先順位の状況記述子は、状況コンポーネントを表示する初期カラーを決めます。基礎となるチェーニングされた優先順位番号は、連続した詳細状況表示を表示するカラーを決めます。

状況記述子は、ツリー構造の状況コンポーネントの各レベルとのチェーンが切り離されます。低レベル状況コンポーネントにチェーニングされた状況記述子は、高レベルの状況コンポーネントにも、再び優先順位の順序でチェーニングされます。また、状況記述子は、ルート・コンポーネントからもチェーンが切り離されます。これらの状況記述子は、現在、ツリー構造のすべてのレベルで存在しているすべての状況記述子です。

例えば、233 ページの図 50 は、現在、システム SY1 用に生成されている状況記述子を示しています。各状況記述子の優先順位は、番号で示されます。

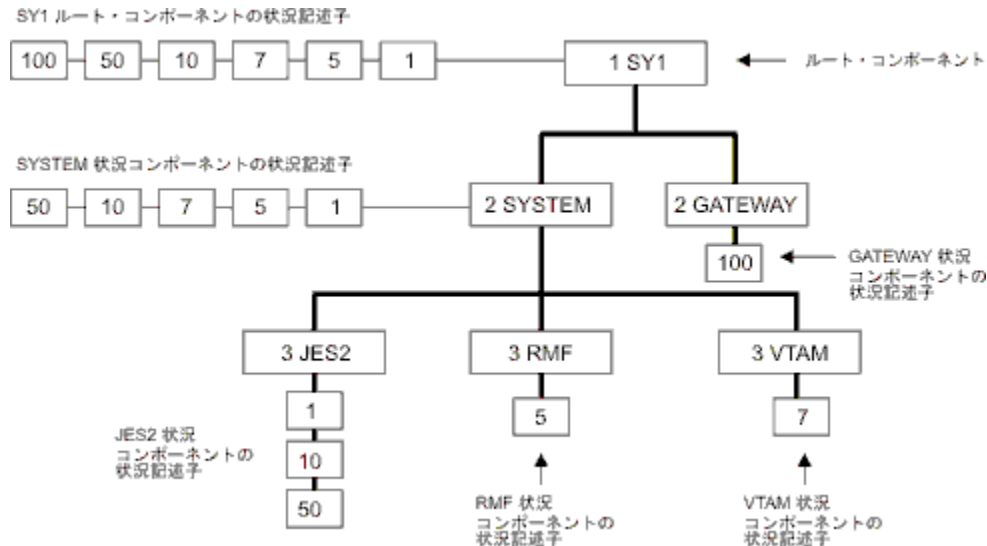


図 50. 状況コンポーネントにチェーニングされた状況記述子

このツリー構造の最下位レベルの状況コンポーネント JES2、RMF、および VTAM には、チェーンが切り離された状況記述子があります。状況コンポーネント JES2 には、優先順位が 1、10、および 50 のチェーニングされた状況記述子が 3 つあります。1 は最高の優先順位であるため、優先順位 1 を持つ状況記述子は、チェーン内で最初に編成されます。この最高優先順位の状況記述子は、状況パネルに JES2 を表示するカラーを決定します。オペレーターが「detail」PF キーを使用して JES2 についての詳細状況ディスプレイを表示すると、優先順位 1 を持つ状況記述子に含まれている情報が最初に表示され、次に優先順位 10 を持つ状況記述子についての詳細状況ディスプレイが表示されます。それ以降は同様です。

ツリー構造の SYSTEM 状況コンポーネント・レベルでは、低レベルの状況コンポーネントからの状況すべてもチェーニングされます。RMF および VTAM にチェーニングされた状況記述子は、JES2 の状況記述子の優先順位 10 および 50 より高い優先順位をもっているため、そのチェーン中の優先順位 1 の状況記述子の後ろに編成されます。SYSTEM レベルで「detail」PF キーを使用するオペレーターには、優先順位 1 から優先順位 50 までの範囲の 5 つの詳細状況ディスプレイが表示されます。

同様に、ツリー構造中の SY1 レベルでは、ツリー構造の状況コンポーネントすべてにチェーニングされたすべての状況記述子が、優先順位の順序でチェーニングされます。SY1 レベルで「detail」PF キーを使用するオペレーターには、優先順位 1 から優先順位 100 までの範囲の 6 つの詳細状況ディスプレイが表示されます。

状況コンポーネントに、等しい優先順位を持つ複数の状況記述子があると、その状況記述子は、到着時刻の順序で状況コンポーネントのチェーンから切り離されます。

状況記述子が、もはやリソースの実際の状況を正確に反映していないと、SDF は自動的に状況記述子チェーンからその状況記述子を削除します。優先順位が状況記述子の順序をどのように決定するか例として、現在、状況コンポーネント JES2 に 2 つの状況記述子があるものとします。JES2 の状況記述子が 2 つあり、優先順位 120 と 140 をもっている場合、優先順位 120 を持つ状況記述子が最初に表示されます。いずれの場合も、JES は SDF 状況パネルに赤で表示されます。

SA z/OS では、すべての状況は自動化制御ファイルに定義されます。自動化イベントが発生すると、SA z/OS AOCUPDT コマンドが、自動化制御ファイルを走査してその状況の SDF 項目があるか調べます。SA z/OS は、自動化制御ファイルからの情報を使用して、状況を追加する要求を出します。

例えば、230 ページの図 48 の SDF パネル例に示されているサブシステム RMF が STOPPING 状態に設定されているものとします。SA z/OS AOCUPDT コマンドは、自動化制御ファイルを走査して SDF の STOPPING 状態項目を調べ、優先順位 420 を指定して状況記述子を生成します。SDF は、状況記述子を RMF 状況コンポーネントに追加します。RMF は状況表示パネル上に反転した黄色で示されます。RMF が STOPPED 状態になると、AOCUPDT コマンドは自動化制御ファイルを走査して STOPPED 状態の SDF 項目を調べ、優先順位 120 を持つ状況記述子を生成します。SDF は、この新しい状況記述子を RMF 状況コンポーネントに追加します。これで、RMF は SDF 状況表示パネル上に赤で表示されます。

### 状況記述子のツリー構造の上下への伝搬

ツリー構造に定義されている依存要素の順序に基づいて、状況記述子は、ツリー構造中の状況コンポーネントの上方または下方に伝搬することができます。

状況記述子のこの伝搬は、オペレーターが、特定の状況コンポーネントに対して「detail」PFキーを使用して表示できる詳細状況表示の他に、状況コンポーネントを表示するカラーに影響を与えます。

ツリー構造の上方または下方への状況の伝搬は、AOFINIT メンバー中の PROPUP および PROPDOWN パラメーターによって定義されます (詳細については、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください)。

AOFINIT メンバーにある状況伝搬の SA z/OS 提供のデフォルトは、状況を上方に伝搬します (PROPUP=YES) が、下方には伝搬しません (PROPDOWN=NO)。

状況がツリー構造中で上方に伝搬されると、状況記述子がツリー構造の下位レベルで追加または削除された場合、その状況記述子は、ツリー構造の高レベル・ノードの状況記述子の累積チェーンからも追加または削除されます。

状況がツリー構造中で上方に伝搬されると、モニターされるリソースすべての状況がルート・ノードで統合整理されます。この方法で、ルート・ノードのカラーは、コンピューター・オペレーション・センター内で最も重要またはクリティカルな状況を反映します。例えば、[231 ページの図 49](#) では、SDF がツリー構造で状況変化を上方に伝搬すると、AOFSSI のすべてのカラー変更が AOFAPPL、APPLIC、SYSTEM、および SY1 に反映されます。[230 ページの図 48](#) で、モニターされるリソースすべてが緑になっていると、「データ・センター・システム (Data Center Systems)」パネル上のルート・ノード CHI01 も緑で表示されます。

状況がツリー構造中で下方に伝搬されると、状況変化がツリー構造の上位レベルで発生した場合に、この変化はツリー構造中の下方に送信されます。この下方への伝搬により、ツリー構造中の下位レベルの状況記述子が追加または削除されることがあります。

システム全体がダウンしたときに、状況の下方への伝搬が役立つことがあります。このような場合、ユーザーは SDF 状況パネルにシステム状況を正確に反映させたいと考えます。以前に生成された状況記述子 (コンポーネントが UP になっていて、実行中であることを示している) を、ツリー構造の下位レベルのコンポーネントに保存したいとは思いません。なぜなら、これらの状況記述子はコンポーネントの状況を正確に反映していないからです。状況を下方へ伝搬し、ツリー構造中のすべての状況コンポーネントからすべての状況記述子を除去するように SDF の導入で構成できます。オペレーターが、ツリー構造中の下位の状況コンポーネントのいずれかについての詳細状況の表示を試みると、「NO DETAIL INFO AVAILABLE」というメッセージを受信します。メンバー AOFINIT の EMPTYCOLOR パラメーターにより、デフォルト・カラーの青で定義した空チェーン・カラーも、詳細情報がないことを示すのに使用されます。EMPTYCOLOR の説明については、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

### SDF による特定の問題への取り組み支援

SDF 構造および処理によって、問題を識別するプログラムは、特定の問題だけに焦点を当てることができます。

例えば、アプリケーション・プログラムが CHI01 における状況コンポーネント JES の警告メッセージを検出するものとします。以下の処理ステップが行われます。

1. アプリケーション・プログラムは、要求を SDF に出して JES の状況記述子を追加します。
2. ここで、システム CHI01 における JES の状況項目は、JES に関する問題があることを示します。SDF が、階層ツリー構造の情報へ状況を伝搬するように構成されていると、システム CHI01 の状況にも、問題プログラム状態が反映されます。PROPUP SDF 初期設定パラメーターについて詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。
3. ここで、別のさらに重大な問題が起こるものとします。この新しい問題を検出したアプリケーション・プログラムは、最初の問題の状況記述子より低い優先順位番号を持つ状況記述子を追加する別の要求を SDF に出します。
4. 状況記述子は、優先順位の順序でチェーンングされるので、これで、JES 状況は、さらに重大な問題の状況記述子のカラーを反映します。
5. 重大な方の問題が解決されると、問題の解決を検出したアプリケーション・プログラムは、この問題の状況記述子を JES 状況記述子のチェーンから除去する要求を SDF に出します。



6. 状況パネルは、最初の問題を反映するように更新されます。

## SDF パネルの定義方法

詳細状況表示パネルを除くすべての SDF 状況パネルは、NetView DSIPARM データ・セットの AOFPNLS メンバーの中で定義します。

メンバー AOFPNLS には、次の一方または両方を入れることができます。

- パネルの定義を含む他の NetView DSIPARM メンバーを参照する %INCLUDE ステートメント。  
%INCLUDE ステートメントによって、指定されたパネル定義メンバーがロードされます。これは、SA z/OS が提供する AOFPNLS のバージョンで使用されている方式で、この方式をお勧めします。  
%INCLUDE ステートメントでは、システム・シンボルを使用できます。これにより、カスタマイズ作業とエラーの両方を削減することができます。
- すべての SDF パネルのパネル構造定義。

AOFPNLS で定義または参照されたパネル・メンバーは、システム・メモリーにロードされ、SDFPANEL コマンドを使用して削除、置換、または一時的な常駐化を行うことができます(コマンドの説明については、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照してください)。

必要に応じて動的にロードするパネル ([235 ページの『ツリー構造メンバーおよびパネル定義メンバーの動的なロード』](#)を参照) は、パネルそのものと同じメンバー名を持つ NetView DSIPARM メンバーに定義する必要があります。

システム・メモリーを節約するために、頻繁に使用されるパネルだけを AOFPNLS に組み込むことをお勧めします。その他のパネルは、SDF 機能キーを押すか、SCREEN コマンドを使用するかのいずれかによって、必要なときに動的にロードすることができます。

注: 動的な最新表示は、AOFPNLS で定義されたパネルでのみ機能します。

SDF は、状況記述子の情報から、詳細状況表示パネルのフォーマット設定と構築を内部的に行います。ユーザーが詳細状況表示パネルの形式設定と構築を行う必要はありません。パネル定義に定義されている状況コンポーネントも、対応するツリー構造に定義する必要があります。しかし、ツリー構造に定義されている状況コンポーネントのすべてに、SDF 状況パネルで対応する項目が必要なわけではありません。例えば、[231 ページの図 49](#) では、APPLIC 状況コンポーネントは、疑似項目にすぎないので、実際には、どの SDF 状況表示パネルにも表示できません。

SDF 状況パネルは、任意の環境を反映するようにカスタマイズすることができます。例えば、コンピューター・オペレーション・センターのすべてのプロセッサのすべての JES サブシステムの状況を表示するパネルを定義することができます。JES オペレーターは、複合システム内のすべての JES サブシステムの状況を判別するパネルを表示することができます。

SDF パネルの定義の詳細については、[240 ページの『ステップ 2: SDF パネルの定義』](#)を参照してください。

## ツリー構造メンバーおよびパネル定義メンバーの動的なロード

主要な SDF ツリー構造およびパネル定義メンバーで %INCLUDE ステートメントを使用すると、SDF を再始動することなく、ツリー構造およびパネル定義メンバーを動的にロードできます。

SDFTREE コマンドは、ツリー構造定義メンバーをロードします。SDFPANEL コマンドは、パネル定義メンバーをロードします。メンバー AOFTREE および AOFPNLS そのものを動的にロードし直すことができます。*IBM Z System Automation* プログラマーズ・リファレンスを参照。

RESYNC SDFDEFS コマンドは、適用する SDF ルート名に拡張自動化オプション (AAO) AOF\_AAO\_SDFROOT.n を使用して SDF パネルを生成します。(「*IBM Z System Automation* オペレーター・コマンド」を参照)。また、&SDFCxxx で始まる変数が定義された SDF ツリー定義を、対応する AAO AOF\_AAO\_SDFCxxx.n を使用して複製します。構成リフレッシュ・モニターでは、予約された AAO AOF\_AAO\_SDFCsaplex.n が使用されます。

注: メンバー AOFPNLS と AOFTREE にはいずれも、複製の対象である変数が含まれてはなりません

AOF\_AAO\_SDFROOT.n 変数のセットアップについては、「*IBM Z System Automation* プログラマーズ・リファレンス」の『状況コンポーネント・パネル定義』を参照してください。

AOF\_AAO\_SDFCxxx.n 変数のセットアップおよび使用については、「IBM Z System Automation プログラマーズ・リファレンス」の『AOF TREE』を参照してください。

## 複数システムでの SDF の使用

自動化ネットワーク内の複数システムが、フォーカル・ポイント・システム上の SDF にそのリソース状況の情報を転送できるように、SDF を構成することができます。

複数システム環境では、以下の要件が満たされていることを確認してください。

- それぞれのシステムごとのツリー構造を、フォーカル・ポイント・システムの SDF 上の NetView DSIPARM の AOF TREE メンバーで定義する必要があります。ルート名は、システム・ツリー構造ごとに固有でなければなりません。
- ターゲット・システム状況の更新がフォーカル・ポイント SDF に転送される前に、SA z/OS フォーカル・ポイント・サービスが実装済みであることが必要です。

各ルート名は複数システム環境内で固有でなければならないため、フォーカル・ポイント SDF に定義されているシステム上の状況コンポーネントはすべて、その名前の前にルート・コンポーネント名を付けることによって、一意的に扱うことができます。

```
ROOT_COMPONENT.STATUS_COMPONENT
```

例:

```
SY1.JES2
```

同様に、ターゲット・システムからフォーカル・ポイント SDF に転送されるすべての SDF 状況記述子の先頭には、SA z/OS ルーチンによって、ターゲット・システムのルート名が付けられます。

### 1 次フォーカル・ポイントとバックアップ・フォーカル・ポイント (APAR OA55386)

SDF は、1 つの 1 次フォーカル・ポイント (PFP) と 1 つのバックアップ・フォーカル・ポイント (BFP) をサポートします。PFP が使用可能な場合、PFP と BFP の両方が定義されたターゲット・システムは、その SDF データを PFP システムに転送します。ターゲット・システムは、PFP システムが使用不可であることを検出すると、そのフォーカル・ポイントを BFP に切り替え、すべての SDF データの BFP システムへの送信を即時に開始します。ターゲット・システムは、PFP が再び使用可能になったことを検出するとすぐに、そのフォーカル・ポイントを PFP システムに戻し、BFP システム上のすべての SDF データを削除し、すべての SDF データの PFP システムへの送信を開始します。

同じ PFP および BFP を定義する各ターゲット・システムは、各フォーカル・ポイント・システムの状況をモニターする必要があります。各フォーカル・ポイント・システムに表示されるデータは、各ターゲット・システムとの個別の接続によって異なります。

また、SDF は、ターゲット・システムと PFP 間の 1 つの接続の可用性に依存しないように、両方のフォーカル・ポイントの並行更新もサポートします。

フォーカル・ポイント・システムがローカル・データを他のフォーカル・ポイント・システムに転送してはいけない場合は、ターゲット・システムのみが、カスタマイズ・ダイアログで Network (NTW) 項目タイプの SDF FOCALPOINT ポリシー内の「**並行更新 (Parallel Update)**」フィールドに YES を指定する指定する必要があります。各フォーカル・ポイント・システムがローカル・データを他のフォーカル・ポイント・システムに転送する必要がある場合は、すべてのシステムが 1 つの定義を共有する必要があります。これは、PFP システムがそれ自体を 1 次フォーカル・ポイントとして定義し、BFP システムをバックアップ・フォーカル・ポイントとして定義することを意味します。また、BFP システムはそれ自体をバックアップ・フォーカル・ポイントとして定義し、PFP システムを 1 次フォーカル・ポイントとして定義します。

ローカル・システム名以外の SDF ルート名を定義した場合は、**RESYNC FP** コマンドを使用して、それらを同期できるようにする必要があります。PFP が SY1P という名前で、追加ルート名 PFPONLY を持ち、BFP が SY2B という名前で、類似のルート名 BFPONLY を持つとした場合、それらの定義方法は、以下のスタイル・シート定義のようになります。

```
%>select
%>  when cursys() = 'SY1P' then
%>  do
```

```

/* PFP has defined PFP=SY1P and BFP=SY2B */
COMMON.AOF_AAO_SDFPFP_ROOT.0 = 1
COMMON.AOF_AAO_SDFPFP_ROOT.1 = PFPONLY
%> end
%> when cursys() = 'SY2B' then
%> do
COMMON.AOF_AAO_SDFBFP_ROOT.0 = 1
COMMON.AOF_AAO_SDFBFP_ROOT.1 = BFPONLY
%> end
%> otherwise
%> nop
%>end /* select */

```

**INGAMS REFRESH** コマンドを使用してフォーカル・ポイントの並行更新を活動化および非活動化する予定の場合は、上記のスタイル・シート定義を以下のように調整する必要があります。

```

%>select
%> when cursys() = 'SY1P' then
%> do
/* PFP has defined PFP=SY1P and BFP=SY2B */
COMMON.AOF_AAO_SDFPFP_ROOT.0 = 1
COMMON.AOF_AAO_SDFPFP_ROOT.1 = PFPONLY
COMMON.AOF_AAO_SDFBFP_ROOT.0 = 1
COMMON.AOF_AAO_SDFBFP_ROOT.1 = BFPONLY
%> end
%> when cursys() = 'SY2B' then
%> do
COMMON.AOF_AAO_SDFBFP_ROOT.0 = 1
COMMON.AOF_AAO_SDFBFP_ROOT.1 = BFPONLY
%> end
%> otherwise
%> nop
%>end /* select */

```

これらの定義により、フォーカル・ポイント処理は、並行更新オプションが非活動化されたときにバックアップ・フォーカル・ポイントで削除してはならない SDF ルート名を判別できます。

スタイル・シート変数の説明については、204 ページの『読み取り/書き込み変数』を参照してください。

## SDF コンポーネント

SDF は、以下のコンポーネントから成っています。

表 28. SDF コンポーネント		
名前	タイプ	目的
AOFTDDF	タスク	SDF を初期化し、状況データベースを保守します。この初期化は、自動化機能です。
SDF	コマンド	SDF オペレーター・セッションを開始します。
SDFTREE	コマンド	NetView DSIPARM データ・セットから SDF ツリー構造定義メンバーを動的にロードまたは削除します。
SDFPANEL	コマンド	NetView DSIPARM データ・セットから SDF パネル定義メンバーを動的にロードまたは削除します。
AOFINIT	入力ファイル	「 <i>IBM Z System Automation</i> プログラマーズ・リファレンス」に説明されているステートメントによって定義されている SDF の初期設定パラメーターが入ります。AOFINIT は NetView DSIPARM データ・セットに入っています。
AOFTREE	入力ファイル	「 <i>IBM Z System Automation</i> プログラマーズ・リファレンス」に説明されているツリー構造が入ります。通常、このメンバーは、ツリー構造が入っている他のメンバーを参照する %INCLUDE ステートメントのリストから構成されます。AOFTREE は、NetView DSIPARM データ・セットに入っています。

表 28. SDF コンポーネント (続き)		
名前	タイプ	目的
AOFPNLS	入力ファイル	240 ページの『ステップ 2: SDF パネルの定義』に説明されているステートメントによって定義されている SDF パネル・パラメーターが入ります。通常、このメンバーは、パネル定義が入っている他のメンバーを参照する %INCLUDE ステートメントのリストから構成されます。AOFPNLS は、NetView DSIPARM データ・セットに入っています。
panel_name	入力ファイル	1 つ以上の SDF パネルの定義、または他の DSIPARM パネル定義メンバーを示す %INCLUDE ステートメントが入っている DSIPARM メンバー。パネル定義メンバーには、メンバーと同じ名前を持つ単一のパネル定義を入れるように強くお勧めします。
tree_name	入力ファイル	1 つ以上のツリー構造の定義が入っている DSIPARM メンバー。ツリー定義メンバーには、メンバー名と同じルート・コンポーネント名を持つ単一ツリーの定義を入れるように強くお勧めします。

## SDF タスクの始動と停止の方法

SA z/OS の初期化の際に、AOFTDDF タスクは、パネル形式、パネル・フロー、および ツリー構造を定義しているメンバーをロードします。

メンバー AOFINIT は、すべての SDF パネルに共通するパラメーターと、基本的な初期設定の指定 (画面サイズ、デフォルトの PF キー、および SDF セッションを開始すると表示される初期画面など) を定義します。これらの AOFINIT パラメーターについては、「*IBM Z System Automation プログラマーズ・リファレンス*」で説明されています。

### SDF タスクの開始

SA z/OS コードで、AOFTDDF タスクは、次のコマンドで開始します。

```
START TASK=AOFTDDF
```

### SDF タスクの停止

SA z/OS コードで、AOFTDDF タスクは、次のコマンドで停止します。

```
STOP TASK=AOFTDDF
```

注: 既存の SDF 状況記述子はメモリー内にしか保持されていないので、SDF を再開すると、すべての SDF 状況記述子は失われます。

## SDF の定義プロセス

以下の手順を使用して、SDF セッションに表示されるパネルを定義します。各ステップの詳細は、本章の後の方と、「*IBM Z System Automation プログラマーズ・リファレンス*」にあります。

### 始める前に

SDF 環境の定義に使用される AOFINIT、AOFREE、AOFPNLS などのメンバーはすべて、NetView の Data REXX 機能をサポートします。この機能をソース・メンバーで使用するには、最初のレコードの最初の列に「/\*%DATA」ディレクティブを指定する必要があります (まだ行っていない場合)。

### 手順

1. NetView DSIPARM データ・セット・メンバーでツリー構造ステートメントを使用して、SDF パネルで使用するモニター・リソースの階層を定義します。これらのツリー構造定義メンバーは、NetView DSIPARM データ・セットの中のメイン SDF ツリー構造定義メンバーで、%INCLUDE ステートメントに



よって参照してください。詳しくは、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照してください。

2. NetView DSIPARM データ・セット・メンバーで、パネル定義ステートメントを使用して、SDF 状況パネルを定義します。パネルは、SDF が開始されたときに自動的にロードすることも、SDFPANEL コマンドを使用して動的にロードすることもできます。パネルを自動的にロードする場合は、パネル定義メンバーを指定した %INCLUDE ステートメントを、(conref) DSIPARM データ・セットのメイン・パネル定義メンバー AOFPNLS に追加します。詳しくは、240 ページの『ステップ 2: SDF パネルの定義』を参照してください。

一般的には以下の順序で、SDF 状況パネルの定義とカスタマイズを行います。

- a. ルート・パネル
  - b. ルート・パネル上の各項目の状況コンポーネント・パネル
  - c. その他のすべてのカスタマイズされた状況パネル
3. 必要に応じて (オプション)、NetView DSIPARM メンバー AOFINIT 中の SDF 初期設定パラメーターをカスタマイズするか、デフォルトを使用します。SDF 初期設定パラメーターの詳細については、「*IBM Z System Automation* プログラマーズ・リファレンス」を参照してください。デフォルトを使用することをお勧めします。
  4. カスタマイズ・ダイアログを使用して、SDF リソースの状況、カラー、強調表示、および優先順位の値を定義して、SDF の状況表示 ポリシー・オブジェクトを編集するか、デフォルトを使用します。このステップはオプションです。状況表示ポリシー・オブジェクトの説明については、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。デフォルトを使用することをお勧めします。

注:

- a. SA z/OS が現在自動化していないリソースは、SDF パネルには表示されません。
- b. 複数のシステムの状況を表示し、ターゲット・システムからフォーカル・ポイント・システム上の SDF に状況を転送するには、SA z/OS フォーカル・ポイント・サービスが既に組み込まれている必要があります。フォーカル・ポイント・サービスの構成の詳細については、「*IBM Z System Automation* 自動化ポリシーの定義」を参照してください。

## ステップ 1: SDF 階層の定義

NetView DSIPARM データ・セット内のメンバー AOFTREE には、状況カラーの変更についての伝搬階層を定義する定義の集合が入ります。

コンポーネントの状況が変化すると、対応するカラーの変更がツリーの上または下 (すなわち、次に高いレベルのコンポーネントまたは次に低いレベルのコンポーネント) に伝搬されます。レベルは、各コンポーネントに割り当てられたレベル番号によって決定されます。伝搬のタイプは、AOFINIT メンバーの中の項目、または状況記述子を状況コンポーネントに追加する個別要求のいずれかによって決められます。

注: SA z/OS は、サブシステムのシャットダウン・プロシージャー または始動プロシージャーに、この SDF 階層を使用しません。その代わりに SA z/OS は、自動化ポリシーで定義されているサブシステム項目を使用して、始動とシャットダウンの関係と階層を判断します。

### ツリー構造の定義

AOFTREE には、ツリー構造の定義が入ります。ツリー構造を定義するには、以下の方法が可能です。

- 特定のツリー構造の定義が入っている他のメンバーを参照する %INCLUDE ステートメントを使用する。これは、SA z/OS が提供する AOFTREE のバージョンで使用されている方式で、この方式をお勧めします。
- %INCLUDE ステートメントでは、参照メンバー名は括弧で囲む必要があります。
- すべてのツリー構造定義を AOFTREE に入れる。
- 両方の組み合わせを使用する。

システム・シンボルは、AOFTREE、AOFINIT および AOFPNLS メンバー内で使用される場合は、サポートされます。これにより、カスタマイズ作業とエラーの両方を削減することができます。

240 ページの図 51 は、代表的なツリー構造の定義の例を示したものです。

```

1 SY1
  2 SYSTEM
    3 WTOR
    3 APPLIC
      4 AOFAPPL
        5 AOFSSI
      4 JES
      4 VTAM
    3 TSO
    3 RMF
  2 GATEWAY
  2 MONITOR
  2 APG
  3 GROUPS

```

図 51. ツリー構造の定義の例

このツリー構造では、SY1 がルート・コンポーネントです。この定義は、SY1 という名前の別のメンバーに入っています。これは、AOF TREE メンバーの中の次のステートメントによって参照されます。

```
%INCLUDE(SY1TREE)
```

### ツリー構造のロード

初期化時に、すべてのツリー構造をロードする必要はありません。一部は、SDF が開始された後に動的にロードすることができます。これを行うには、AOF TREE を使用して、初期化時にロードするツリー構造の項目を定義してから、SDF TREE コマンドを使用して必要に応じてさらにツリー構造をロードします。

詳しくは、「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください。

SDF が開始された後でロードされたツリー構造は、別のメンバーに入れておく必要があります。各メンバーは、ツリー構造を定義するルート・コンポーネントの後に名前を指定する必要があります。

## ステップ 2: SDF パネルの定義

SDF 状況パネルは、NetView DSIPARM メンバー AOF PNLS の中に定義されます。

SDF が初期設定されると、SA z/OS は、AOF PNLS の中のパネル定義をロードします。

### パネル定義の方式

AOF PNLS にパネルを定義するには、以下の方法が可能です。

- パネル定義が入っている別の NetView DSIPARM メンバーを参照する %INCLUDE ステートメントを使用する。これは、SA z/OS が提供する AOF PNLS のバージョンで使用されている方式で、この方式をお勧めします。SDF パネル定義メンバーに %INCLUDE ステートメントを使用する際の詳細については、243 ページの『[SDF パネルの %INCLUDE ステートメント](#)』を参照してください。
- すべてのパネルの実際の定義を組み込む。
- %INCLUDE ステートメントとパネル定義の両方の組み合わせを使用する。
- 必要なときだけ追加のパネル定義をロードできるように、初期設定時にロードするパネル項目のサブセットを組み込む（「*IBM Z System Automation プログラマーズ・リファレンス*」を参照してください）。

システム・シンボルは、AOF TREE、AOF INIT および AOF PNLS メンバー内で使用される場合は、サポートされます。これにより、カスタマイズ作業とエラーの両方を削減することができます。

### パネル定義の構造

各パネル定義の構造は、以下のとおりです。

- パネル定義ステートメントの始まり (PANEL)
- 以下のステートメントのペアから成る状況コンポーネント定義ステートメント。
  - STATUSFIELD: パネル上の状況コンポーネントの位置を定義します。
  - STATUSTEXT: STATUSFIELD に表示されるテキストを定義します。
- 以下のステートメントのペアから成るテキスト・フィールドおよびデータ定義ステートメント
  - TEXTFIELD: パネル上の固定フィールドの位置と属性を定義します。

- TEXTTEXT: TEXTFIELD に表示されるテキストを定義します。

- 状況パネル PF キー定義 (PFKnn)

SDFCONF コマンドを PF4 キーに割り当ててください。以下の定義を使用します。

```
PFK4=SDFCONF &ROOT,&COMPAPPL,&RV,&SID,&SNODE,&DATE,&TIME,&DA
```

SDFCONF を使用して SDF 内のレコードを削除すると、実際に削除を実行する前に確認を促すプロンプトが出されるため、便利です。プロンプト・パネルが表示されないようにするには、",VERIFY=NO" を SDFCONF コマンドの末尾に付けてください。

例外メッセージ、すなわち、取り込みメッセージで重大度が Unusual、Important、および Critical のものを削除するには、SDFCONF を呼び出す必要があります。SDFCONF コマンドは、SDF 制御構造からメッセージ項目を削除し、また、そのメッセージが表示された他のすべてのインターフェース (例えば TEP) からメッセージ項目を削除します。

- パネル・ステートメントの終わり (ENDPANEL)

これらのパネル定義ステートメントの説明は、「*IBM Z System Automation プログラマーズ・リファレンス*」にあります。

### パネル定義の推奨する順序

パネルを定義するには、以下の順序でパネルを定義することをお勧めします。

1. ルート・パネル
2. ルート・パネルにリストされる各項目の状況コンポーネント
3. その他のすべてのカスタマイズされた状況パネル

注: この定義パネルの順序は、推奨事項にすぎません。SDF パネルは、どのような順序でも定義することができます。

### パネル定義の例

241 ページの [図 52](#) は、SDF パネルの表示例を示しています。

```
SYSTEM                DATA CENTER SYSTEMS

SY1                    GATEWAY

===>
1=HELP 2=DETAIL 3=RET 6=ROLL 7=UP 8=DN 10=LF 11=RT 12=TOP
```

図 52. SDF パネルの例

242 ページの [図 53](#) は、[241 ページの図 52](#) に示されているパネルを定義するのに必要なパネル定義ステートメントを示しています。

```

PANEL (SYSTEM, 24, 80)
TEXTFIELD(01,02,10,WHITE,NORMAL)
TEXTTEXT (SYSTEM)
TF (01, 25, 57, WHITE, NORMAL)
TT (DATA CENTER SYSTEMS)
STATUSFIELD(SY1,04,04,11,N,,SY1SYS)
STATUSTEXT (SY1)
SF (SY1.GATEWAY,02,40,47,N,,GATEWAY)
ST (GATEWAY)
TF (24,01,79,T,NORMAL)
TT (1=HELP 2=DETAIL 3=RET 6=ROLL 7=UP 8=DN
10=LF 11=RT 12=TOP)
PFK1(AOCHELP SDF)
PFK2(DETAIL)
PFK3(RETURN)
PFK6(ROLL)
PFK7(UP)
PFK8(DOWN)
PFK10(LEFT)
PFK11(RIGHT)
PFK12(TOP)
ENDPANEL

```

図 53. パネル定義項目の例

242 ページの図 53 では、パネル名は SYSTEM です。このパネル定義は、AOFPNLS 中の %INCLUDE ステートメントによって参照される別のメンバーに入れることも、AOFPNLS に直接コーディングすることもできます。推奨する方式は、別のメンバーと %INCLUDE ステートメントを使用する方法です。別のメンバーに入っている場合、メンバー名は SYSTEM です。パネルのすべての PF キーを明示的に定義する必要はありません。指定されない PF キーの定義は、NetView DSIPARM メンバー AOFINIT 中の定義から取り出されます。

242 ページの表 29 は、242 ページの図 53 の各ステートメントの説明です。

表 29. パネル定義項目の説明	
ステートメント	説明と値の例
PANEL (SYSTEM, 24, 80)	パネル定義ステートメント。パネル名は SYSTEM、パネルの長さは 24、パネルの幅は 80 です。
TEXTFIELD(01,02,10,WHITE,NORMAL)	定数パネル・フィールドを定義しているテキスト位置ステートメント。このフィールドは、行 01 の 02 桁目から始まり、10 桁目で終わります。フィールドのカラーは白で、強調表示は通常表示です。
TEXTTEXT (SYSTEM)	定義したばかりのテキスト・フィールドに入る実際のデータを指定するテキスト・データ・ステートメント。このフィールドには、ワード SYSTEM が入ります。  TEXTFIELD および TEXTTEXT は、常にペアでグループ化されます。
TF (01, 25, 57, WHITE, NORMAL)	別の固定フィールド用のもう 1 つの TEXTFIELD ステートメント。
TT (DATA CENTER SYSTEMS)	定義したばかりのテキスト用のもう 1 つの TEXTTEXT ステートメント。
STATUSFIELD(SY1,04,04,11,N,,SY1SYS)	状況コンポーネント・フィールドの位置。状況コンポーネントは SY1 です。このフィールドは、行 04 の 04 桁目から始まり、11 桁目で終わります。強調表示レベルは通常表示です。「Down」PF キーを押したときに表示される次のパネルは SY1SYS です。

表 29. パネル定義項目の説明 (続き)	
ステートメント	説明と値の例
STATUSTEXT (SY1)	STATUSFIELD ステートメントで定義したばかりのフィールド名に使用するテキスト・データ。この場合、フィールド名は SY1 です。  STATUSFIELD および STATUSTEXT ステートメントは、ペアでグループ化されます。
SF (SY1.GATEWAY,02,40,47,N,,GATEWAY)	別の STATUSFIELD 定義。
ST (GATEWAY)	別の STATUSTEXT 定義。
TF (24,01,79,T,NORMAL) TT (1=HELP 2=DETAIL 3=RET 6=ROLL 7=UP, 8=DN 10=LF 11=RT 12=TOP)	ここで、TEXTFIELD および TEXTTEXT は、PF キーの定義を表示するのに使用されます。このパネルの場合、これは、AOFINIT で定義されているデフォルトの定義です。デフォルトとは異なる値が必要な場合は、このパネルに固有の PF キーを定義するためのステートメント (DPFKnn) があります。このステートメントの説明については、「 <i>IBM Z System Automation</i> プログラマーズ・リファレンス」を参照してください。
PFK1 (AOHELP SDF) PFK2 (DETAIL) PFK3 (RETURN) PFK6 (ROLL) PFK7 (UP) PFK8 (DOWN) PFK10 (LEFT) PFK11 (RIGHT) PFK12 (TOP)	PF キーの定義ステートメント
ENDPANEL	このパネルの定義の終わりであることを示すパネル・ステートメントの終わり。

### SDF パネルの %INCLUDE ステートメント

SDF の %INCLUDE ステートメントには、以下の機能があります。

- SDF %INCLUDE ステートメントによって、単一メンバーだけではなく、メンバー・リストを指定することができます。リストの中の各メンバー名は、ロードされる DSIPARM メンバーを表します。リストの中の各メンバー名は、コマンドで区切ります。
- SDF %INCLUDE ステートメントは、指定する 1 つ以上のメンバーを 括弧で囲む必要があります。
- SDF %INCLUDE ステートメントには、オプション STATIC または DYNAMIC を指定できます。DYNAMIC を指定すると、AOF\_AAO\_SDFROOT.n 共通グローバル変数で指定したすべてのシステム名について、パネル定義が生成されます (204 ページの表 26 を参照してください)。STATIC がデフォルトです。
- ターゲット DSIPARM メンバーには、完全なパネル定義または追加の %INCLUDE メンバーしか入れられません。パネル定義は、単一メンバー内に入っている必要があり、そのため、共通に定義されたセグメントを使用して作成することはできません。

システム・シンボルは、AOFTREE、AOFINIT および AOFPNLS メンバー内で使用される場合は、サポートされます。これにより、カスタマイズ作業とエラーの両方を削減することができます。

### ステップ 3: (オプション) SDF 初期設定パラメーターのカスタマイズ

メンバー AOFINIT を使用すると、以下のようなすべての SDF パネルと SDF 初期化指定に共通するパラメーターを定義することができます。

- SDF を開始すると表示される初期画面

- オペレーター・ログオンの最大限度
- デフォルトの PF キーの定義
- 詳細状況表示パネルの PF キーの定義
- 詳細状況表示パネルの PF キーの記述
- デフォルトの優先順位およびカラー

これらのパラメーターは、SDF が開始するときの SDF の値を定義します。

システム・シンボルは、AOFTREE、AOFINIT および AOFPNLS メンバー内で使用される場合は、サポートされます。これにより、カスタマイズ作業とエラーの両方を削減することができます。

SDF のカスタマイズのこのステップは、オプションです。これらのパラメーターに、SA z/OS 提供のデフォルトを使用することをお勧めします。

**注:** ユーザー定義の状況は、リサイクルまたはモニター・サイクルを越えて保管されることはありません。これは、サブシステムの状況がユーザー定義の状況から適切な SA z/OS 状況に変更される ことになるという意味です。

#### **ステップ 4: (オプション) カスタマイズ・ダイアログでの SDF の定義**

状況表示ポリシー・オブジェクトの SDF 項目によって、状況とその状況に割り当てる優先順位を定義することができます。

これらの項目は、状況記述子を状況コンポーネントに追加する要求についてのデータを収集するために、SA z/OS コマンドによって使用されます。SDF の状況詳細定義で使用する形式と値については、「*IBM Z System Automation* プログラマーズ・リファレンス」で説明されています。

SDF のカスタマイズのこのステップは、オプションです。SDF についての SA z/OS 提供の定義を使用するようお勧めします。



## 付録 C システム操作と自動リスタート・マネージャーの調整

SA z/OS のシステム操作には、自動リスタート・マネージャーとの調整機能があります。自動リスタート・マネージャー (ARM) は z/OS の基本コンポーネントです。これは、以下のようなときに、指定されたアプリケーションを自動的に再始動するリカバリー機能です。

- アプリケーションが異常終了した。
- アプリケーションを実行するシステムがシスプレックスの一部であり、そのシステムに障害が起きた。この場合、ARM はシスプレックス内の他のシステムでアプリケーションを再始動しようとします。

SA z/OS は ARM と調整して以下のことを行います。

- 特定のアプリケーションの再始動を行う機能を判別する。
- アプリケーションのリカバリー試行における起こりうる重複または競合を回避する。
- シスプレックスで実行中のアプリケーション用の SA z/OS フォールバック機能をユーザーが十分に利用できるようにする。SA z/OS は、アプリケーションがフォールバック・システムに移された後、SA z/OS がそのシステムにインストールされていれば、アプリケーションの自動化を続行します。フォールバック・システムにインストールされていない場合、SA z/OS はアプリケーションがその 1 次システム以外のシステムでアクティブであることをまだ認識するため、アプリケーションを再始動しようとしません。

「z/OS MVS シスプレックスのセットアップ」で説明されている ARM ポリシー・データ (IXCMIAPU) について、管理データ・ユーティリティーを使用して自動リスタート・マネージャー・ポリシーを定義する必要があります。

SA z/OS は自動リスタート・マネージャーの状況を解決して SA z/OS の状況とし、自動リスタート・マネージャーの関連条件を取り入れて、自動リスタート・マネージャーに関連した以下の 1 つの状況を提供します。

- EXTSTART - アプリケーションは外部から始動または再始動されています。

### ARM エlement名の定義

自動リスタート・マネージャーは Element 名を使用して、それが動作するアプリケーションを識別します。自動リスタート・マネージャーが有効になっているアプリケーションは、それぞれ、自動リスタート・マネージャーとのすべての通信で使用する、アプリケーション自身の固有の Element 名を持っている必要があります。

自動リスタート・マネージャーは Element 名をトラッキングし、Element 名によってそのポリシーを定義させます。アプリケーションは、システム間で移動する場合、元のシステムで使用していたものと同じ Element 名を使用し続ける必要があります。SA z/OS への自動リスタート・マネージャー名の定義についての詳細は、「IBM Z System Automation 自動化ポリシーの定義」の『アプリケーション項目タイプ』を参照してください。

最初はすべての自動リスタート・マネージャー・Element が未登録です。状況から他の状況への移行は、以下によって発生します。

- IXCARM マクロ呼び出し
- アプリケーション障害
- システム障害
- タイムアウト

マイナー・リソース定義 *subsystem.OARM* は、ARM の再始動処理の際に自動化動作を調整するために使用できます。例として、ARM 再始動中にユーザー提供出口を駆動するために RESTART EXIT を有効にして、*subsystem.OARM* マイナー・リソースが指定できるとします。このユーザー出口は、サブシステムの ARM 再始動時に取られる追加のアクションを制御するとします。このマイナー・リソースの RESTART フラグが「N」に解決されると、SA z/OS は、ARM がアプリケーションの再始動を試みるのを許可しません。



*subsystem.0*ARM マイナー・リソース定義を使用するのではなく、RESTART EXIT をアプリケーションのメジャー・リソース定義に対して指定することもできます。この場合、出口は、ARM だけでなくすべてのアプリケーション再始動について駆動されます。

SA z/OS が ARM にアプリケーション再始動の試行を許可しないその他の理由には、以下のものがあります。

- アプリケーションのモニターに、アドレス・スペースが既にアクティブであることが示されている。
- アプリケーションがシャットダウンを行っている。
- アプリケーションが、状況 BREAKING、BROKEN、または CTLDOWN のいずれかである。

## 自動リスタート・マネージャーの MOVE グループの定義

---

同じ ARM エlement名を持つすべてのリソースは、性質が MOVE (MOVE グループ) である 1 つのシスプレックス・アプリケーション・グループにリンクされている必要があります。

アプリケーションの ARM エlement名は、アプリケーションの「Define New Entry」パネルでの作成時か、またはポリシー項目 APPLICATION INFO による作成後に定義されますが、いずれの場合も「**MVS 自動再始動管理Element名 (MVS Automatic Restart Management Element Name)**」フィールドで定義されます。

MOVE グループ内のアプリケーションを自動化マネージャーが再始動しようとする前に、そのアプリケーションが ARM から完全に登録解除されるようにするために、MOVE グループ内の ARM 化された各アプリケーションについて、MOVE グループをサポートするリソースとして定義したうえで、Prepareavailable/WhenObservedDown (passive) 関係を定義する必要があります。

自動化マネージャーが MOVE グループにリンクしているアプリケーションを確実に始動するようにするには、アプリケーションが HardDown 状況になってはいけません。「IPL 時の始動 (Start On IPL)」オプションは、NO に設定しないでください。

MOVE グループの定義方法について詳しくは、「*IBM Z System Automation* 自動化ポリシーの定義」の『新規アプリケーション・グループの作成』を参照してください。

# 付録 D メッセージの自動化

## 汎用シノニム: AOFMSGSY

この AOFMSGSY NetView 自動化テーブル (AT) フラグメントには、適切に設定する必要がある多くのシノニムが含まれています。これは、その他のフラグメントの環境パラメーターをセットアップするために、大部分のマスター自動化テーブルで使用されます。SA z/OS は、AOFMSGSY メンバーを SINGPARM データ・セットで提供します。

このメンバーは、それぞれのユーザー・システムごとにカスタマイズする必要があります。カスタマイズしたコピーは、そのシステムのドメイン固有のデータ・セットに入れてください。

INGSYNU を使用して、独自のシノニムを作成することができます。このメンバーは AOFMSGSY から自動的に組み込まれます。このメンバーを使用すれば、製品提供の AOFMSGSY メンバーの変更を回避できます。

このテーブル・フラグメントの多くの値が、三重の引用符で囲まれていることに注意してください。これは、シノニムの値が一組の単一引用符に囲まれて入力された値であることを意味します。これは、その値が自動化テーブル変数ではなくリテラルとして取り扱われるようにするために必要です。

シノニム	使用法とデフォルト
%AOFALWAYSACTION%	<p>このシノニムには、SA z/OS がいずれのアクションも起動しない Begin-End ブロック内のメッセージすべてに使用されるアクション・ステートメントが含まれます。</p> <p><b>デフォルト: NULL</b></p> <p>デフォルトでは、アクションは実行されず、メッセージは同じ AT 内でさらなる検索を行いません。</p>
%AOFDOM%	<p>このシノニムには、自動化するシステム上の SA z/OS NetView のドメイン ID が入ります。シノニムは、このマシン上の SA z/OS が別のマシンからのメッセージに対応しないようにメッセージを選別するために使用されます。正しく設定しないと、自動化は失敗します。</p> <p><b>デフォルト: &amp;DOMAIN</b></p> <p>これは、サンプルのいくつかで使用されるデフォルトのドメイン・ネームです。</p>
%AOFSYS%	<p>このシノニムには、システム上の最後の IPL で使用されたシステム名が入ります。これは、このマシン上の SA z/OS が、他のマシンで発生したイベントに対応しないようにメッセージを選別するために使用されます。JES3 グローバルで、または EMCS コンソールを持つシスプレックス内で実行する場合、これは重要になります。正しく設定しないと、自動化は失敗します。</p> <p><b>デフォルト: &amp;SYSNAME</b></p> <p>これは、多くのサンプルで使用されるデフォルトのシステム名です。</p>
%AOFARMPPI%	<p>このシノニムには、SA z/OS から z/OS への PPI インターフェースを実行している NetView 自動タスクの名前を入れる必要があります。これは、コマンドを NetView 自動化テーブルから自動開始タスクにルーティングするのに使用されます。</p> <p><b>デフォルト: AOFARCAT</b></p>

## SA z/OS メッセージ表示: AOFMSGSY

NetView 下での SA z/OS メッセージ (接頭部が AOF、ING、HSA、EVJ、EVE および EVI のメッセージ) の表示は、自動化テーブルによって制御されます。これは、メッセージの表示特性を示す数多くのシノニムとタスク・グローバルを使用します。以下のシノニムは、それぞれのタイプごとにメッセージの表示特性を決めるものです。メッセージの通常表示用のセット (AOFNORMx) と、保留メッセージ用の 2 番目のセット (AOFHOLDx) があります。

シノニム	使用法とデフォルト
%AOFHOLDI%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS 通知 (タイプ I) メッセージに対して行うアクションを定義します。</p> <p><b>デフォルト:</b> HOLD(Y) COLOR(GRE) XHILITE(REV)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージを緑色で反転表示します</li> </ul>
%AOFHOLDA%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS 即時アクション (タイプ A) メッセージに対して行うアクションを定義します。規則として、アクションで HOLD(Y) を指定してください。</p> <p><b>デフォルト:</b> HOLD(Y) COLOR(RED) XHILITE(REV) BEEP(Y)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージを赤で反転表示します</li> <li>• メッセージを表示するときに端末アラームを鳴らします</li> </ul>
%AOFHOLDD%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS デシジョン (タイプ D) メッセージに対して行うアクションを定義します。規則として、アクションで HOLD(Y) を指定してください。</p> <p><b>デフォルト:</b> HOLD(Y) COLOR(WHI) XHILITE(REV) BEEP(Y)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージを白で反転表示します</li> <li>• メッセージを表示するときに端末アラームを鳴らします</li> </ul>
%AOFHOLDE%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS 最終アクション (タイプ E) メッセージに対して行うアクションを定義します。規則として、アクションで HOLD(Y) を指定してください。</p> <p><b>デフォルト:</b> HOLD(Y) COLOR(YEL) XHILITE(REV) BEEP(Y)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージを黄色で反転表示します</li> <li>• メッセージを表示するときに端末アラームを鳴らします</li> </ul>

シノニム	使用法とデフォルト
%AOFHOLDW%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS 待ち状態 (タイプ W) メッセージに対して行うアクションを定義します。規則として、アクションで HOLD(Y) を指定してください。</p> <p><b>デフォルト:</b> HOLD(Y) COLOR(PIN) XHILITE(REV) BEEP(Y)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージをピンクで反転表示します</li> <li>• メッセージを表示するときに端末アラームを鳴らします</li> </ul>
%AOFNORMI%	<p>このシノニムは、NCCF コンソールで保留されていない SA z/OS 通知 (タイプ I) メッセージに対して行うアクションを定義します。規則として、アクションで HOLD(Y) を指定してはいけません。</p> <p><b>デフォルト:</b> COLOR(GRE)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されないようにします</li> <li>• メッセージを緑色で表示します</li> </ul>
%AOFNORMA%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS 即時アクション (タイプ A) メッセージに対して行うアクションを定義します。規則として、アクションで HOLD(Y) を指定してはいけません。</p> <p><b>デフォルト:</b> COLOR(YEL) XHILITE(REV) BEEP(Y)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージを黄色で表示します</li> <li>• メッセージを表示するときに端末アラームを鳴らします</li> </ul>
%AOFNORMD%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS デジジョン (タイプ D) メッセージに対して行うアクションを定義します。これらのメッセージを強制的に保留すると便利な場合があります。</p> <p><b>デフォルト:</b> COLOR(WHI) XHILITE(BLI)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージを明滅する白色で表示します</li> </ul>
%AOFNORME%	<p>このシノニムは、NCCF コンソールで保留されていない SA z/OS 最終アクション (タイプ E) メッセージに対して行うアクションを定義します。規則として、アクションで HOLD(Y) を指定してはいけません。</p> <p><b>デフォルト:</b> COLOR(YEL)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されないようにします</li> <li>• メッセージを黄色で表示します</li> </ul>

シノニム	使用法とデフォルト
%AOFNORMW%	<p>このシノニムは、NCCF コンソールで保留中の SA z/OS 待ち状態 (タイプ W) メッセージに対して行うアクションを定義します。これらのメッセージを強制的に保留すると便利な場合があります。</p> <p><b>デフォルト:</b> HOLD(Y) COLOR(PIN) XHILITE(REV) BEEP(Y)</p> <p>これは、以下のことを行います。</p> <ul style="list-style-type: none"> <li>• メッセージが保留されるようにします</li> <li>• メッセージをピンクで反転表示します</li> <li>• メッセージを表示するときに端末アラームを鳴らします</li> </ul>

## オペレーター・カスケード: AOFMSGSY

シノニムの次の集合は、一連のオペレーター・カスケードを定義します。カスケードは、基本的に、コマンドをルーティングするために、フラグメントの多くで使用される自動化オペレーターのリストです。%CASCADE%が 'AUTMON AUTOBASE AUTINIT1' のシノニムとして定義されていて、EXEC ステートメントに ROUTE (ONE %CASCADE%) を指定してコマンドをルーティングすると、そのコマンドは、ログオン済みのカスケードの中の最初の自動タスクで実行されます。これは、通常タスクが使用できない場合に、ユーザーにバックアップ処理タスクを提供する融通性の高い制御可能な手段を提供します。

シノニム	使用法とデフォルト
%AOFOPINITOPR1%	<p>このカスケードは、コマンドを AUTINIT1 にルーティングするのに使用されます。AUTINIT1 の名前を変更すると、シノニムを変更する必要があります。</p> <p><b>デフォルト:</b> AUTINIT1</p> <p>AUTINIT1 のバックアップはありません。必要なときに AUTO1 に障害が起こると、他の多くのことも同様に失敗する可能性があります。</p>
%AOFOPINITOPR2%	<p>このカスケードは、コマンドを AUTINIT2 にルーティングするのに使用されます。AUTINIT2 の名前を変更すると、このシノニムを変更する必要があります。</p> <p><b>デフォルト:</b> AUTINIT2 AUTINIT1</p> <p>AUTINIT2 がアクティブでない場合、AUTINIT1 は AUTINIT2 の作業を行います。</p>
%AOFOPBASEOPER%	<p>このカスケードは、コマンドを BASEOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。BASEOPER は主にフォールバック・オペレーターとして定義され、直接 BASEOPER にルーティングされる作業はほんのわずかしきありません。</p> <p><b>デフォルト:</b> AUTOBASE AUTINIT1</p> <p>AUTOBASE は SA z/OS が他のサンプルで BASEOPER 用に使用するオペレーター ID です。AUTOBASE がアクティブでない場合、AUTINIT1 が試みられます。</p>
%AOFOPRPCOPER%	<p>このカスケードは XCF 通信管理に使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。</p> <p><b>デフォルト:</b> AUTRPC AUTSYS AUTOBASE AUTINIT1</p>

シノニム	使用法とデフォルト
%AOFOPSYSOPER%	<p>このカスケードは、コマンドを SYSOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。SYSOPER は主にフォールバック・オペレーターとして定義され、直接 SYSOPER にルーティングされる作業はほんのわずかしきありません。</p> <p><b>デフォルト:</b> AUTSYS AUTOBASE AUTINIT1</p> <p>AUTSYS は SA z/OS が他のサンプルで SYSOPER 用に使用するオペレーター ID です。</p>
%AOFOPMSGOPER%	<p>このカスケードは、コマンドを MSGOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。MSGOPER は主にその他のメッセージに応答するように定義されます。</p> <p><b>デフォルト:</b> AUTMSG AUTSYS AUTOBASE AUTINIT1</p> <p>AUTMSG は SA z/OS が他のサンプルで MSGOPER 用に使用するオペレーター ID です。</p>
%AOFOPNETOPER%	<p>このカスケードは、コマンドを NETOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。NETOPER は VTAM 自動化用に定義されます。</p> <p><b>デフォルト:</b> AUTNET1 AUTNET2 AUTSYS AUTOBASE AUTINIT1</p> <p>AUTNET1 および AUTNET2 は SA z/OS が他のサンプルで NETOPER 用に使用するオペレーター ID です。NETOPER は、サンプルでバックアップが定義されている唯一のサンプル自動化機能です。</p>
%AOFOPJESOPER%	<p>このカスケードは、コマンドを JESOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。JESOPER は主に JES 自動化用に定義されます。</p> <p><b>デフォルト:</b> AUTJES AUTSYS AUTOBASE AUTINIT1</p> <p>AUTJES は SA z/OS が他のサンプルで JESOPER 用に使用するオペレーター ID です。</p>
%AOFOPMONOPER%	<p>このカスケードは、コマンドを MONOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。MONOPER は、通常のモニターおよびサブシステムの始動に使用されます。</p> <p><b>デフォルト:</b> AUTMON AUTSYS AUTOBASE AUTINIT1</p> <p>AUTMON は SA z/OS が他のサンプルで MONOPER 用に使用するオペレーター ID です。</p>
%AOFOPRECOPER%	<p>このカスケードは、コマンドを RECOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。RECOPER は、リカバリー処理に使用されます。</p> <p><b>デフォルト:</b> AUTREC AUTSYS AUTOBASE AUTINIT1</p> <p>AUTREC は SA z/OS が他のサンプルで RECOPER 用に使用するオペレーター ID です。</p>

シノニム	使用法とデフォルト
%AOFOPSHUTOPER%	<p>このカスケードは、コマンドを SHUTOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。SHUTOPER は、自動化されたシャットダウンを調整します。</p> <p><b>デフォルト:</b> AUTSHUT AUTSYS AUTOBASE AUTINIT1</p> <p>AUTSHUT は SA z/OS が他のサンプルで SHUTOPER 用に使用するオペレーター ID です。</p>
%AOFOPGSSOPER%	<p>このカスケードは、コマンドを GSSOPER に送信するのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。GSSOPER は、汎用サブシステムの自動化に使用されます。</p> <p><b>デフォルト:</b> * AUTGSS AUTSYS AUTOBASE AUTINIT1</p> <p>AUTGSS は SA z/OS が他のサンプルで GSSOPER 用に使用するオペレーター ID です。</p> <p>「ASSIGN BY JOBNAME」機能をオフにしたい場合、つまり、拡張自動化 CGLOBAL 変数 <u>AOF_ASSIGN_JOBNAME</u> (203 ページの『付録 A グローバル変数』を参照) が 0 に設定されている場合は、シリアライゼーションの問題が発生する可能性があるため、アスタリスク (*) を除去する必要があります。</p> <p>注: 自動化テーブル処理の前に出される NetView コマンドの ASSIGN-BY-JOBNAME は、MVS ジョブ名に関連付けられているメッセージだけに影響します。</p>
%AOFOPWTORS%	<p>このカスケードは、WTORS に関係したコマンドをルーティングするのに使用されます。SA z/OS 自動タスクに標準名を使用していない場合、このシノニムを変更する必要があります。これを使用すると、すべての WTOR 処理が同一タスクで実行され、WTOR 処理がシリアライズ化されます。</p> <p><b>デフォルト:</b> * AUTGSS AUTSYS AUTOBASE AUTINIT1</p> <p>これは、AUTSYS が WTOR 処理すべてを実行することを指定します。</p>
%AOFOPGATOPER%	<p>このカスケードは、コマンドをこのドメインのゲートウェイ自動タスクにルーティングするのに使用されます。自動タスク名にドメイン ID が含まれているので、このシノニムを修正する必要があります。</p> <p><b>デフォルト:</b> GATRdomain</p> <p>AOF01 は、他のサンプルで使用されるデフォルトのドメインです。ゲートウェイ CLIST は GATOPER で実行されると考えられるため、バックアップはありません。</p>



## 付録 E TSO ユーザーのモニター

アクティブ TSO ユーザーは、SDF で SA z/OS コマンド DFTSOU (EVJETSOU) を使用してモニターできます。

TSO ユーザーのモニターを使用可能にするには、次の項目をユーザー AT INCLUDE フラグメント INGMSGU1 (またはユーザー自身のユーザー・メッセージ・テーブル) に追加します。

```
IF (MSGID='IEF125I' | MSGID='IEF126I' | MSGID='IEF450I')
  THEN EXEC(CMD('DFTSOU UPDATE') ROUTE(ALL *))
  DISPLAY(N) NETLOG(N) CONTINUE(Y);
```

さらに、「DFTSOU SCAN」を TSO サブシステムの STARTUP-REFRESHSTART ポリシーに書き込みます。

DFTSOU が UPDATE パラメーターを伴って呼び出されると、次のようになります。

- IEF125I の場合、ADD 要求が、メッセージを生成する TSO ユーザーの SDF に送信されます。
- IEF126I の場合、DELETE 要求が、メッセージを生成する TSO ユーザーの SDF に送信されます。
- IEF450I の場合、DELETE 要求が、失敗した TSO ユーザーの SDF に送信されます。IEF450I が指定され、かつトラップが INGMSGU1 でコード化されているときは、CONTINUE(Y) もコード化する必要があります。

DFTSOU が SCAN パラメーターを伴って呼び出される場合、**MVS D TS,L** コマンドが、現在アクティブなすべての TSO ユーザーの識別のために発行されます。このデータは次に SDF に渡されます。

SDF の更新は、SDF ツリー項目 TSUSERS に関連付けられます。



---

# 付録 F 選択されたハードウェア・コマンドの z13/z13 固有のタイムアウト値

このセクションでは、SAV4R1 APAR フィックスに組み込まれている APAR OA52002 (<http://www-01.ibm.com/support/docview.wss?uid=swg10A52002>) の実装について説明します。ProcOps フォーカル・ポイント・システムに APAR/PTF を適用後、ProcOps ランタイム使用のために OA52002 機能を活動化することができます。

---

## 特定タイムアウト値の定義

### AOF\_AAO\_ISQ\_Z13TMO\_USE Y

この SA AAO 変数を CNMSTYLE メンバーで定義し、その値を Y に設定します。ProcOps がターゲット・ハードウェアへの接続を初期化し、Z13TMO USE フラグが設定されるたびに、ProcOps は、特別なタイムアウト処理のために検出された D/T2964 または D/T2965 をそれぞれ登録します。SA ポリシーにそのようなプロセッサ・タイプがない場合、Z13TMO フラグを設定する必要はありません。

D/T2964 または D/T2965 の特定のタイムアウト値を定義します。上記のフラグを設定すると、通常のタイムアウトの ISQCCMD デフォルト値または独自の ISQ.SNMP.WAIT 設定の継続使用を決定できます。この場合、ネットログ内のデバッグ・メッセージのみが、使用される値に関する通知を行います。構成で D/T2964 プロセッサまたは D/T2965 プロセッサにそれぞれ AAO 変数を定義することにより、他の設定をオーバーライドするタイムアウト値を指定できます。

AOF\_AAO\_ISQ\_Z13TMO\_<thw> hh:mm:ss

#### thw

SA z/OS ポリシーで定義されたとおりの ProcOps ターゲット・ハードウェア名。

#### hh:mm:ss

タイムアウト値。ISQ.SNMP.WAIT の指定と同じ構文です。詳しくは、ISQCCMD のヘルプを参照してください。

妥当なタイムアウト値の基本計算については、「SA z/OS - z13 Specific Operational Characteristics」を参照してください。このリファレンスは『Add-on policies』からダウンロードできます。

---

## デバッグ・メッセージ

この APAR/PTF (OA52002) は、単一マシン・タイプ・セットに固有のハードウェア障害に対処するため、この RAS および問題判別サポートについて知らせるメッセージはネットログにのみ書き込まれます。

以下のメッセージ変数が使用されます。

#### cmd

CLEAR STORAGE 操作を許可するハードウェア・コマンド (ACTIVATE、DEACTIVATE、SYSRESET、および LOAD)

#### dtmo1

ISQCCMD デフォルト・タイムアウト (秒表記)

ACTIVATE=600

DEACTIVATE=300

SYSRESET=60

LOAD=60

#### dtmo2

ISQCCMD タイムアウト (M(m) 表記)

ACTIVATE=M(10)

DEACTIVATE=M(5)

SYSRESET=M(1)

LOAD=M(1)

***mtype***

マシン・タイプ番号、2964 または 2965

***thw***

SA ポリシーからの ProcOps ターゲット・ハードウェア名

***tmo***

ユーザー指定のタイムアウト (hh:mm:ss 表記)

APAR OA52002 に関連する以下の ProcOps デバッグ・メッセージが発行される場合があります。

**ISQ511I OA52002 Possible elongated CLEAR STORAGE period on *thw*, machine type *mtype*.**

説明: CPC のマシン・タイプが判別されたときに、ISQSTART 時または ISQXIII 時に発行されるメッセージ。これは、このターゲット・ハードウェアの後続のタイムアウト問題が、既知のハードウェア障害に関連しているかどうかを判別するために役立ちます。

**ISQ511I OA52002 Default or ISQ.SNMP.WAIT period *tmo* used on *thw* for command *cmd*.**

説明: このメッセージは、可能な CLEAR STORAGE オプションを指定したハードウェア・コマンドが実行されたが、この 2964/2965 タイプのターゲット・ハードウェアに対して Z13TMO オーバーライドが指定されていない場合に発行されます。

**ISQ511I OA52002 User defined z13 special timeout period *tmo* used for command *cmd*.**

説明: このメッセージは、CLEAR STORAGE を許可するハードウェア・コマンドに Z13TMO ユーザー・タイムアウト値が指定されている場合に発行されます。

**ISQ511I User timeout value exceeds z API maximum. Defaults *dtmo1* and *dtmo2* are used on *thw* for command *cmd*.**

説明: このメッセージは、サポートされる System z API 最大タイムアウトの 596 時間を超える Z13TMO ユーザー・タイムアウト値または ISQ.SNMP.WAIT タイムアウト値が指定された場合に発行されます。ProcOps デフォルト・タイムアウト値が使用されます。

## 付録 G 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使用することができますが、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認できません。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 の将来の方向性および指針に関する記述は、予告なく変更または撤回される場合があります。これらは目標および目的を提示するものにすぎません。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

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、類似する個人や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。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。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。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

©(お客様の会社名)(年).

このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. \_年を入れる\_.

## 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 Corporation 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。





この用語集には、下記の資料から引用した用語および定義が含まれています。

- *IBM Dictionary of Computing* New York: McGraw-Hill、1994。
- *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990、copyright 1990、米国規格協会 (ANSI)。このコピー版は、米国規格協会 (ANSI) (American National Standards Institute, 1430 Broadway, New York, New York 10018) からお求めいただけます。この資料からの定義には、定義の最後に (A) という記号を付けて区別しています。
- *Information Technology Vocabulary*。国際標準化機構および国際電気標準会議 (ISO/IEC JTC1/SC1) の第1 合同技術委員会第1 小委員会によって作成された資料。この用語集の既刊部分で定義されている用語については、その後に記号 (I) を付けて示してあります。ISO/IEC JTC1/SC1 で検討中の国際規格草案、委員会草案、および作業文書から転載した定義については、その後に記号 (T) を付けて示してあります。これは、SC1 の参加各国の間で最終的な合意に達していない用語であることを示しています。

この用語集では、次の相互参照が使用されています。

- ～と**対比**。これは、反対の意味または実質的に異なる意味を持つ用語を示します。
- ～の**使用すべきでない用語**。これは、この用語を使用すべきでないことを示します。用語集内の適切な場所で定義されている優先用語を示します。
- ～を**参照**。これは、この用語が現れる複数語の用語を示します。
- ～も**参照**。これは、同義語ではないが関連した意味を持つ用語を示しています。
- ～の**同義語**。これは、用語集内で定義されている優先用語と同じ意味を持つ用語を示します。
- ～と**同義**。これは、同じ意味を持つ他のすべての用語に対しての、定義済み用語の逆方向参照です。

## A

### ACF

自動化構成ファイル (automation configuration file) を参照。

### ACF/NCP

ネットワーク制御プログラムの拡張通信機能。拡張通信機能 (Advanced Communications Function) および ネットワーク制御プログラム (Network Control Program) を参照。

### ACF/VTAM

仮想記憶通信アクセス方式 (VTAM) 拡張通信機能 (Advanced Communications Function for the Virtual Telecommunications Access Method)。VTAM と同義。拡張通信機能 (Advanced Communications Function) および 仮想記憶通信アクセス方式 (VTAM) (Virtual Telecommunications Access Method (VTAM)) を参照。

### 能動的モニタリング (active monitoring)

SA z/OS 自動化制御ファイルでは、ユーザーが定義した定期的な間隔でリソース状況情報を送信請求することによって、この情報を獲得すること。受動モニター (passive monitoring) も参照。

### アダプター (adapter)

ワークステーションなどの装置が、モニター、プリンター、または他の一部の入出力装置などの別の装置と、通信できるようにするハードウェア・カード。

### 隣接ホスト (adjacent hosts)

モニターと制御の目的で、隣接する NetView セッションを使用して、対等関係で接続されたシステム。

### 隣接 NetView (adjacent NetView)

SA z/OS では、直接リンクを持たない 2 つの SA z/OS システム間で、通信パスとして定義されたシステム。隣接 NetView は、メッセージ転送のため、また、2 つの SA z/OS システム間の通信リンクとして使用される。例えば、フォーカル・ポイントからリモート・システムへの応答の送信時に、隣接 NetView が使用される。

### 拡張通信機能 (ACF) (Advanced Communications Function (ACF))

機能の分散およびリソースの共有を含む、システム・ネットワーク体系 (SNA) の概念を使用する、IBM ライセンス・プログラムのグループ (主として VTAM、TCAM、NCP、および SSP)。

## **拡張プログラム間通信 (APPC) (advanced program-to-program communication (APPC))**

システム・ネットワーク体系 (SNA) ネットワークにおける共同トランザクション処理をサポートするプログラム間通信サービスの集合。APPC は、指定されたシステムで、SNA 論理装置タイプ 6.2 を実現したものである。

## **Advanced Workload Analysis Reporter (zAware)**

zACI モードで活動化される、z Systems<sup>®</sup> 区画で稼働する IBM 分析アプライアンス。カスタマーはこのアプライアンスを使用して、同じ IBM Z クラスターで動作している他の LPAR のコンソール・メッセージ・ストリームをモニターし、傾向レポートを作成することができる。zAware とこれらの傾向レポートを利用すると、OS の障害やパフォーマンス低下をより正確に予測できるようになり、事前対応型のクラスターの導入に役立つ。

## **アラート (alert)**

SNA では、アラート条件の存在を通信するために、システム問題管理フォーカル・ポイントまたはコレクション・ポイントに送られる記録。

NetView では、即時に注意を払う必要がある優先度の高いイベントのこと。データベース・レコードは、ユーザー構成フィルターで定義された特定の イベント・タイプの場合に生成される。

## **アラート条件 (alert condition)**

問題判別、診断、および解決のプロセスの一部または全部が、制御点 (CP) でアクションを必要としている問題または切迫した問題のこと。

## **アラートしきい値 (alert threshold)**

SA z/OS がグラフィカル・インターフェースの関連アイコンをアラート・カラーに変更するレベルを決める、アプリケーションまたはボリューム・サービスの値。SA z/OS はアラートを出すこともできる。警告しきい値 (warning threshold) を参照。

## **AMC**

自動化マネージャー構成 (Automation Manager Configuration) を参照。

## **情報交換用米国標準コード (ASCII) (American Standard Code for Information Interchange (ASCII))**

データ処理システム、データ通信システム、および関連する装置間での情報交換に使用される標準コード。ASCII では、7 ビットのコード化文字 (パリティ検査を含め 8 ビット) からなるコード化文字セットが使用される。ASCII コード・セットは、制御文字と図形文字からなる。拡張 2 進化 10 進コード (Extended Binary Coded Decimal Interchange Code) も参照。

## **APF**

許可プログラム機能 (authorized program facility) を参照。

## **API**

アプリケーション・プログラミング・インターフェース (application programming interface) を参照。

## **APPC**

拡張プログラム間通信機能 (advanced program-to-program communication) を参照。

## **アプリケーション (application)**

SA z/OS では、アプリケーションとは、z/OS サブシステム、開始タスク、または SA z/OS により自動化およびモニターされるジョブを指す。SNMP 対応プロセッサでは、アプリケーションはサブシステムまたはプロセスを指すために使用できる。

## **アプリケーション項目 (Application entry)**

カスタマイズ・ダイアログによって作成され、アプリケーションのポリシーを表し、それを入れておくために使用される構造。

## **アプリケーション・グループ (application group)**

アプリケーションの名前のついた集合。アプリケーション・グループは SA z/OS エンタープライズ定義の一部で、モニター目的で使用される。

## **アプリケーション・プログラム (application program)**

在庫管理や給与計算を行うプログラムなど、ユーザーのために作成されたか、またはユーザーによって作成され、ユーザーの業務に適用されるプログラム。

ネットワークで、ステーションと接続したり、通信したりするために使用するプログラムのことで、ユーザーがアプリケーション本位の活動を実行できるようにする。

## アプリケーション・プログラミング・インターフェース (API) (application programming interface (API))

高水準言語で書かれたアプリケーション・プログラムが、オペレーティング・システムまたは他のプログラムの特定データまたは機能を使用できるようにするためのインターフェース。

## ApplicationGroup 項目 (ApplicationGroup entry)

カスタマイズ・ダイアログによって作成され、アプリケーション・グループのポリシーを表し、それを入れておくために使用される構成。

## ARM (Automatic restart management)

[自動リスタート管理 \(automatic restart management\)](#) を参照。

## ASCB (Address space control block)

アドレス・スペース制御ブロック。

## ASCB 状況 (ASCB status)

SA z/OS によって引き出される、アプリケーションの状況。z/OS アドレス・スペース制御ブロック (ASCB) を検索するルーチン (ASCB チェッカー) を実行して、特定のジョブ名を持つアドレス・スペースを見つけることによって得られる。ASCB チェッカーによって使用されるジョブ名は、アプリケーションに対してカスタマイズ・ダイアログで定義されたジョブ名である。

## ASCII

[情報交換用米国標準コード \(American Standard Code for Information Interchange\)](#) を参照。

## ASF

[自動化状況ファイル \(automation status file\)](#) を参照。

## 許可プログラム機能 (APF) (authorized program facility (APF))

制限付き機能の使用を許可されたプログラムの識別を可能にする機能。

## 自動コンソール操作 (ACO) (automated console operations (ACO))

システム・イベントまたはネットワーク・イベントへの応答として、オペレーターがコンソールから行うアクションを置き換えるため、または簡素化するための、自動化プロシージャの使用。

## 自動化機能 (automated function)

SA z/OS 自動化機能は、特定の自動化機能を実行するために割り当てられた自動化オペレーター、つまり NetView 自動開始タスクである。ただし、SA z/OS は、NetView 自動開始タスクに対して独自の同義語 (つまり自動化機能名) を定義するため、その機能名は SA z/OS で提供されるサンプル・ポリシー・データベースで参照される。例えば、自動化オペレーター AUTOBASE は、SA z/OS 自動化機能の BASEOPER に対応している。

## 自動リスタート管理 (ARM) (automatic restart management (ARM))

特定の状況下で、指定されたサブシステムおよびアプリケーションを自動的にリスタートすることにより、それらの可用性を改善する z/OS のリカバリー機能。自動リスタート管理は、z/OS のシステム間カップリング・ファシリティ (XCF) コンポーネントの 1 つの機能である。

## 自動再始動管理エレメント名 (automatic restart management element name)

MVS 5.2 以降の z/OS 自動再始動管理では、自動再始動管理に登録される アドレス・スペースごとに 16 文字の固有な名前を指定する必要がある。すべての自動リスタート管理のポリシーは、SA z/OS とのインターフェースを含めて、エレメント名によって定義される。

## 自動化 (automation)

検出された条件またはイベントに応じたアクションの自動開始。SA z/OS は、z/OS アプリケーション、z/OS コンポーネント、および z/OS を実行するリモート・システムを自動化できる。SA z/OS はまた、新たに自動化を開発するためのツールも提供する。

## 自動化エージェント

SA z/OS では、自動化機能は、自動化マネージャーと自動化エージェントの間で分割されている。監視、対処、および実施部分は NetView アドレス・スペース内に置かれ、自動化エージェントと呼ばれる。自動化エージェントは以下のことを行う。

- リカバリー処理
- メッセージ処理
- アクティブ・モニター: 伝搬状況を自動化マネージャーに伝搬

## 自動化構成ファイル (automation configuration file)

自動化構成ファイルの作成には、SA z/OS カスタマイズ・ダイアログを使用する必要がある。以下から構成される。

- 自動化マネージャー構成ファイル (AMC)
- NetView 自動化テーブル (AT)
- NetView メッセージ改訂テーブル (MRT)
- MPFLSTxx メンバー

### 自動化制御ファイル (ACF) (automation control file (ACF))

SA z/OS では、システム・レベルの自動化ポリシー情報が入っているファイル。SA z/OS がインストールされている NetView システムごとに 1 つのマスター自動化制御ファイルがある。追加のポリシー情報と、すべてのリソース状況情報は、ポリシー・データベース (PDB) に含まれている。自動化制御ファイルの作成には、SA z/OS カスタマイズ・ダイアログを使用する必要がある。これらは、手作業で編集してはならない。

### 自動化フラグ (automation flags)

SA z/OS では、リソース用に自動化されるオペレーター機能と、その自動化がアクティブになっている時間とを決める自動化ポリシー設定値。SA z/OS の実行中には、自動化は、自動化フラグ・ポリシー設定値、およびオペレーターが入力したオーバーライド設定値 (ある場合) によって制御される。自動化フラグは、カスタマイズ・ダイアログを使用して設定される。

### 自動化マネージャー

SA z/OS では、自動化機能は、自動化マネージャーと自動化エージェントの間に分割されている。調整機能、意思決定機能、および制御機能は、各シスプレックスの **自動化マネージャー** によって処理される。

自動化マネージャーには、シスプレックス内のすべての自動化リソースのモデルが含まれる。自動化エージェントは、状況情報を自動化マネージャーに送り、自動化マネージャーで指示されたアクションを実行する。

自動化マネージャーは、シスプレックス全体の自動化を提供する。

### 自動化マネージャー構成 (Automation Manager Configuration)

自動化マネージャー構成ファイル (AMC) には、シスプレックスにおける、または独立型システムの、自動化されたシステムのイメージが含まれる。 **自動化構成ファイル (automation configuration file)** も参照。

### 自動化 NetView (Automation NetView)

SA z/OS では、コマンド・プロシージャーを使用して定常オペレーター・タスクを実行し、また他の方法でシステムおよびネットワークの管理を自動化して、メッセージまたは管理サービスの単位に対して自動応答を発行する NetView。

### 自動化オペレーター

NetView 自動化オペレーターは、特定の自動化機能を実行するために割り当てられた NetView 自動開始タスクである。 **自動化機能 (automated function)** も参照。NetView 自動化オペレーターは、メッセージを受信し、自動化プロシージャーを処理します。自動化オペレーターと関連したログオン済みユーザーはない。各自動化オペレーターはオペレーティング・システム・タスクであり、他の NetView タスクと並行して実行される。自動化オペレーターは自動化プロシージャーをスケジューリングする JES2 メッセージを扱うようにセットアップでき、自動化ステートメントによってそのようなメッセージを自動化オペレーターに送付することができる。オペレーター端末タスク (*operator station task*) と同様。SA z/OS メッセージ・モニター・タスクおよびターゲット制御タスクは自動化オペレーターである。

### 自動化ポリシー (automation policy)

個別のシステムについて、自動化を規制するポリシー情報。これには、アプリケーション、z/OS サブシステム、z/OS データ・セット、および z/OS コンポーネントの自動化が含まれる。

### 自動化ポリシー設定 (automation policy settings)

自動化制御ファイルに入っている自動化ポリシー情報。この情報は、カスタマイズ・ダイアログを使用して入力する。ユーザーは、カスタマイズ・ダイアログを使用してこれらの設定値を表示または修正することができる。

### 自動化プロシージャー (automation procedure)

NetView のコマンド・リストまたは高水準言語で書かれたコマンド・プロセッサとしてパッケージされた、一連のコマンド。自動化プロシージャーは自動化機能を実行し、NetView のもとで稼働する。



### 自動化ルーチン (automation routines)

SA z/OS では、NetView 自動化テーブル、またはユーザー作成の自動化プロシージャー から呼び出すことができる自己完結型自動化ルーチンのセット。

### 自動化状況ファイル (ASF) (automation status file (ASF))

SA z/OS で、自動化されるサブシステム、コンポーネント、データ・セットのそれぞれの状況情報が入っているファイル。この情報は、アクションを取るとき、または取るべきアクションを決めるときに、SA z/OS 自動化によって使用される。AOC/MVS のリリース 2 およびそれより後のリリースでは、状況情報も操作情報ベースで保守される。

### 自動化テーブル (AT) (automation table (AT))

[NetView 自動化テーブル \(NetView automation table\)](#) を参照。

### 自動開始タスク (autotask)

メッセージを受信したり、自動化プロシージャーを処理する NetView の自動化タスク。自動開始タスクと関連したログオン済みユーザーはない。各自動開始タスクはオペレーティング・システム・タスクであり、他の NetView タスクと並行して実行される。自動開始タスクは、自動化プロシージャーをスケジュールする JES2 メッセージを処理するようにセットアップできる。また、自動化ステートメントを使用すれば、そのようなメッセージを自動開始タスクにルーティングできる。オペレーター端末タスク (*operator station task*) と同様。SA z/OS のメッセージ・モニター・タスクとターゲット制御タスクは自動開始タスクである。自動化オペレーター (*automation operator*) とも呼ばれている。

### 使用可能 (available)

VTAM プログラムでは、アクティブ、接続済み、使用可能であり、そのセッション限度でない論理装置 (LU) に関する用語。

## B

### 基本制御プログラム (BCP) (Base Control Program (BCP))

MVS および z/OS オペレーティング・システムに不可欠のサービスを提供するプログラム。プログラムには、システム・リソースを管理する機能が含まれる。この機能には、入出力、作業単位のディスパッチ、z/OS UNIX System Services カーネルがある。[多重仮想記憶 \(Multiple Virtual Storage\)](#) および z/OS も参照。

### 基本モード (basic mode)

論理区画を使用しない、中央処理装置のモードのこと。[論理分割モード \(logically partitioned mode\)](#) と対比。

## BCP

[基本制御プログラム \(Base Control Program\)](#) を参照。

### BCP 内部インターフェース (BCP Internal Interface)

IBM Z プロセッサ・ファミリーのプロセッサ機能。情報の交換またはプロセッサ制御機能を実行するため、z/OS のような基本制御プログラムとプロセッサ・サポート・エレメントの間の通信が可能になる。この機能を使用するプログラムは、ACTIVATE または SYSTEM RESET などのハードウェア操作を実行することができる。

### ビーコン (beaconing)

回線の切断または故障の検出時における、コンソールまたはワークステーションによるフレームまたはメッセージ (ビーコン) の反復伝送。

### BookManager®

ユーザーがワークステーション上にソフトコピー文書を表示できる IBM 製品。

## C

### 中央処理装置 (CP) (central processor (CP))

命令実行、初期プログラム・ロード (IPL)、および他の機械操作のための順序付けおよび処理機能を含むコンピューターの一部。

### 中央演算処理複合システム (CPC) (central processor complex (CPC))

中央記憶装置、(1 つ以上の) 中央処理装置、(1 つ以上の) タイマー、および (1 つ以上の) チャンネルからなるハードウェアの物理的な集合体。

### 中央設置場所 (central site)

分散データ処理ネットワークでは、中央設置場所は、通常、アラート、アプリケーション設計、および問題管理などのリモート・システム管理タスクのフォーカル・ポイントとして定義される。

### チャンネル (channel)

シグナルをそれに沿って送信できるパス。例えば、データ・チャンネル、出力チャンネルなど。[リンク \(link\)](#) も参照。

### チャンネル・パス ID (channel path identifier)

各チャンネル・パスに割り当てられたシステム固有の値。

### チャンネル接続 (channel-attached)

入出力チャンネルによって、ホスト処理装置に直接接続すること (例えば、チャンネル接続装置など)。

通信回線によってではなく、ケーブルによって制御装置に接続すること。[リンク接続 \(link-attached\)](#) と対比。[ローカル \(local\)](#) と同義。

### CHPID

SA z/OS では、チャンネル・パス ID、すなわちチャンネルのアドレス。

### CHPID ポート (CHPID port)

システム名、論理区画、およびチャンネル・パスを記述するラベル。

### CI (Console Integration)

[コンソール統合 \(console integration\)](#) を参照。

### CICS/VS

VS 顧客情報管理システム (CICS/VS) (Customer Information Control System for Virtual Storage)。[顧客情報管理システム \(Customer Information Control System\)](#) を参照。

### CLIST

[コマンド・リスト](#) を参照。

### クローン (clone)

いくつかの異なるシステム特有の値を基本定義に置き換えることにより、基本アプリケーション定義から得られるアプリケーション・インスタンスのための定義の集合。

### クローン ID (clone ID)

MVS SYSCClone または VTAM サブエリア番号などのシステム特有の値を処理する一般的な方法。クローン ID は、アプリケーション定義およびコマンドに置き換えられて、インスタンスを生成するシステムのための基本のアプリケーション定義をカスタマイズすることができる。

### コマンド (command)

操作の実施または特定プログラムの実行の要求。

### コマンド機能 (command facility)

ネットワークの操作をモニター、制御、自動化、および改善できる、コマンド・プロセッサの基本となる NetView のコンポーネント。NCCF の後継。

### コマンド・リスト (CLIST) (command list (CLIST))

ユーザーのために特定の機能を実行するように設計された、NetView コマンド・リスト言語または REXX 言語で書かれたコマンドとステートメントのリスト。最も単純な形式のコマンド・リストはコマンドのリスト。より複雑なコマンド・リストでは、変数置換と条件ロジックが組み合わさって、従来のプログラムのようなものになる。コマンド・リストは、通常、コンパイルするのではなく解釈する。

SA z/OS では、自動化プロシージャに使用できる REXX コマンド・リストのこと。

### コマンド・プロシージャ (command procedure)

NetView では、コマンド・リストまたはコマンド・プロセッサのいずれか。

### コマンド・プロセッサ (command processor)

特定の機能を実行するように設計されたモジュール。アセンブラーまたは高水準言語 (HLL) で書くことができるコマンド・プロセッサはコマンドとして発行される。

### コマンド・ツリー/2 (Command Tree/2)

OS/2 ウィンドウでコマンドを作成するのに役立ち、指定した宛先 (3270 セッション、ファイル、コマンド行、またはアプリケーション・プログラムなど) にそのコマンドをルーティングする OS/2 ベースのプログラム。これは、コマンドを構築して、それを指定の宛先に送付するための機能をオペレーターに提供する。

### 共通コマンド (common commands)

CPC 操作管理コマンドの SA z/OS サブセット。



## 共通ユーザー・アクセス (CUA) 体系 (Common User Access (CUA) architecture)

人とワークステーションまたは端末との間のダイアログのための指針。

## 通信制御装置 (communication controller)

装置に保管され実行される 1 つ以上のプログラムによって、あるいは制御装置が接続されたプロセッサで実行されるプログラムによって、動作が制御される通信制御装置のタイプ。これは、回線制御の詳細およびネットワークを経由するデータのルーティングを管理する。

## 通信回線 (communication line)

通信回線 (telecommunication line) の非推奨用語。

## 接続ビュー (connectivity view)

SA z/OS では、図形イメージで入出力装置を表し、線でそれらの接続方法を表した表示。

## コンソール自動化 (console automation)

通常はオペレーターが扱うコンソール入力を NetView 機能に行わせるプロセス。

## コンソール接続 (console connection)

SA z/OS では、PS/2 コンピューターとターゲット・システム間の 3270 または ASCII (シリアル) 接続。この接続においては、ワークステーションは (ターゲット・システムには) コンソールとして見える。

## コンソール統合 (CI) (console integration (CI))

オペレーティング・システム (OS) によってサポートされている場合に、オペレーティング・システム (OS) メッセージを内部ハードウェア・インターフェースを介して転送し、システム・コンソールで表示できるようにするハードウェア機能。逆に、システム・コンソールで入力されたオペレーティング・システム (OS) コマンドは、処理用のオペレーティング・システム (OS) との内部ハードウェア・インターフェースを介して転送できる。

## コンソール (consoles)

ユーザーのエンタープライズ・システムを管理する、ワークステーションおよび 3270 タイプの装置。

## 結合データ・セット (couple data set)

XCF 結合データ・セット・フォーマット・ユーティリティーによって作成され、その指定されたタイプに応じて、シスプレックス内の一部またはすべての z/OS システムで共有されるデータ・セット。シスプレックス結合データ・セット (sysplex couple data set) および XCF 結合データ・セット (XCF couple data set) も参照。

## カップリング・ファシリティ (coupling facility)

シスプレックス内で、高速キャッシュ、リスト処理、およびロック機能を提供するハードウェア・エレメント。

## CP

中央処理装置 (central processor) を参照。

## CPC

中央処理装置複合体 (central processor complex) を参照。

## CPC 操作管理機能コマンド (CPC operations management commands)

System/390® CPC の操作を制御するためのコマンドおよび応答の集合。

## CPC サブセット (CPC subset)

CPC の全部または一部。これには、単一の制御プログラム (CP) をサポートするための最小限のリソースが含まれる。

## CPU

中央演算処理装置 (Central processing unit)。 プロセッサ (processor) の非推奨用語。

## システム間カップリング・ファシリティ (XCF) (cross-system coupling facility (XCF))

シスプレックス内で実行される許可プログラム間の連携をサポートする機能を提供する、z/OS のコンポーネント。

## 顧客情報管理システム (CICS) (Customer Information Control System (CICS))

リアルタイム・ベースで多数のエンド・ユーザーに対して端末ユーザーとデータベース間のオンライン通信を制御する汎用のトランザクション用プログラム。

## カスタマイズ・ダイアログ (customization dialogs)

カスタマイズ・ダイアログは ISPF アプリケーションである。このダイアログは、エンタープライズ・ポリシー (例えば、エンタープライズ・リソース、リソース間の関係、エンタープライズにおけるシス

テムの自動化ポリシーなどをカスタマイズするために使用される。カスタマイズ・ダイアログの使用  
方法については、「[IBM Z System Automation カスタマイズとプログラミング](#)」を参照。

## D

### DataPower® X150z

[IBM Websphere DataPower Integration Appliance X150 for zEnterprise® \(DataPower X150z\)](#) を参照。

### DASD

[直接アクセス記憶装置 \(direct access storage device\)](#) を参照。

### データ・サービス・タスク (data services task (DST))

VSAM ファイルまたはネットワーク管理情報を含むネットワーク装置 内のデータを収集、記録、管理す  
る NetView のサブタスク。

### データ・セット (data set)

複数の規定された配置の 1 つの集合体からなり、システムがアクセスする 制御情報によって記述され  
た、データ・ストレージおよび検索の大きい単位。

### データ・セット・メンバー (data set members)

名前別に検索できる大きいファイルの、個別に名前が付けられた エレメントである、区画データ・セ  
ットのメンバー。

### DBCS

[2 バイト文字セット \(double-byte character set\)](#) を参照してください。

### DCCF

[使用禁止コンソール通信機能 \(disabled console communication facility\)](#) を参照。

### DCF

[文書構成プログラム \(Document Composition Facility\)](#) を参照。

### DELAY レポート (DELAY Report)

システム内の各ジョブの活動および各ジョブを遅らせている、ハードウェアとソフトウェアのリソー  
スを示す RMF 報告書。

### 装置 (device)

1 個の機器。装置には、ワークステーション、プリンター、ディスク・ドライブ、磁気テープ装置、リ  
モート・システム、または通信制御装置がある。特定のスイッチに接続されたすべての装置について、  
情報を表示したり、装置に対するパスとジョブを制御したりできる。

### DEVR レポート (DEVR Report)

ジョブを遅らせている入出力装置のアクティビティーについての 情報を表す RMF 報告書。

### ダイアログ (dialog)

対話式 3270 パネル。

### 直接アクセス記憶装置 (DASD) (direct access storage device (DASD))

ディスク・ドライブなど、ストレージに直接アクセス可能な装置。

### コンソール通信制御機能 (DCCF) (disabled console communication facility (DCCF))

システムのリカバリー中に、限定された機能のコンソール通信を提供する z/OS コンポーネント。

### ディスク・オペレーティング・システム (DOS) (disk operating system (DOS))

プログラムおよびデータの補助ストレージとして、ディスクやディスクレットを 使用するコンピューター  
・システムのオペレーティング・システム。

プログラムの処理を制御するパーソナル・コンピューター (PC) 用ソフトウェア。IBM パーソナル・コ  
ンピューターの場合は、フルネーム はパーソナル・コンピューター・ディスク・オペレーティング・  
システム (Personal Computer Disk Operating System (PCDOS)) である。

### ディスプレイ (display)

ワークステーション画面やハードコピー装置 (通常の場合) における情報表示装置。

[パネル \(panel\)](#) の非推奨用語。

### 分散マネージャー (distribution manager)

ホスト・システムが、コンピューターのネットワーク内のファイルおよびプログラムを使用、送信、  
削除できるようにする、NetView プログラムのコンポーネント。

## 文書構成プログラム (Document Composition Facility (DCF))

プリンターへの入力のフォーマット設定に使用される IBM ライセンス・プログラム。

## ドメイン (domain)

アクセス方式とそのアプリケーション・プログラム、通信制御装置、接続回線、モデム、および接続されたワークステーション。

SNA では、システム・サービス制御点 (SSCP) と物理装置 (PU)、論理装置 (LU)、リンク、リンク・ステーション、および SSCP が活動化要求と非活動化要求によって制御できる関連リソース。

## 2 バイト文字セット (DBCS) (double-byte character set (DBCS))

各文字が 2 バイト・コードで表される漢字などの文字セット。

## DP エンタープライズ (DP enterprise)

データ処理 (DP) エンタープライズ。

## DSIPARM

このファイルは NetView カスタマイズ用のメンバーの集合である。

## DST

データ・サービス・タスク (DST) (Data Services Task)。

## E

## EBCDIC

[拡張 2 進化 10 進コード \(Extended Binary Coded Decimal Interchange Code\)](#) を参照。

## ECB

[イベント制御ブロック \(event control block\)](#) を参照。

## EMCS

拡張複数コンソール・サポート (Extended multiple console support)。 [複数コンソール・サポート \(multiple console support\)](#) を参照。

## エンタープライズ (enterprise)

総合的事業体を形成し、情報システムを必要とする、すべての操作可能エンティティ、機能、およびリソースの複合体。

## エンタープライズ・システム・アーキテクチャー (ESA) (Enterprise Systems Architecture (ESA))

データ・セットの管理に必要な労力を削減し、システム、サブシステム、およびアプリケーション機能のアドレス可能性を拡張する、ハードウェア・アーキテクチャー。

## 項目 (entries)

パネルで入力されるプロセッサなどのリソース。

## 項目タイプ

自動化やモニターに使用されるリソース。例えば、プロセッサまたはアプリケーション。

## 環境 (environment)

データ処理 (DP) エンタープライズ。

## エラーしきい値 (error threshold)

SA z/OS が、アプリケーション、サブシステム、またはコンポーネントを再始動またはリカバリーしようとしたり、データ・セットをオフロードしようとしたりするのを停止する必要があるタイミングを指定する、自動化ポリシー設定。

## ESA

[エンタープライズ・システム・アーキテクチャー \(Enterprise Systems Architecture\)](#) を参照。

## イベント (event)

NetView では、ネットワークの物理エレメントにおける操作の不規則性を示す記録。

タスクに対して意味のあることが発生すること。例えば、入出力操作などの非同期操作の完了など。

イベントはトリガー条件の一部で、トリガー条件の全イベントが発生した場合に、アプリケーションの開始またはシャットダウンが実行される。

## イベント制御ブロック (event control block (ECB))

イベントの状況を表すために使用される制御ブロック。

### 例外条件 (exception condition)

通常動作からの逸脱がシステムで発生すること。SA z/OS のモニターでは、例外条件を強調表示して、SA z/OS エンタープライズを例外別に管理できるようにする。

### 拡張 2 進化 10 進コード (Extended Binary Coded Decimal Interchange Code (EBCDIC))

テキスト・データ表記用に開発された、256 個の 8 ビット文字からなるコード化文字セット。情報交換用米国標準コード (American Standard Code for Information Interchange) も参照。

### 拡張回復機能 (XRF) (extended recovery facility (XRF))

高可用性アプリケーションと指定端末間のセッションにおいて、z/OS、VTAM、ホスト・プロセッサ、または高可用性アプリケーションにおける障害の影響を最小限にする機能。この機能により、障害が発生したサブシステムからセッションを代替サブシステムが引き継ぐことができる。

## F

### フォールバック・システム (fallback system)

2 次システム (secondary system) を参照。

### フィールド (field)

論理的に関連し、1 つの単位として処理される、1 つのレコード内のバイトの集合体。

### ファイル・マネージャー・コマンド (file manager commands)

自動化制御ファイルまたは操作情報ベースとの間でデータの読み書きを行う SA z/OS コマンドのセット。これらのコマンドは、SA z/OS 機能を使用する自動化の開発に役立つ。

### フォーカル・ポイント (focal point)

NetView では、フォーカル・ポイント・ドメインは中央ホスト・ドメインを指す。ネットワーク管理データの制御を含む、管理サービス・エレメントの中央制御点である。

### フォーカル・ポイント・システム

1 つ以上のターゲット・システムを管理し、制御できるシステム。IBM プロダクトに関連した一連の各種フォーカル・ポイント・システムがある。

**SA z/OS プロセッサ操作 フォーカル・ポイント・システム。**これは、SA z/OS ホスト・コードがインストールされている NetView システムである。SA z/OS プロセッサ操作フォーカル・ポイント・システムは、システムや、システムが制御するマシンのオペレーター・コンソールからメッセージを受け取る。これは、そのターゲット・システム用に全部のシステムおよび操作コンソール機能を提供する。これは、これらのシステムを IPL するために使用することができる。S/390<sup>®</sup> マイクロプロセッサ・クラスターのハードウェア管理コンソールには、一部の制限が適用されることに注意されたい。

**SA z/OS SDF フォーカル・ポイント・システム。**SA z/OS SDF フォーカル・ポイント・システムは、エンタープライズ内の他の SA z/OS NetView から状況情報を収集する SA z/OS NetView システム。

**状況フォーカル・ポイント・システム。**NetView では、STATMON、VTAM、および NLDM がネットワーク・リソースに関する状況情報を送るシステム。

**ハードウェア管理コンソール。**フォーカル・ポイント・ハードウェアとしてリストにあげられていないが、ハードウェア管理コンソールは S/390 マイクロプロセッサ・クラスターのコンソール機能用のフォーカル・ポイントとして働く。この定義の他のすべてのフォーカル・ポイントと違って、ハードウェア管理コンソールは LAN 接続ワークステーション上で稼働する。

### フレーム (frame)

System/390 マイクロプロセッサ・クラスターの場合には、フレームには 1 つまたは 2 つの中央処理装置複合システム (CPC)、サポート・エレメント、および AC 電源分配が含まれる。

### フルスクリーン・モード (full-screen mode)

NetView では、一度にワークステーション画面全体の内容を表示できるようにする、パネル表示の 1 つの形式。フルスクリーン・モードは、空欄書き込み式のプロンプト用として使用できる。ライン・モード (line mode) と対比。

## G

### ゲートウェイ・セッション (gateway session)

他のシステムとの NetView-NetView タスク・セッションのことで、そのセッションでは SA z/OS アウトバウンド・ゲートウェイ・オペレーターは人間のオペレーターの介入なしに他の NetView セッションにログオンする。ゲートウェイ・セッションの両端には、インバウンドとアウトバウンドの両方のゲートウェイ・オペレーターがある。

## 総称アラート (generic alert)

NetView などのアラート受信側に保管されるコード・ポイント (IBM が定義し、ユーザーまたはアプリケーション・プログラムによってカスタマイズされる) を使用するエンコード・アラート情報。

## グループ (group)

構成ダイアログによって定義されたターゲット・システムの集合体。インストールすることにより、物理サイト、組織エンティティまたはアプリケーション・エンティティを参照するグループをセットアップできる。

## グループ項目 (group entry)

グループのポリシーを表し、これを入れておくために使用される、カスタマイズ・ダイアログで作成される構成。

## グループ項目タイプ (group entry type)

カスタマイズ・ダイアログによって定義されたターゲット・システムの集合体。物理サイトまたは組織エンティティを参照するグループをインストール時にセットアップできる。グループのタイプは STANDARD や SYSPLEX などに行うことができる。

## H

### ハードウェア管理コンソール (HMC) (Hardware Management Console (HMC))

データ・センター担当者が IBM Z のハードウェアおよびソフトウェアのリソースを構成、制御、モニター、および管理するために使用するユーザー・インターフェース。HMC はサポート・エレメントを通じて各中央演算処理装置複合システム (CPC) と通信する。

### ハードウェア管理コンソール・アプリケーション (HWMCA) (Hardware Management Console Application (HWMCA))

ハードウェア・エレメントに対する単一の制御点および単一システム・イメージを提供する、直接操作のオブジェクト指向のグラフィカル・ユーザー・インターフェース。HWMCA は、グループ分けのサポート、カラーを使用した集合体のリアルタイムのシステム状況、統合されたハードウェア・メッセージ・サポート、統合されたオペレーティング・システム・メッセージ・サポート、統合されたサービス・サポート、および単一システム、複数システム、またはシステムのグループをターゲットとするハードウェア・コマンドを提供する。

### ヘルプ・パネル (help panel)

コマンドの使用方法またはプロダクトの別の機能を通知するオンライン・パネル。

### 階層 (hierarchy)

NetView プログラムでは、ネットワーク内で編成 (レベル) を構成する、リソース・タイプ、表示タイプ、およびデータ・タイプ。

### 高水準言語 (HLL) (high-level language (HLL))

アSEMBラー言語からの一定レベルの抽象化、および特定タイプのマシンからの独立性を提供するプログラミング言語。NetView プログラムの場合には、高水準言語は PL/I および C である。

## HLL

[高水準言語 \(high-level language\)](#) を参照。

### ホスト (1次プロセッサ) (host (primary processor))

コマンドを入力するプロセッサ (発行側プロセッサとも呼ばれる)。

### ホスト・システム (host system)

結合システムまたは分散システム環境では、中央自動化の機能が稼働するシステム。SA z/OS 関係の資料では、ホストの代わりにターゲット・システムまたはフォーカル・ポイント・システムを指す。

## HWMCA

[ハードウェア管理コンソール・アプリケーション \(Hardware Management Console Application\)](#) を参照。

### ハイパーバイザー (Hypervisor)

同じハードウェア装置上でオペレーティング・システムまたは仮想サーバーの複数のインスタンスを同時に実行できるようにするプログラム。ハイパーバイザーは、ハードウェア上で直接実行する、オペレーティング・システム内で実行する、あるいはプラットフォーム・ファームウェアに組み込むことができる。ハイパーバイザーの例として、PR/SM、z/VM、および PowerVM® Enterprise Edition がある。

## I



### **IBM Secure Service Container (SSC)**

zAware または z/VSE® VNA などのファームウェアまたはソフトウェア・ベースのアプライアンス用に基本インフラストラクチャーのランタイム・サポートおよびデプロイメント・サポートを提供する、SSC 動作モードで実行するように活動化される IBM Z 区画。

### **IBM System z Application Assist Processor (zAAP)**

Java 実行環境を提供する特殊プロセッサ。Java ベースの Web アプリケーションを、中核の z/OS ビジネス・アプリケーションおよびバックエンド・データベース・システムと統合できる。

### **IBM System z Integrated Information Processor (zIIP)**

[統合情報プロセッサ \(IIP\) \(Integrated Information Processor \(IIP\)\)](#) を参照。

### **IBM Websphere DataPower Integration Appliance X150 for zEnterprise (DataPower X150z)**

XML および Web サービス処理を単純化し、機密保護機能を高め、最適化するための専用アプライアンス。

### **IBM Workload Scheduler (IWS)**

[ZWS](#) を参照。

### **IBM Z Workload Scheduler (ZWS)**

z/OS 環境内でジョブを計画し、実行し、追跡するスケジューラー。以前には、IBM Workload Scheduler for z/OS (IWS)、IBM Tivoli Workload Scheduler for z/OS (TWS)、または OPC/A と呼ばれていました。

### **IBM zEnterprise 196 (z196)**

新しいプロセッサ・チップ上に構築され、拡張されたメモリー機能とキャパシティー、セキュリティー、およびオンデマンド機能拡張を備え、既存のメインフレーム・ワークロードと大規模な統合に対応する最新世代の System z ファミリーのサーバー。

### **入出力リソース番号 (I/O resource number)**

チャンネル・パス ID (CHPID)、装置番号、などの組み合わせ。 [内部トークン \(internal token\)](#) を参照。

### **イメージ (images)**

定義するプロセッサおよび入出力装置のグループ化。マルチプロセッサが 1 つの中央処理装置イメージとして機能できるように、単一イメージ方式を定義することができる。

### **IMS**

[情報管理システム \(Information Management System\)](#) を参照。

### **IMS/VS**

[情報管理システム/仮想ストレージ \(Information Management System/Virtual Storage\)](#) を参照。

### **インバウンド (inbound)**

SA z/OS では、PC またはターゲット・システムからフォーカル・ポイント・システムに送信されるメッセージ。

### **インバウンド・ゲートウェイ・オペレーター (inbound gateway operator)**

送信システムでアウトバウンド・ゲートウェイ・オペレーターから 着信メッセージ、コマンド、および応答を受け取る自動化オペレーター。インバウンド・ゲートウェイ・オペレーターは、ゲートウェイ・セッションを使用して他のシステムとのコミュニケーションを処理する。

### **情報管理システム (IMS) (Information Management System (IMS))**

複合データベースおよび端末ネットワークを管理できる、データベース・マネージャーおよびトランザクション処理で使用可能な複数のシステム環境。

### **情報管理システム/VS (IMS/VS) (Information Management System/Virtual Storage (IMS/VS))**

複合システム・データベースおよびネットワークを管理できる データベース/データ・コミュニケーション (DB/DC) システム。 [情報管理システム \(Information Management System\)](#) と同義。

### **初期マイクロプログラム・ロード (initial microprogram load)**

マイクロプログラムをコンピューター・ストレージにロードする操作。

### **初期プログラム・ロード (IPL) (initial program load (IPL))**

オペレーティング・システムが操作を開始できるようにする初期設定手順。

就業日の開始時またはシステム誤動作の後に、構成イメージをストレージにロードするプロセスのこと。

システム・プログラムをロードし、ジョブを実行するためにシステムを準備するプロセス。

### 初期化の自動化 (initialize automation)

SA z/OS が初期化される時に、各サブシステムに正しい z/OS 開始コマンドを送出する、SA z/OS 提供の自動化機能。自動化により、確実に、サブシステムは自動化制御ファイルに指定された順序で始動され、前提条件のアプリケーションが使用できる。

### 入出力構成データ・セット (IOCDs) (input/output configuration data set (IOCDs))

入出力構成プログラム (IOCP) によって作成され、プロセッサ・コントローラーに関連付けられたディスク・ファイルに保管される構成定義。

### 入出力サポート・プロセッサ (IOSP) (input/output support processor (IOSP))

1次サポート・プロセッサの入出力サポート機能およびプロセッサ・コントローラーの保守サポートを提供するハードウェア装置。

### 統合情報プロセッサ (IIP) (Integrated Information Processor (IIP))

選択されたデータやトランザクションの処理ワークロード、および選択されたネットワーク暗号化ワークロードのコンピューティング能力を提供する特殊プロセッサ。

### 対話式システム生産性向上機能 (ISPF) (Interactive System Productivity Facility (ISPF))

フルスクリーン・エディターおよびダイアログ・マネージャとして機能する IBM ライセンス・プログラムの 1 つ。これはアプリケーション・プログラムの作成に使用され、標準的な画面パネルおよびアプリケーション・プログラマーと、端末ユーザーとの間の対話式ダイアログを生成する手段を提供する。タイム・シェアリング・オプション (Time Sharing Option) も参照。

### 関係オペレーター・リスト (interested operator list)

特定のターゲット・システムからのメッセージを受け取るオペレーターのリスト。

### 内部トークン (internal token)

論理トークン (LTOK) の 1 つ。入出力リソースまたはオブジェクトを認識する名前で、IODF に保管される。

### IOCDs

[入出力構成データ・セット \(input/output configuration data set\)](#) を参照。

### IOSP

[入出力サポート・プロセッサ \(input/output support processor\)](#) を参照。

### IPL

[初期プログラム・ロード \(initial program load\)](#) を参照。

### ISPF

[対話式システム生産性向上機能 \(Interactive System Productivity Facility\)](#) を参照。

### ISPF コンソール (ISPF console)

SA z/OS カスタマイズ・パネルのランタイム・パネルを使用するには、この 3270 タイプのコンソールから ISPF にログオンする。

### 発行ホスト (issuing host)

処理用にコマンドを入力するための基本プログラム。[1次ホスト \(primary host\)](#) を参照。

### J

#### JCL

[ジョブ制御言語 \(job control language\)](#) を参照。

#### JES

[ジョブ入力サブシステム \(job entry subsystem\)](#) を参照。

#### JES2

システムにジョブを受け入れ、内部形式に変換し、実行のために選択し、出力を処理し、それらをシステムからパーズする、MVS サブシステム。プロセッサが複数あるインストール先では、各 JES2 プロセッサがそのジョブ入力、スケジューリング、および出力処理を独立して制御する。[ジョブ入力サブシステム \(job entry subsystem\)](#) および [JES3](#) も参照

#### JES3

システムにジョブを受け入れ、内部形式に変換し、実行のために選択し、出力を処理し、それらをシステムからパーズする、MVS サブシステム。いくつかの疎結合処理装置を持つ複合システムにおいて、JES3 プログラムはプロセッサを管理し、グローバル・プロセッサが複数のローカル・プロセッサに対して中央制御を行い、共通ジョブ・キューを使用してジョブをそれらのプロセッサに配布できるようにする。[ジョブ入力サブシステム \(job entry subsystem\)](#) および [JES2](#) も参照。



## ジョブ (job)

コンピューターの作業単位を完全に定義するデータの集合。通常、1つのジョブにはすべての必要なコンピューター・プログラム、リンケージ、ファイル、およびオペレーティング・システムに対する命令が含まれる。

アドレス・スペース。

## ジョブ制御言語 (JCL) (job control language (JCL))

ジョブのステートメント (オペレーティング・システムに対しジョブの識別あるいはジョブの要件の記述を行うために使用する) を表すために設計された問題指向の言語。

## ジョブ入力サブシステム (JES) (job entry subsystem (JES))

ジョブをシステムに受け入れ、ジョブによって生成されるすべての出力データを処理する、IBM ライセンス・プログラム。SA z/OS 資料では、特に明記されていない限り、JES は JES2 または JES3 を指す。JES2 および JES3 も参照。

## K

### 漢字 (Kanji)

日本語で使用される表意文字セット。2 バイト文字セット (double-byte character set) も参照してください。

## L

### LAN

ローカル・エリア・ネットワーク (Local Area Network) を参照してください。

### ライン・モード (line mode)

端末装置画面のメッセージ域に、情報が一度に 1 行で表される、画面表示の形式。フルスクリーン・モード (full-screen mode) と対比。

### リンク (link)

SNA では、リンク接続と、ネットワーク・ノードを結合するリンク・ステーションの組み合わせ。例えば、システム/370 チャンネルとその関連したプロトコル、同期データ・リンク制御 (SDLC) の制御下のシリアル・バイ・ビット接続など。同期データ・リンク制御 (synchronous data link control) を参照。

SA z/OS では、リンク接続は伝送の物理メディアのことである。

### リンク接続 (link-attached)

通信回線によって物理的に接続された装置を記述する。チャンネル接続 (channel-attached) と対比。

### Linux on z Systems

Linus Torvalds によって考案され、インターネットを介して発展した、UNIX に似たオープン・ソース・オペレーティング・システム。

### ローカル (local)

通信回線を使用せずに直接アクセスされる装置に関する用語。チャンネル接続 (channel-attached) と同義。

### ローカル・エリア・ネットワーク (LAN) (local area network (LAN))

通信用に 1 セットの装置が接続されたネットワーク。これは大規模ネットワークに接続することができる。トークンリング (token ring) も参照。

限定された地域内 (1 つのビルまたは大学構内など) で複数の装置を接続し、より大きなネットワークへ接続可能なネットワーク。

### 論理区画 (LP) (logical partition (LP))

オペレーティング・システムをサポートするために定義されたプロセッサ・ハードウェアのサブセット。論理分割モード (logically partitioned mode) も参照。

### 論理トークン (LTOK) (logical token (LTOK))

IODF 内のオブジェクトのリソース番号。

### 論理装置 (LU) (logical unit (LU))

SNA において、エンド・ユーザーが SNA ネットワークおよびシステム・サービス制御点 (SSCP) によって提供される機能にアクセスする際に使用するポートのこと。1 つの LU は、少なくとも 2 つのセッション (1 つは SSCP、もう 1 つは別の LU) をサポートできる。他の複数の LU との多数のセッションをサポート可能な場合もある。物理装置 (physical unit) および システム・サービス制御点 (system services control point) も参照。

## 論理装置タイプ 6.2 (LU 6.2) ((logical unit 6.2 (LU 6.2))

分散処理環境において、プログラム間の一般的な通信をサポートする論理装置のタイプの1つ。LU 6.2には以下のような特徴がある。

- セッション・パートナー間の対等関係
- 複数トランザクション用のセッションの効率的な使用
- 包括的な終端間エラー処理
- 製品実装環境にマップされる、構造化された動詞からなる汎用アプリケーション・プログラム・インターフェース (API)

拡張プログラム間通信機能 (advanced program-to-program communication) と同義。

## 論理分割 (LPAR) モード (logically partitioned (LPAR) mode)

オペレーターが複数の論理区画間でシステム・プロセッサ・ハードウェア・リソースを割り振ることができる中央処理装置モード。基本モード (basic mode) と対比。

## LOGR

シスプレックス・ロガー (sysplex logger)。

## LP

論理区画 (logical partition) を参照。

## LPAR

論理分割モード (logically partitioned mode) を参照。

## LU

論理装置 (logical unit) を参照。

## LU 6.2

論理装置 6.2 (logical unit 6.2) を参照。

## LU 6.2 セッション (LU 6.2 session)

LU 6.2 アプリケーション・プログラムのために VTAM により開始されるセッション、または APPCCMD マクロを使用することにより、VTAM がセッションを制御するように、アプリケーション・プログラムが指定しているリモート LU によって開始されるセッション。論理装置 6.2 (logical unit 6.2) を参照。

## LU-LU セッション (LU-LU session)

SNA において、SNA ネットワークの2つの論理装置 (LU) 間のセッションのこと。2つのエンド・ユーザー間、またはエンド・ユーザーと LU サービス・コンポーネント間の通信を提供する。

## M

## MAT

NetView 自動化テーブル (NetView automation table) の非推奨用語。

## MCA

マイクロチャネル・アーキテクチャー (Micro Channel architecture) を参照。

## MCS

複数コンソール・サポート (multiple console support) を参照。

## メンバー (member)

XCF に対して定義され、多重システム・アプリケーションによってグループに割り当てられた、多重システム・アプリケーションの特定の機能 (1つ以上のモジュールまたはルーチン)。メンバーは、シスプレックスの1つのシステム上に存在し、XCF サービスを使用して、同じグループの他のメンバーと通信 (データの送受信) できる。

## メッセージ自動化テーブル (MAT) (message automation table (MAT))

NetView 自動化テーブル (NetView automation table) の非推奨用語。

## メッセージ・クラス (message class)

SA z/OS がメッセージのルーティングを制御するためにメッセージと関連付ける番号。自動化された操作中に、SA z/OS が発行する各メッセージに関連付けられたクラスは、各通知オペレーターに割り当てられたクラスと比較されます。メッセージ・クラスの1つと一致するクラスを持つオペレーターが、そのメッセージを受信します。

### メッセージ転送 (message forwarding)

SA z/OS ターゲット・システムで生成されたメッセージを、SA z/OS フォーカル・ポイント・システムに送信する、SA z/OS のプロセス。

### メッセージ・グループ (message group)

1 単位として一緒に表示される複数のメッセージ。

### メッセージ・モニター・タスク (message monitor task)

開始され、通信タスクの数と関連したタスク。メッセージ・モニター・タスクは、通信タスクからインバウンド・メッセージを受信し、開始ターゲット・システムを判別して、そのメッセージを適切なターゲット制御タスクにルーティングする。

### メッセージ処理機能 (message processing facility (MPF))

z/OS コンソールに送信するすべてのメッセージを選別する z/OS テーブルの 1 つ。MPF は、これらのメッセージと、メッセージのユーザー定義リスト (このメッセージ・リストに基づいてメッセージが z/OS コンソール・ディスプレイで自動化/抑止される) を比較し、自動化または抑制すべきメッセージにマークを付ける。その後でメッセージはサブシステム・インターフェース (SSI) 上にブロードキャストされる。

### メッセージ抑止 (message suppression)

z/OS コンソールに表示するメッセージ・トラフィックの量を制限する機能。

### マイクロチャネル・アーキテクチャー (Micro Channel architecture)

コンピューター内のマイクロチャネル・バスをどのようにサブシステムやアダプターに使用させるかを定義した規則。このアーキテクチャーでは、各サブシステムが提供できるかまたは提供する必要のあるサービスを定義する。

### マイクロプロセッサ (microprocessor)

1 つまたは少数のチップに実装されたプロセッサ。

### マイグレーション (migration)

プログラムの旧バージョンまたは旧リリースを置換するために、プログラムの新規バージョンまたは新規リリースをインストールすること。

### MP

マルチプロセッサ (Multiprocessor)。

### MPF

メッセージ処理機能 (message processing facility) を参照。

### MPFLSTxx

SA z/OS によってビルドされる MPFLST メンバー。

### 複数 MVS 環境 (multi-MVS environment)

複数の MVS イメージを稼働可能な物理処理システム。MVS イメージ (MVS image) も参照。

### 複数コンソール・サポート (MCS) (multiple console support (MCS))

複数のコンソールに対してメッセージ経路選択指定を可能にする MVS の機能。

### 多重仮想記憶 (MVS) (Multiple Virtual Storage (MVS))

仮想記憶内の複数のアドレス・スペースにアクセスする IBM オペレーティング・システム。z/OS の前身です。

### マルチプロセッサ (MP)(multiprocessor (MP))

2 つのオペレーティング・プロセッサ複合体を形成するために、物理的に区画できる CPC のこと。

### 複数システム・アプリケーション (multisystem application)

複数システム環境内の複数の z/OS イメージにまたがって、さまざまな機能を分散させているアプリケーション・プログラム。

### 多重システム環境 (multisystem environment)

複数のシステムが 1 つ以上のプロセッサ上で稼働している環境。あるいは、1 つ以上のプロセッサが、他のシステム上のプログラムと通信可能な環境。

### MVS

多重仮想記憶 (Multiple Virtual Storage) を参照。

### MVS イメージ (MVS image)

作業を処理する機能を持つ MVS オペレーティング・システムの単一オカレンス。複数 MVS 環境 (multi-MVS environment) および 単一 MVS 環境 (single-MVS environment) も参照。

## MVS/ESA

多重仮想記憶/エンタープライズ・システム体系 (Multiple Virtual Storage/Enterprise Systems Architecture)。z/OS を参照。

## MVS/JES2

多重仮想記憶/ジョブ入力システム 2。z/OS のサブシステムの 1 つ。システムに各ジョブを受け入れ、内部形式に変換し、選択して実行し、出力を処理し、その後でそれらをシステムからパーズする。プロセッサが複数あるインストール先では、各 JES2 プロセッサがそのジョブ入力、スケジューリング、および出力処理を独立して制御する。

## N

### NAU

ネットワーク・アドレス可能単位 (network addressable unit) を参照。

ネットワーク・アクセス可能単位 (network accessible unit) を参照。

### NCCF

ネットワーク通信制御機能 (Network Communications Control Facility) を参照。

### NCP

ネットワーク制御プログラム (network control program) (一般用語) を参照。

ネットワーク制御プログラム (Network Control Program) (IBM ライセンス・プログラム) を参照。そのフルネームは、ネットワーク制御プログラムの拡張通信機能である。ACF/NCP と同義。

### NCP/トークンリング相互接続 (NCP/token ring interconnection)

トークンリング接続 SNA 装置をサポートするために ACF/NCP で使用される機能。NTRI は、トークンリング接続 SNA 装置 (PU) から交換 (ダイヤルアップ) 装置への変換機能も備えている。

### NetView

ネットワークのモニター、その管理、およびネットワーク問題の診断に使用される IBM ライセンス・プログラムの 1 つ。NetView は、表示サービスを含むコマンド機能、コマンド・プロセッサ、コマンド・リストに基づく自動化、およびトランザクション処理構造 (セッション・モニター、ハードウェア・モニター、および端末アクセス機能 (TAF) ネットワーク管理アプリケーションが構築されている) から構成されます。

### NetView (NCCF) コンソール (NetView (NCCF) console)

NetView コマンドの場合は 3270 型コンソール、またシステム操作とプロセッサ操作の場合は実行時のパネル。

### NetView 自動化プロシージャ

NetView のコマンド・リストまたは高水準言語で書かれたコマンド・プロセッサとしてパッケージされた、一連のコマンド。自動化プロシージャは自動化機能を実行し、NetView プログラムのもとで稼働する。

### NetView 自動化テーブル (AT) (NetView automation table (AT))

NetView プログラムが着信メッセージと比較するテーブル。テーブルの項目と一致すると、指定された応答が起動されます。NetView 自動化テーブルに SA z/OS の項目があると、ターゲット・システム条件に対して SA z/OS 応答を起動する。以前は、メッセージ自動化テーブル (MAT) と呼ばれていた。

### NetView コマンド・リスト言語 (NetView command list language)

コマンド・リストの作成に使用される、NetView に固有な解釈言語。

### NetView ハードウェア・モニター (NetView hardware monitor)

対話式表示技法を使用して、中央制御点からハードウェア、ソフトウェア、およびマイクロコードなどのネットワークの問題を識別するのに役立つ NetView のコンポーネント。以前、ネットワーク問題判別プログラム (network problem determination application) と呼ばれていたもの。

### NetView ログ (NetView log)

NetView がその関連イベント、および NetView での SA z/OS アクティビティを記録するログ。

### NetView メッセージ・テーブル (NetView message table)

NetView 自動化テーブル (NetView automation table) を参照。

### NetView パス (論理装置 LU 6.2 経由) (NetView paths via logical unit (LU 6.2))

エンド・ユーザーが、SNA ネットワーク・リソースに対するアクセス権を獲得および互いに通信できるようにする、ネットワーク・アクセス可能ポート (VTAM 接続) のタイプの 1 つ。LU 6.2 により、プ



ロッセッサ操作とワークステーションとの間の通信が許可される。[論理装置 6.2 \(logical unit 6.2\)](#) を参照。

#### **NetView 間タスク (NNT) (NetView-NetView task(NNT))**

クロスドメイン NetView オペレーター・セッションが実行されるタスク。各 NetView プログラムは 1 つの NNT セッションを確立するために NetView 間タスクを 1 つ持たなければならない。[オペレーター端末タスク \(operator station task\)](#) も参照。

#### **NetView 間タスク・セッション (NetView-NetView task session)**

NetView 間タスクのもとで実行される 2 つの NetView プログラム間のセッション。SA z/OS では、NetView 間タスク・セッションはフォーカル・ポイントとリモート・システム間の通信に使用される。

#### **ネットワーク (network)**

相互接続されたノードのグループ。

データ処理では、ユーザー・アプリケーション・ネットワーク。[SNA ネットワーク \(SNA network\)](#) を参照。

#### **ネットワーク・アクセス可能単位 (NAU) (network accessible unit (NAU))**

SNA ネットワーキングにおいて、論理装置 (LU)、物理装置 (PU)、制御点 (CP)、またはシステム・サービス制御点 (SSCP) を含む、ネットワーク・アドレスを持つ装置。これは、パス制御ネットワークによって伝送された情報の起点または宛先である。[ネットワーク・アドレス可能単位 \(network addressable unit\)](#) と同義。

#### **ネットワーク・アドレス可能単位 (NAU)(network addressable unit (NAU))**

[ネットワーク・アクセス可能単位 \(network accessible unit\)](#) と同義。

#### **ネットワーク通信管理機能 (NCCF) (Network Communications Control Facility (NCCF))**

ネットワークの操作制御機能。NCCF は、表示サービス、コマンド・プロセッサ、コマンド・リストに基づく自動化、およびネットワーク管理アプリケーション NLDM が構築されているトランザクション処理構造からなる。NCCF は、NetView コマンド機能の前身である。

#### **ネットワーク制御プログラム (NCP) (Network Control Program (NCP))**

IBM ライセンス・プログラムの 1 つで、単一ドメイン、マルチドメイン、および相互接続ネットワーク機能の通信コントローラー・サポートを提供する。そのフルネームは、ネットワーク制御プログラムの拡張通信機能である。

#### **ネットワーク制御プログラム (NCP) (Network Control Program (NCP))**

通信コントローラーの稼働を制御するプログラム。

データ・フロー制御のためにネットワーク内の物理装置間で要求および応答を交換するために使用されるプログラム。

#### **ネットワークング NetView (Networking NetView)**

SA z/OS では、ネットワークの構成の管理などのネットワーク管理機能を実行する、NetView。SA z/OS では、ネットワークング NetView にもアラートを送るのが一般的である。

#### **NIP**

[中核初期設定プログラム \(nucleus initialization program\)](#) を参照。

#### **NNT**

[NetView 間タスク \(NetView-NetView task\)](#) を参照。

#### **通知メッセージ (notification message)**

重要な自動化アクションについての情報を提供するため、人間の通知オペレーターに送信される SA z/OS メッセージ。通知メッセージは、カスタマイズ・ダイアログを使用して定義される。

#### **通知オペレーター (notification operator)**

SA z/OS 通知メッセージの受信を許可されている NetView コンソール・オペレーター。許可は、カスタマイズ・ダイアログを介して行われる。

#### **NTRI**

[NCP/トークンリング相互接続 \(NCP/token ring interconnection\)](#) を参照。

#### **中核初期設定プログラム (NIP) (nucleus initialization program (NIP))**

常駐制御プログラムを初期設定するプログラム。これにより、オペレーターは、システムの生成中に指定された特定のオプションに対し最終変更を行うよう要求することができる。

#### **O**

## 目標値 (objective value)

SA z/OS が過去のサービス・データからアプリケーション用に計算できる 平均の Workflow または Using の値。SA z/OS は、警告しきい値およびアラートしきい値が明示的に定義されていない場合に、目標値を使用してこれらのしきい値を計算する。

## OCA

SA z/OS では、ターゲット・システムのアクティブなオペレーター・コンソールである、オペレーター・コンソール A (operator console A) のこと。OCB と対比。

## OCB

SA z/OS では、ターゲット・システムのバックアップ・オペレーター・コンソールである、オペレーター・コンソール B (operator console B) のこと。OCA と対比。

## OPC/A

運用計画管理プログラム/拡張版 (Operations Planning and Control/Advanced) を参照。

## OPC/ESA

運用計画管理プログラム/エンタープライズ・システム・アーキテクチャー (Operations Planning and Control/Enterprise Systems Architecture) を参照。

## オペレーティング・システム (OS) (operating system (OS))

プログラムの実行を制御し、リソース割り振り、スケジューリング、入出力制御、およびデータ管理などのサービスを提供できるソフトウェア。オペレーティング・システムは基本的にソフトウェアであるが、部分的なハードウェア実装も可能である。(T)

## 操作 (operations)

ハードウェア装置またはソフトウェア機能のリアルタイム制御。

## 運用計画管理プログラム/拡張版 (OPC/A) (Operations Planning and Control/Advanced (OPC/A))

バッチ・ワークロードを自動化、計画、管理する、1組の IBM ライセンス・プログラム。OPC/A は、システムおよび作業負荷を分析し、それに従ってジョブを実行依頼する。

## 運用計画管理プログラム/エンタープライズ・システム・アーキテクチャー (OPC/ESA) (Operations Planning and Control/Enterprise Systems Architecture (OPC/ESA))

バッチ・ワークロードを自動化、計画、管理する、1組の IBM ライセンス・プログラム。OPC/ESA は、システムおよび作業負荷を分析し、それに従ってジョブを実行依頼する。OPC/A の後継ソフトウェア。

## オペレーター (operator)

システムの実行を保持する担当者。

z/OS、NetView プログラム、または IMS などのソフトウェアのある部分によって制御される活動を管理する責任のある担当者またはプログラム。

装置を操作する担当者。

言語ステートメントでは、オペラントで実行するアクションを指示する字句エンティティ。

## オペレーター・コンソール (operator console)

コンピューター・オペレーターとコンピューター間のやりとりに使用される装置を含む機能単位の 1 つ。(T)

オペレーターとシステムとの間のコミュニケーション用に使用される ディスプレイ・コンソールで、アプリケーション・プログラムに関する情報を指定するため、およびシステム操作機能をモニターするために、主として使用される。

SA z/OS では、オペレーティング・システム (z/OS、LINUX、VM、VSE) からの出力を表示したり、それらに入力を送信したりするコンソール。オペレーティング・システム・コンソールとも呼ばれます。SA z/OS オペレーター・コマンドおよび構成ダイアログでは、OC はターゲット・システム・オペレーター・コンソールの指定に使用されます。

## オペレーター端末タスク (operator station task (OST))

ネットワーク・オペレーターとのオンライン・セッションを確立および保守する NetView タスク。

NetView プログラムにログオンするネットワーク・オペレーターごとに、1つのオペレーター端末タスクがある。

## オペレーター・ビュー (operator view)

モニターの目的で一緒に関連付けられたグループ、システム、およびリソース定義の集合。オペレーター・ビューは、グラフィック・ディスプレイとしてグラフィカル・インターフェースに表示され、定義されたグループ、システム、およびリソースの状況を示します。

## OperatorView 項目 (OperatorView entry)

カスタマイズ・ダイアログによって作成され、オペレーター・ビューのポリシーを表し、それを入れておくために使用される構成。

## オブティマイザー (optimizer)

汎用のプロセッサに比べて、最適化されたパフォーマンスで、一連の限られた特定の機能を実行できる特殊用途のハードウェア・コンポーネントまたはアプライアンス。オブティマイザーは、限られた1セットの機能であるため、スタンドアロンの装置ではなく、処理環境の一部を統合化したものである。

## OS

[オペレーティング・システム \(operating system\)](#) を参照。

## OST

[オペレーター端末タスク \(operator station task\)](#) を参照。

## アウトバウンド (outbound)

SA z/OS で、フォーカル・ポイント・システムからターゲット・システムに送られるメッセージまたはコマンド。

## アウトバウンド・ゲートウェイ・オペレーター (outbound gateway operator)

他のシステムとの接続を確立する自動化オペレーター。アウトバウンド・ゲートウェイ・オペレーターは、ゲートウェイ・セッションを介して他のシステムとのコミュニケーションを処理する。自動化オペレーターは、受信システムでインバウンド・ゲートウェイ・オペレーターへメッセージ、コマンド、および応答を送信する。

## P

### ページ (page)

表示面に一度に表示されるパネルの部分。

実記憶装置と外部ページまたは補助記憶装置との間で命令、データ、またはその両方を転送すること。

### パネル (panel)

端末装置の画面に現れる情報の定様式表示画面。パネルは、モノスペース・フォント、限定カラー、およびグラフィックスを備えたフルスクリーンの 3270 型ディスプレイです。

SA z/OS パネルを使用して、状況を確認し、キーボードを使用してコマンド行にコマンドを入力し、システムを構成し、他のコンソールへのパススルーを行うことが可能。[ヘルプ・パネル \(help panel\)](#) も参照。

コンピューター・グラフィックスでは、表示面での表示フィールドの位置 および特性を定義する表示イメージ。[画面 \(screen\)](#) と対比。

### パラメーター (parameter)

例えば、指定されたアプリケーションに定数値を与え、アプリケーションを表すことができる変数。

ユーザーが値を指定するか、またはメニューが解釈されるときに、システムが値を提供するメニュー内のエンタリー。

ユーザーまたは他のプログラムによってプログラムまたはプロシージャに渡されるデータ。具体的には、言語ステートメントではオペランドとして、メニューでは項目として、または共有データ構造として渡されるデータ。

### 区画 (partition)

ストレージの固定サイズの分割部分。

VSE では、プログラム処理用に使用可能な仮想アドレス域の分割部分。

IBM パーソナル・コンピューターのハード・ディスク上の、可変サイズの 4 つの使用可能なストレージの 1 つ。1 つは DOS でアクセス可能であるが、他はそれぞれ、別のオペレーティング・システムに割り当てることができる。

### 分割可能 CPC (partitionable CPC)

2 つの独立した CPC に分割できる CPC。[物理分割 \(physical partition\)](#)、[単一イメージ・モード \(single-image mode\)](#)、MP、[サイド \(side\)](#) も参照。

### 区分データ・セット (PDS) (partitioned data set (PDS))

メンバーと呼ばれる、区画に分割された直接アクセス記憶装置のデータ・セットで、それぞれにプログラム、プログラムの一部、またはデータを含めることができる。



### 受動的モニタリング (passive monitoring)

SA z/OS では、z/OS システムおよびそのリソースから非送信請求メッセージを受信すること。これらのメッセージは、リソース状況の表示を更新するようにプロンプト指示することができる。[アクティブ・モニター \(active monitoring\)](#) も参照

### PCE

プロセッサ・コントローラー。一部のプロセッサ・ファミリーではサポート・プロセッサまたはサービス・プロセッサとしても知られている。

### PDB

[ポリシー・データベース \(policy database\)](#) を参照。

### PDS

[区分データ・セット \(partitioned data set\)](#) を参照。

### 物理分割 (physical partition)

それ自体で CPC として稼働する CPC の一部。専用のオペレーティング・システムのコピーを持つ。

### 物理装置 (PU) (physical unit (PU))

SNA では、SSCP-PU セッションを通してシステム・サービス制御点 (SSCP) によって要求されたとおりに、ノードのリソース (接続されたリンク および隣接リンク・ステーションなど) を管理およびモニターするコンポーネント。SSCP は、物理装置のあるセッションを活動化して、PU を通して接続されたリンクなどのノードのリソースを間接的に管理する。

### 物理分割 (PP) 構成 (physically partitioned (PP) configuration)

1つのマルチプロセッサ (MP) システムが、個別の電源、ユーティリティ、保守境界を持つ2つ以上の独立 CPC として機能することができる動作モード。[単一イメージ・モード \(single-image mode\)](#) と対比。

### PLEXID グループ (PLEXID group)

PLEXID グループまたは「拡張 XCF 通信グループ」は、シスプレックスに関連して使用される用語。PLEXID グループには、シスプレックスのサブセットまたはシスプレックス全体の System Automation エージェントが含まれる。SAplex 境界を越える XCF 通信を提供するために使用される。詳しくは、「*IBM Z System Automation* 計画とインストール」の『[拡張 XCF 通信グループの定義](#)』を参照してください。

### POI

[プログラム・オペレーター・インターフェース \(program operator interface\)](#) を参照。

### ポリシー (policy)

SA z/OS エンタープライズの自動化およびモニターに関する指定。「*IBM Z System Automation* 自動化ポリシーの定義」を参照。

### ポリシー・データベース (policy database)

自動化管理者がカスタマイズ・ダイアログを使用して指定する自動化定義 (自動化ポリシー) はポリシー・データベースに保管される。PDB とも呼ばれる。[自動化ポリシー \(automation policy\)](#) も参照。

### POR

[パワーオン・リセット \(power-on reset\)](#) を参照。

### ポート (port)

入出力装置が接続されるシステム・ハードウェア。

データの入り口または出口のアクセス・ポイント (例えば、論理装置)。

データがデータ・ネットワークに入るため、またはデータ・ネットワークから出るために経由する、ノードの機能単位。

データ通信において、1つ以上の外部リモート装置に対してデータの送受信を行うために単一データ・チャンネル専用となるデータ・プロセッサ部分。

### パワーオン・リセット (POR) (power-on reset (POR))

CPC のすべてのハードウェアを再初期設定し、CPC がオペレーティング・システムをロードおよび実行できるようにする内部コードをロードする機能。[初期マイクロプログラム・ロード \(initial microprogram load\)](#) を参照。

### PP

[物理区画 \(physical partition\)](#) を参照。

## PPI

プログラム間インターフェース (program to program interface) を参照。

## PPT

一次 POI タスク (primary POI task) を参照。

## PR/SM

プロセッサ・リソース/システム管理機構 (Processor Resource/Systems Manager) を参照。

### 1 次ホスト (primary host)

処理用にコマンドを入力できる基本プログラム。

### 1 次 POI タスク (PPT) (primary POI task (PPT))

VTAM プログラム・オペレーター・インターフェース (POI) から受信したすべての非送信請求メッセージを処理し、それらを制御オペレーターまたはコマンド・プロセッサに送達する NetView サブタスク。PPT はまた、NetView が初期設定されたときに実行するように指定された初期コマンド、および PPT の下で実行するようにスケジュールされたタイマー要求コマンドも処理する。

### 1 次システム (primary system)

アプリケーションを通常そのシステムで実行する場合には、そのシステムがそのアプリケーションの 1 次システムとなる。SA z/OS は、アプリケーションに定義されたすべての 1 次システム上でアプリケーションを開始する。

### 問題判別 (problem determination)

問題のソースを判別するプロセス。例えば、プログラム・コンポーネント、機械故障、通信機能、ユーザーまたは請負業者がインストールしたプログラムまたは装置、停電などの環境障害、またはユーザー・エラーなど。

### プロセッサ (processor)

プログラムされた命令からのデータを処理する装置。他の装置の一部である場合がある。

コンピューター内で、命令を解釈および実行する部分。プロセッサの代表的な構成要素には、制御装置と演算論理装置の 2 つがある。

### プロセッサ・コントローラー (processor controller)

中央処理装置用のサポートおよび診断機能を提供するハードウェア。

### プロセッサ操作

プロセッサ (ハードウェア) 操作をモニターおよび制御する SA z/OS の一部。プロセッサ操作は、フォーカル・ポイント・システムからターゲット・システムへの接続を提供する。プロセッサ操作は、フォーカル・ポイント・システムの NetView を介して、ターゲット・システムのモニターやリカバリーを行うためにオペレーター・コンソールおよびシステム・コンソールを自動化する。ProcOps としても知られている。

### Processor Resource/Systems Manager (PR/SM)

プロセッサがいくつかのオペレーティング・システム・イメージを同時に使用できるようにし、論理区画機能を提供する機能。論理分割モード (logically partitioned mode) も参照。

### ProcOps

プロセッサ操作 (processor operations) を参照。

### ProcOps サービス・マシン (ProcOps Service Machine) (PSM)

PSM は、VM ホスト・システム上の CMS ユーザーである。ProcOps の「仮想ハードウェア」として機能する CMS マルチタスキング・アプリケーションを実行する。ProcOps は PSM を経由して、ProcOps 内のターゲット・システムとして定義された VM ゲスト・システムと通信する。

### プロダクト自動化 (product automation)

CICS、Db2、IMS、IBM Workload Scheduler の各プロダクト用に SA z/OS のベースに組み込まれた自動化 (以前はフィーチャーと呼ばれたもの)。

### プログラム・オペレーター・インターフェース (POI) (program operator interface (POI))

VTAM メッセージを受信するための NetView 機能。

### プログラム間インターフェース (PPI) (program to program interface (PPI))

ユーザー・プログラムが他のユーザー・プログラムからデータ・バッファーを送受信でき、システムおよびアプリケーション・プログラムから NetView ハードウェア・モニターへアラートを送信できるようにする NetView 機能の 1 つ。

## プロトコル (protocol)

SNA では、ネットワークの管理、データの転送、およびネットワーク・コンポーネントの状態の同期化に使用される要求と応答の意味、および順序付け規則。

## プロキシ・リソース (proxy resource)

プロセッサ操作ターゲット・システムを表す項目タイプ APL と同様に定義されたリソース。

## PSM

ProcOps サービス・マシン (ProcOps Service Machine) を参照。

## PU

物理装置 (physical unit) を参照。

## R

### RACF

リソース・アクセス制御機能 (Resource Access Control Facility) を参照。

### リモート・システム (remote system)

SA z/OS フォーカル・ポイント・システムからリソースの状況情報を受信するシステム。SA z/OS リモート・システムは、それが関連している SA z/OS フォーカル・ポイント・システムと同一の SA z/OS エンタープライズの一部として定義される。

### リクエスター (requester)

ワークステーションの 1 つ。ユーザーはそこからドメイン (ドメインに属するサーバー) にログオンし、ネットワーク・リソースを使用できる。ユーザーは共有リソースにアクセスし、サーバーの処理能力を利用することで、ハードウェアへの投資を削減できる。

### リソース (resource)

主記憶域、入出力装置、処理装置、データ・セット、および制御または処理プログラムを含む、ジョブまたはタスクに必要なコンピューター・システムまたはオペレーティング・システムの機能。

NetView では、ネットワークに機能を提供するハードウェアまたはソフトウェア。

SA z/OS において、SA z/OS を使用してモニターまたは自動化できる、任意の z/OS アプリケーション、z/OS コンポーネント、ジョブ、装置、またはターゲット・システム。

### リソース・アクセス管理機能 (RACF) (Resource Access Control Facility (RACF))

ユーザーのすべてのリソースにデータ・セキュリティを提供できるプログラム。RACF は、事故または故意による、許可されない開示、修正、または消滅からデータを保護する。

### リソース・グループ (resource group)

プロセッサの物理的な分離可能部分。サイド (side) としても知られている。

### リソース測定機能 (RMF) (Resource Measurement Facility (RMF))

選択された領域のシステム・アクティビティを測定し、収集したデータを印刷されたレポート、システム管理機能 (SMF) レコード、または表示レポートの形式で提示する z/OS の機能。

### 再始動の自動化 (restart automation)

実行中であることを確認するためにサブシステムをモニターする、SA z/OS が提供する自動化。サブシステムに障害が起こると、SA z/OS は自動化構成ファイル内のポリシーに従ってサブシステムを再始動しようとする。

### 再構造化拡張実行プログラム (REXX) (Restructured Extended Executor (REXX))

汎用の高水準プログラミング言語。特にパーソナル・コンピューティング用の EXEC プロシージャまたはプログラムに適しており、コマンド・リストの作成に使用される。

### 戻りコード (return code)

後続の命令の実行に影響を与えるために使用される、プログラムから戻されるコード。

## REXX

再構造化拡張実行プログラム (Restructured Extended Executor) を参照。

### REXX プロシージャ (REXX procedure)

インタプリタ言語である再構造化拡張実行プログラム (REXX) で書き込まれたコマンド・リスト。

## RMF

リソース測定機能 (Resource Measurement Facility) を参照。

## S

## SAF

[セキュリティ許可機能 \(Security Authorization Facility\)](#) を参照。

## SA IOM

[System Automation for Integrated Operations Management](#) を参照。

## SAplex

SAplex または「SA z/OS Subplex」は、シスプレックスに関連して使用される用語。実際には、SAplex はシスプレックスのサブセットである。ただし、SAplex 自体がシスプレックスになることもできる。詳しくは、「*IBM Z System Automation 計画とインストール*」の『SA z/OS サブプレックスの使用』を参照。

## SA z/OS

287 ページの『*IBM Z System Automation*』を参照。

## SA z/OS カスタマイズ・ダイアログ

SA z/OS ポリシー管理者が各 z/OS システムのポリシーを定義し、自動化制御データを作成する際に使用する ISPF アプリケーションの 1 つ。

## SA z/OS カスタマイズ・フォーカル・ポイント・システム (SA z/OS customization focal point system)

[フォーカル・ポイント・システム \(focal point system\)](#) を参照。

## SA z/OS データ・モデル (SA z/OS data model)

SA z/OS の機能および NetView 自動化プラットフォームをサポートするのに必要なオブジェクト、クラス、およびエンティティ・リレーションシップ (ER) のセット。

## SA z/OS エンタープライズ

1 つのエンタープライズ名の下でカスタマイズ・ダイアログに定義された システムおよびリソースのグループ。SA z/OS エンタープライズは、SA z/OS を実行する、相互接続された z/OS システムで構成されます。

## SA z/OS フォーカル・ポイント・システム

[フォーカル・ポイント・システム \(focal point system\)](#) を参照。

## SA z/OS ポリシー (SA z/OS policy)

SA z/OS エンタープライズを構成するシステムおよびリソースと、それらのモニターおよび自動化定義の記述。

## SA z/OS ポリシー管理者 (SA z/OS policy administrator)

SA z/OS ポリシーの定義を担当する操作スタッフのメンバー。

## SA z/OS SDF フォーカル・ポイント・システム

[フォーカル・ポイント・システム \(focal point system\)](#) を参照。

## SCA

SA z/OS では、ターゲット・ハードウェアのアクティブなシステム・コンソールである、システム・コンソール A (system console A) のこと。SCB と対比。

## SCB

SA z/OS では、ターゲット・ハードウェアのバックアップ・システム・コンソールである、システム・コンソール B (system console B) のこと。SCA と対比。

## 画面 (screen)

[パネル \(panel\)](#) の非推奨用語。

## スクリーン・ハンドラー (screen handler)

SA z/OS で、ターゲット・システムのフルスクリーン・イメージとの間でやりとりされるすべてのデータを解釈するソフトウェア。解釈は、フルスクリーン・イメージのデータの形式によって異なる。すべてのプロセッサおよびオペレーティング・システムはフルスクリーン・イメージの独自の形式を持っている。スクリーン・ハンドラーはターゲット・システムに対して 1 つの PS/2 接続を制御する。

## SDF

[状況表示機能 \(status display facility\)](#) を参照。

## SDLC

[同期データ・リンク制御 \(synchronous data link control\)](#) を参照。

## SDSF

[システム表示および検索機能 \(System Display and Search Facility\)](#) を参照。

## 2次システム (secondary system)

アプリケーションがシステム上で自動化するように定義されていても、そのアプリケーションが通常そこで稼働することになっていなければ、そのシステムはそのアプリケーションの2次システムとなる。2次システムは、1つ以上の1次システムが使用不能な場合にアプリケーションを移動できる先のシステムである。SA z/OSは、その2次システムではアプリケーションを始動しない。

## セキュリティ許可機能 (SAF) (Security Authorization Facility (SAF))

MVS インターフェースの1つ。プログラムはこれを使用して、RACFなどの外部セキュリティ・マネージャーと通信することができる。

## サーバー (server)

サーバーは、リソースを共有するワークステーションであり、ディレクトリー、プリンター、シリアル装置、および計算機能、などが含まれる。

## サービス言語コマンド (SLC)(service language command (SLC))

プロセッサ・コントローラーまたはサービス・プロセッサの行指向のコマンド言語。

## サービス期間 (service period)

サービス期間を用いることによって、ユーザーはアプリケーションの可用性をスケジュールすることができる。サービス期間は、アプリケーションがアクティブになっているべき時間間隔のセット (サービス・ウィンドウ) である。

## サービス・プロセッサ (SVP)(service processor (SVP))

小規模のシステム/370 プロセッサのプロセッサ・コントローラーに与えられた名前。

## サービスしきい値 (service threshold)

リソースに対するサービスの低下をオペレーターにいつ通知するかを決める SA z/OS ポリシー設定値の1つ。アラートしきい値 (alert threshold) および警告しきい値 (warning threshold) も参照。

## セッション (session)

SNAでは、2つのネットワーク・アドレス可能装置 (NAU) 間の論理接続のことで、要求どおりに活動化、各種プロトコルを提供するように調整、および非活動化することができる。各セッションは、セッション中に交換される任意の伝送の発信元および宛先 NAU を識別する1組のネットワーク・アドレスによって、伝送ヘッダー中で固有に識別される。

## セッション・モニター (session monitor)

セッション関連データを収集して相互に関連させ、この情報に対するオンライン・アクセスを提供する、NetView プログラムのコンポーネント。NLDMの後続。

## シャットダウン自動化 (shutdown automation)

シャットダウン・コマンドを実行し、追加情報のプロンプトに応答することによって、サブシステムのシャットダウン・プロセスを管理する、SA z/OS 提供の自動化機能。

## サイド (side)

区画化可能な CPC の一部で、物理区画として実行でき、一般的には A サイドまたは B サイドと呼ばれる。

## Simple Network Management Protocol (SNMP)

複合ネットワーク内のモニタリング・システムおよび装置のためのプロトコルのセット。管理対象装置に関する情報は、管理情報ベース (MIB) に定義および保管される。

## 単一イメージ (single image)

物理的に分割されていない、物理分割が可能なプロセッサ・システム。単一イメージのシステムはターゲット・ハードウェア・プロセッサにできる。

## 単一 MVS 環境 (single-MVS environment)

1つの MVS イメージをサポートする環境。MVS イメージ (MVS image) も参照。

## 単一イメージ (SI) モード (single-image (SI) mode)

1つの CPC として機能できる、マルチプロセッサ (MP) システムの操作モードの1つ。定義によって、ユニプロセッサ (UP) は単一イメージ・モードで作動する。物理分割 (PP) 構成 (physically partitioned (PP) configuration) と対比。

## SLC

サービス言語コマンド (service language command) を参照。

## SMP/E

システム修正変更プログラム拡張版 (System Modification Program/Extended) を参照。



## SNA

システム・ネットワーク体系 (Systems Network Architecture) を参照。

## SNA ネットワーク (SNA network)

SNA では、システム・ネットワーク体系のフォーマットおよびプロトコル に適合させるユーザー・アプリケーション・ネットワークの一部。これにより、エンド・ユーザー間のデータ転送が信頼できるものになり、各種ネットワーク構成のリソースを制御するためのプロトコルが提供される。SNA ネットワークはネットワーク・アドレス可能装置 (NAU)、境界機能コンポーネント、およびパス制御ネットワークからなる。

## SNMP

Simple Network Management Protocol (SNMP) を参照。

## 送信請求メッセージ (solicited message)

コマンドに直接応答する SA z/OS メッセージ。 非送信請求メッセージ (unsolicited message) と対比。

## SSCP

システム・サービス制御点 (system services control point) を参照。

## SSI

サブシステム・インターフェース (subsystem interface) を参照。

## 始動自動化 (start automation)

サブシステムの始動プロセスを管理および実行する、SA z/OS によって提供される自動化。このプロセス中、SA z/OS は、追加情報のプロンプトに応答し、指定された時間制限内に始動プロセスを確実に完了させ、必要であればオペレーターに問題を通知し、サブシステムを稼働 (作動可能) 状態にします。

## 始動 (startup)

サブシステムまたはアプリケーションが開始される時点。

## 状況 (status)

リソースの状態または可用性の程度。

## 状況表示機能 (Status Display Facility) (SDF)

動的に色分けされたパネルで、アプリケーション、ゲートウェイ、およびオペレーター宛メッセージ (WTOR) などのリソースの状況を表示する、SA z/OS のシステム操作機能部分。SDF は、複数システムからのスプール使用法の問題およびリソース・データを表示する。

## 定常状態自動化 (steady state automation)

サブシステム、アプリケーション、ボリューム、およびシステムの表示とパフォーマンスの両方についてのルーチン・モニター。定常状態自動化は、メッセージ、パフォーマンスの例外、およびシステムのモデルと現実との矛盾に 応答することがある。

## ストラクチャー (structure)

カップリング・ファシリティーのストレージをマップおよび管理するために、z/OS が使用する構成。

## サブグループ (subgroup)

システムの名前の付いた集合。サブグループは SA z/OS エンタープライズ定義の一部で、モニター目的で使用される。

## SubGroup 項目 (SubGroup entry)

カスタマイズ・ダイアログによって作成され、サブグループのポリシーを表し、それを入れておくために使用される構成。

## サブプレックス (subplex)

SAplex を参照。

## サブシステム (subsystem)

2 次システムまたは従属システムのこと、通常、制御システムとは独立して、または非同期で操作可能である。

SA z/OS では、SA z/OS に対して定義された z/OS アプリケーションまたはサブシステム。

## サブシステム・インターフェース (SSI) (subsystem interface (SSI))

z/OS コンソールに送信されるすべてのメッセージのブロードキャストに使用される z/OS インターフェース。

## サポート・エレメント (SE) (support element (SE))

中央演算処理複合システム (CPC) に対してコミュニケーション、モニター、および診断機能を提供するハードウェア装置。

### サポート・プロセッサ (support processor)

小規模のシステム/370 プロセッサのプロセッサ・コントローラに与えられたもう 1 つの名前。  
サービス・プロセッサ (service processor) を参照。

### SVP

サービス・プロセッサ (service processor) を参照。

### 記号宛先名 (SDN) (symbolic destination name (SDN))

VTAM アプリケーション名に関連づけるためにローカルにワークステーションで使用される。

### 同期データ・リンク制御 (SDLC) (Synchronous data link control (SDLC))

リンク接続における、コードを意識させない同期的なシリアル・バイ・ビット情報転送の管理に関する規則。伝送交換は、交換リンクまたは非交換リンクにおいて二重または半二重とすることができる。リンク接続の構成は 2 地点間、マルチポイント、またはループとすることができる。SDLC は、米国規格協会 (ANSI) の Advanced Data Communication Control Procedures (ADCCP) のサブセットと国際標準化機構 (International Standards Organization) のハイレベル・データ・リンク制御 (HDLC) に準拠する。

### SYSINFO レポート (SYSINFO Report)

システムの概説、その作業負荷、およびリソースを使用しているか、またはリソースのために遅れているジョブの合計数を表示する RMF 報告書。

### SysOps

システム操作 (system operations) を参照。

### シスプレックス (sysplex)

特定の複数システム・ハードウェア・コンポーネント (結合装置およびタイマー) とソフトウェア・サービス (結合データ・セット) を使用して、互いに通信および連携する z/OS システムのセット。

シスプレックスでは、z/OS は結合サービスを提供します。このサービスは、接続された複数のプロセッサ、シスプレックス・タイマー、カップリング・ファシリティー、および結合データ・セット (自動化のポリシーと状況を含む) にわたるワークロードを持つ、マルチシステム・アプリケーションの各部分に関するメッセージ、データ、および状況を処理します。

Parallel Sysplex は、カップリング・ファシリティーを含むシスプレックスです。

### シスプレックス・アプリケーション・グループ (sysplex application group)

シスプレックス・アプリケーション・グループは、シスプレックス内の任意のシステム上で実行できるアプリケーションのグループである。

### シスプレックス結合データ・セット (sysplex couple data set)

システム、グループ、および XCF サービスを使用するメンバーについて、シスプレックス全体にわたるデータを含む結合データ・セット。シスプレックス内のすべての z/OS システムは、シスプレックス結合データ・セットへの接続を持つ必要がある。結合データ・セット (couple data set) も参照。

### シスプレックス・タイマー (Sysplex Timer)

複数のプロセッサまたはプロセッサ・サイドの時刻機構 (TOD) を同期化する IBM の装置。外部時刻参照 (ETR) は、IBM Sysplex Timer (9037) の z/OS 総称名。

### システム (system)

SA z/OS では、システムとはフォーカル・ポイント・システム (z/OS) またはターゲット・システム (MVS、VM、VSE、LINUX または CF) を意味する。

### System Automation for Integrated Operations Management

メインフレーム/分散システムに対し、セキュアにリモート・アクセスするための外部自動化ソリューション。Tivoli System Automation for Integrated Operations Management (以前の Tivoli AF/REMOTE) を使用すると、ユーザーは任意のロケーションからメインフレームおよび分散システムを管理できます。

SA IOM の正式名称。

### IBM Z System Automation

SA z/OS の正式名称。

### システム・コンソール (system console)

通常、キーボードおよび表示画面を持つコンソールで、システムの制御およびシステムとの連絡のためにオペレーターによって使用される。



ハードウェア機能 (例えば、IPL、変更表示、および再構成など) の操作 および制御のために使用される論理装置。システム・コンソールは、プロセッサ・コントローラーまたはサポート・プロセッサに接続された物理表示画面に割り当てることができる。

SA z/OS では、SA z/OS を使用して接続されたプロセッサのプロセッサ・コントローラーまたはサービス・プロセッサ用の、ハードウェア・システム・コンソール。SA z/OS オペレーター・コマンドおよび構成ダイアログでは、SC はターゲット・ハードウェア・プロセッサ用のシステム・コンソールを指定するのに使用される。

### システム表示/検索機能 (SDSF) (System Display and Search Facility (SDSF))

IBM ライセンス・プログラムの 1 つで、一連のパネルで、JES2 の下で実行中のジョブ、キュー、およびプリンターについての情報を提供する。SA z/OS のもとで、プルダウン・メニューから SDSF を選択して、リソースの状況の確認、z/OS システム・ログの参照、WTOR メッセージの表示、およびシステム上でアクティブなジョブの確認を行うことができる。

### System 項目 (System entry)

システムのポリシーを表し、これを入れておくために使用される、カスタマイズ・ダイアログで作成される構成。

### システム修正変更プログラム拡張版 (SMP/E) (System Modification Program/Extended (SMP/E))

z/OS システムをインストールし、保守するプロセスを支援する IBM ライセンス・プログラム。

### システム操作

NetView, SDSF, JES, RMF, TSO, ACF/VTAM, CICS, IMS, および OPC などのシステム操作アプリケーションおよびサブシステムをモニターおよび制御する SA z/OS の一部。SysOps としても知られている。

### システム・サービス制御点 (SSCP) (system services control point (SSCP))

SNA では、構成の管理、ネットワーク・オペレーターと問題判別要求の調整、およびディレクトリー・サポートと、ネットワークのエンド・ユーザー用の他のセッション・サービスを提供する、SNA ネットワーク内のフォーカル・ポイント。対等機能として協調処理する複数の SSCP は、そのドメイン内の物理装置と論理装置との階層制御関係を持つそれぞれの SSCP で、ネットワークを制御のドメインに分割できる。

### システム/390 マイクロプロセッサ・クラスター (System/390 microprocessor cluster)

中央処理装置複合システム (CPC) からなり、1 つ以上の統合カップリング・ファシリティーを持つことができる構成。

### システム・ネットワーク体系 (SNA) (Systems Network Architecture (SNA))

ネットワークを介して情報単位を伝送し、ネットワークの構成および操作を管理するための論理構造、フォーマット、プロトコル、および操作シーケンスを記述したもの。

## T

### TAF

端末アクセス機能 (terminal access facility) を参照。

### ターゲット (target)

フォーカル・ポイント・システムによってモニター および制御されるプロセッサまたはシステム。

### ターゲット制御タスク (target control task)

SA z/OS では、ターゲット制御タスクは、コマンドを処理し、通信タスクを介してデータをターゲット・システムおよびワークステーションに送信する。ターゲット制御タスク (NetView 自動開始タスク) は、ターゲット・システムが初期設定されるときにターゲット・システムに割り当てられる。

### ターゲット・ハードウェア (target hardware)

SA z/OS では、ターゲット・システムが稼働する物理的ハードウェアのこと。これは、単一イメージまたは物理区画プロセッサにすることができる。ターゲット・システム (target system) と対比。

### ターゲット・システム (target system)

分散システム環境では、フォーカル・ポイント・システムによってモニター および制御されるシステムのこと。単一のフォーカル・ポイント・システムが複数のターゲット・システムを制御することができる。

SA z/OS では、モニターと制御用にフォーカル・ポイント・システムに接続されたコンピューター・システム。ターゲット・システムの定義には、リモート・セッションの確立方法、使用するハードウェア、使用するオペレーティング・システムが含まれる。

## タスク (task)

コンピューターが行う作業の基本単位。

NetView 環境では、オペレーター端末タスク (ログオン・オペレーター)、自動化オペレーター (自動開始タスク)、アプリケーション・タスク、またはユーザー・タスクのこと。NetView タスクは NetView 環境で作業を実行する。SA z/OS タスクはすべて、NetView タスクである。[メッセージ・モニター・タスク](#)および[ターゲット制御タスク](#)も参照。

## 通信回線 (telecommunication line)

データの伝送に使用される、ワイヤーまたはマイクロウェーブ・ビームなどの物理メディア。

## 端末アクセス機能 (TAF) (terminal access facility (TAF))

NetView 機能の 1 つで、ユーザーのシステム上または他のシステム上の複数のアプリケーションにログオンできるようにする機能。TAF セッションは SA z/OS カスタマイズ・パネルで定義できるため、使用するたびに TAF セッションを設定する必要はない。

NetView では、ネットワーク・オペレーターが多くのサブシステムを制御できるようにする機能。フルスクリーンまたはオペレーター制御セッションでは、オペレーターがサブシステム の組み合わせを同時に制御できる。

## 端末エミュレーション (terminal emulation)

データをアクセスするために、処理装置にリンクされた特定のタイプの端末 装置であるかのように操作する、マイクロコンピューターまたはパーソナル・コンピューターの機能。

## しきい値 (threshold)

SA z/OS 自動化機能が事前定義アクションを実行する時点を決める値。[アラートしきい値 \(alert threshold\)](#)、[警告しきい値 \(warning threshold\)](#)、および[エラーしきい値 \(error threshold\)](#)を参照。

## 時刻 (TOD) (time of day (TOD))

一般的には時刻クロックのこと。

## タイム・シェアリング・オプション (TSO) (Time Sharing Option (TSO))

オペレーティング・システムのオプション構成の 1 つで、リモート・ステーションからの会話型タイム・シェアリングを提供する。これは z/OS、MVS/ESA、および MVS/XA の対話式サービスである。

## タイム・シェアリング・オプション拡張機能 (TSO/E) (Time-Sharing Option/Extended (TSO/E))

リモート端末からの会話型タイム・シェアリングを提供する、z/OS のオプション。TSO/E によって、広範なユーザーが多くの異なる種類のタスクを実行できる。この機能を使用すれば、リソースをあまり消費しない短期実行アプリケーションだけでなく、大量のリソースを必要とする長期実行アプリケーションも扱うことができる。

## タイマー (timers)

指定された時刻または時間間隔でコマンドまたはコマンド・プロセッサ (コマンドのリスト) を実行する NetView 命令。

## TOD

時刻 (Time of day)。

## トークンリング (token ring)

ある接続装置から別の接続装置へトークンを渡すリング・トポロジーを備えた ネットワーク。例えば、IBM トークンリング・ネットワーク・プロダクト。

## TP

[トランザクション・プログラム \(transaction program\)](#) を参照。

## トランザクション・プログラム (transaction program)

VTAM プログラムでは、トランザクションの処理に関連するサービスを実行するプログラム。VTAM アプリケーション・プログラム・インターフェース (API) を使用している VTAM アプリケーション・プログラムの中で、1 つまたはいくつかのトランザクション・プログラムを操作することができる。この状況では、トランザクション・プログラムは、該当のアプリケーション・プログラムによって定義されるプロトコルを用いて、アプリケーション・プログラムからサービスを要求する。アプリケーション・プログラムは、順に APPCCMD マクロ命令を用いて VTAM プログラムからサービスを要求することができる。

## 処置の自動化 (transitional automation)

SA z/OS に対して定義されたサブシステムおよびアプリケーションの開始および停止に関する アクション。これには、コマンドの発行およびメッセージへの応答を含めることができる。

## 変換ホスト (translating host)

単一化プロセス中に、リソース番号をトークンにするようにホストが再生する役割。

## トリガー (trigger)

トリガーはイベントおよびサービス期間と組み合わせて使用され、単一システム または並列シスプレックス内のアプリケーションの開始および停止を制御する。

## TSO

タイム・シェアリング・オプション (Time Sharing Option) を参照。

## TSO コンソール (TSO console)

この 3270 タイプのコンソールから、TSO または ISPF にログオンして、SA z/OS カスタマイズ・パネルの実行時パネルを使用する。

## TSO/E

タイム・シェアリング・オプション拡張機能 (Time-Sharing Option/Extended) を参照。

## TWS

ZWS を参照。

## U

### 非送信請求メッセージ (unsolicited message)

コマンドに対する直接応答でない SA z/OS メッセージ。

## Uniform Resource Identifier (URI)

Uniform Resource Identifier は、Web リソースの名前を識別するための文字ストリングである。このような識別により、特定のプロトコルを使用して、インターネット経由で Web リソース表現との対話が可能になる。

## ユーザー・タスク (user task)

NetView TASK 定義ステートメントで定義された、NetView プログラムのアプリケーション。

## 累積使用 (Using)

RMF モニター III の定義。ハードウェア・リソース (プロセッサ または装置) からサービスを受けるジョブは、これらのリソースを **使用** している。アドレス・スペースによるリソースの使用の範囲は、0% から 100% までである。この場合の 0% の値は、範囲の期間内に使用しなかったことを示す。100% の値は、そのアドレス・スペースについて、その期間内の各サンプルで、リソースが使用されたことを示す。

## V

### ビュー (view)

NetView Graphic Monitor Facility において、ネットワーク またはネットワークの一部を表すグラフィカル表示。ビューは、リンクによって接続されたノードから構成され、テキストと背景行も含めることができる。ビューによってネットワーク・リソースの状況情報を表示、編集、およびモニターすることができる。

## 仮想サーバー (Virtual Server)

特定のアーキテクチャーに準拠しているプロセッサ、メモリー、および I/O リソースを構成しているように認識される論理構造。仮想サーバーはオペレーティング・システム、関連ミドルウェア、およびアプリケーションをサポートできる。ハイパーバイザーが仮想サーバーを作成および管理する。

## 仮想サーバー・コレクション (Virtual Server Collection)

ワークロードをサポートする仮想サーバーの集合体。この集合体は静的とは限らない。ある任意の時点でのコレクションの構成要素は、その時にワークロードのサポートに関わっている仮想サーバーにより決定される。

## 仮想サーバー・イメージ (virtual Server Image)

システム要件、仮想ストレージ・ドライブ、および仮想マシンに対する任意のゴールと制約 (例えば独立性と可用性) を記述するメタデータが含まれたパッケージ。Open Virtual Machine Format (OVF) は、仮想サーバー・イメージ用にパッケージ化するためのフォーマットを記述する Distributed Management Task Force (DMTF) 標準である。

## 仮想サーバー・イメージ・キャプチャー (Virtual Server Image Capture)

既存の仮想サーバーのメタデータおよびディスク・イメージを保管するための機能。メタデータは仮想サーバーのストレージ、ネットワーク・ニーズ、ゴール、および制限を記述する。キャプチャーされ

た情報は仮想サーバー・イメージとして保管され、他の類似イメージを作成してデプロイするために参照および使用ができる。

### **仮想サーバー・イメージ・クローン (Virtual Server Image Clone)**

仮想サーバー・イメージと同一のコピー (クローン) を作成する機能。この機能を使用して、新しい仮想サーバーを作成できる。

### **拡張仮想記憶 (VSE) (Virtual Storage Extended (VSE))**

基本オペレーティング・システム (VSE/拡張機能) と、ユーザーのデータ処理ニーズを満たすために必要な IBM 提供およびユーザー作成のプログラムから構成されるシステム。VSE および VSE が制御するハードウェアにより、完全なコンピューティング・システムが形成される。現行バージョンは VSE/ESA と呼ばれる。

### **仮想記憶通信アクセス方式 (VTAM) (Virtual Telecommunications Access Method (VTAM))**

SNA ネットワークでの通信およびデータのフローを制御する IBM ライセンス・プログラムの 1 つ。これは単一ドメイン、マルチドメイン、および相互接続ネットワーク機能を提供する。正式な製品名は、仮想記憶通信アクセス方式拡張通信機能である。[ACF/VTAM](#) と同義。

### **VM 第 2 レベル・システム・サポート (VM Second Level Systems Support)**

この機能によって、プロセッサ操作は、実ハードウェア上で実行中のシステムを制御するのと同じ方法で、VM 第 2 レベル・システム (VM ゲスト・システム) を制御できる。

### **VM/ESA**

VM/エンタープライズ・システム体系 (Virtual Machine/Enterprise Systems Architecture)。現行バージョンは z/VM と呼ばれる。

### **ボリューム (volume)**

SA z/OS エンタープライズでシステムにサービスを提供する直接アクセス・ストレージ (DASD) ボリュームまたはテープ・ボリューム。

### **VSE**

[拡張仮想記憶 \(Virtual Storage Extended\)](#) を参照。

### **VTAM**

[仮想記憶通信アクセス方式 \(Virtual Telecommunications Access Method\)](#) を参照。

### **W**

#### **警告しきい値 (warning threshold)**

SA z/OS がグラフィカル・インターフェースの関連アイコンを警告カラーに変更するレベルを決める、アプリケーションまたはボリューム・サービスの値。[アラートしきい値 \(alert threshold\)](#) を参照。

#### **ワークステーション (workstation)**

SA z/OS では、ワークステーションは、オペレーターが日常の業務に使用するグラフィック・ワークステーションを意味する。

#### **オペレーター宛メッセージ (WTO) (write-to-operator (WTO))**

z/OS オペレーター・コンソールのオペレーターにメッセージを送信する要求。この要求は、アプリケーションによって行われ、z/OS 監視プログラムの一部である WTO プロセッサによって処理される。

#### **要応答オペレーター宛メッセージ (WTOR) (write-to-operator-with-reply (WTOR))**

z/OS オペレーター・コンソールのオペレーターにメッセージを送信する要求で、オペレーターからの応答を必要とするもの。この要求は、アプリケーションによって行われ、z/OS 監視プログラムの一部である WTO プロセッサによって処理される。

### **WTO**

[オペレーター宛メッセージ \(write-to-operator\)](#) を参照。

### **WTOR**

[要応答オペレーター宛メッセージ \(write-to-operator-with-reply\)](#) を参照。

### **WWV**

標準時刻の情報を提供する米国連邦情報技術局 (NIST) ラジオ・ステーション。WWVB として知られている第 2 ステーションは、異なる周波数で標準時刻の情報を提供する。

### **X**

### **XCF**

[システム間カップリング・ファシリティ \(cross-system coupling facility\)](#) を参照。

**XCF 結合データ・セット (XCF couple data set)**

MVS/ESA システム・プロダクト、バージョン 5 リリース 1 より前のシスプレックス結合データ・セットの名前。[シスプレックス結合データ・セット \(sysplex couple data set\)](#) も参照。

**XCF グループ (XCF group)**

複数システム・アプリケーションが XCF に対して定義する 関連メンバーのセット。メンバーは、アプリケーションの特定の機能、または インスタンスである。メンバーは、1つのシステム上に存在し、シスプレックス全体にわたって 同じグループの他のメンバーと通信できる。

**XRF**

[拡張リカバリー機能 \(extended recovery facility\)](#) を参照。

**Z****z/OS**

64 ビット実ストレージを使用する IBM メインフレーム・オペレーティング・システム。[基本制御プログラム \(Base Control Program\)](#) も参照。

**z/OS コンポーネント (z/OS component)**

特定の z/OS 機能を実行する、z/OS の一部。SA z/OS では、コンポーネントとは SA z/OS 自動化機能によって管理されるエンティティを指す。

**z/OS subsystem**

z/OS オペレーティング・システムを拡張するソフトウェア・プロダクト。z/OS サブシステムの例としては、JES や TSO/E があります。SA z/OS には、いくつかの z/OS サブシステム用の自動化機能が含まれています。

**z/OS システム (z/OS system)**

z/OS イメージとそれに関連するハードウェアをまとめてこう呼ぶ。両者はひとまとめにして単にシステムまたは z/OS システムと呼ばれることが多い。

**z196**

[IBM zEnterprise 196 \(z196\)](#) を参照。

**zAAP**

[IBM System z Application Assist Processor \(zAAP\)](#) を参照。

**zCPC**

zEnterprise メインフレーム内の主ストレージ、中央処理装置、タイマー、およびチャネルの物理的コレクション。[中央処理装置複合体 \(central processor complex\)](#) も参照。



# 索引

日本語, 数字, 英字, 特殊文字の順に配列されています。  
なお, 濁音と半濁音は清音と同等に扱われています。

## [ア行]

アウトバウンド・ゲートウェイ [134](#)  
アクセシビリティ [xvii](#)  
アクティブ・コネクタ [115](#)  
アクティブ・ヘルス・モニター [26](#)  
アプリケーション  
    自動化への追加 [1](#)  
    ヘルス状況 [23](#)  
アプリケーション、VTAM、SA z/OS への定義 [137](#)  
アプリケーション、z/OS UNIX [83](#)  
アプリケーション・タイプ IMAGE、定義 [127](#)  
アプリケーションのモニター [23](#)  
アプリケーション・モニター状況 [23](#)  
アラート  
    アラート・レベルでの使用可能化 [58](#)  
    概要 [55](#)  
    グローバルに使用可能化 [57](#)  
    システム・レベルでの使用可能化 [57](#)  
    通信のフロー [55](#)  
    通知 [55](#)  
    通知リストでの使用可能化 [57](#)  
    有効にする [56](#)  
    INGCNTL での使用可能化 [57, 58](#)  
一時データ・セット HLR  
    定義 [130](#)  
イベント、リソースのライフ・サイクル [61](#)  
インストール  
    自動化プロシージャ [17](#)  
インバウンド・ゲートウェイ [134](#)  
エラー・コード [10](#)  
エレメント名  
    自動リスタート・マネージャ内 [245](#)  
エレメント名、自動リスタート・マネージャ  
    定義 [245](#)  
エンキュー  
    長時間、リカバリーのカスタマイズ [120](#)  
    長時間、自動化 [129](#)  
    長時間、処理 [118](#)  
大文字小文字の区別、Linux コンソール・メッセージ [72](#)  
オペレーター・カスケード [250](#)

## [カ行]

解決  
    WTO(R) バッファ不足 [116](#)  
開始済みタスク・ジョブ名  
    定義 [130](#)  
外部共通グローバル変数 [203](#)  
外部コード、自動化プロシージャ [10](#)  
概要  
    アラート [55](#)  
    OMEGAMON を使用したモニター [31](#)  
拡張自動化オプション

拡張自動化オプション (続き)  
    外部グローバル変数 [203, 204](#)  
    出口 [154](#)  
カスケード [250](#)  
カスタマイズ  
    Linux ターゲット・システム [80](#)  
    システム・ロガー・リカバリー [115](#)  
    ターゲット・システム [80](#)  
    代替 CDS リカバリー [114](#)  
    ハング・コマンド・リカバリー [120](#)  
    プロキシー・リソースの自動化 [70](#)  
    並列シブプレックスの機能強化を使用するシステム [131](#)  
    IXC102A メッセージの自動化 [121](#)  
    IXC402D メッセージの自動化 [121](#)  
    MVS ゲスト・ターゲット・システム [80](#)  
    SDF [229](#)  
    VM ターゲット・システム [81](#)  
    VSE ターゲット・システム [81](#)  
カスタマイズ・ダイアログ 出口  
    呼び出し [152](#)  
カップリング・ファシリティ [115](#)  
カップリング・ファシリティ、管理 [115](#)  
可用性、報告  
    概要 [61](#)  
    リソースのライフ・サイクル [61](#)  
    DB2 への書き込み [66](#)  
    INGPUSMF ユーティリティ [64](#)  
    INGPUSMF ユーティリティ JCL [65](#)  
    INGPUSMF ユーティリティ JCL、ユーザー・オプション [65](#)  
    INGPUSMF ユーティリティの出力 [64](#)  
    INGPUSMF ユーティリティの戻りコード [66](#)  
    SMF レコード・レイアウト [62](#)  
環境設定出口 [155](#)  
監視状況  
    モニター [23](#)  
管理  
    カップリング・ファシリティ [115](#)  
    結合データ・セット [113](#)  
    システム・ロガー [114](#)  
キーボード [xvii](#)  
疑似出口 [165](#)  
基本 CDS [113](#)  
共通グローバル変数 [10, 203](#)  
共通自動化項目、定義 [130](#)  
グローバル変数名、自動化プロシージャ [21](#)  
クローン ID  
    自動リスタート・マネージャ [245](#)  
クローン ID、自動リスタート・マネージャ [204](#)  
ゲートウェイ  
    アウトバウンド [134](#)  
    インバウンド [134](#)  
ゲートウェイ・セッション  
    定義 [134](#)  
ゲスト・ターゲット・システム  
    Linux [79](#)  
    Linux、ユーザー・ログオン [79](#)

ゲスト・ターゲット・システム (続き)

MVS [79](#)

MVS、NIP コンソール [79](#)

MVS、NIP メッセージ [79](#)

MVS、問題判別モード [79](#)

ProcOps サービス・マシン [79](#)

VSE [79](#)

ゲスト・マシン、プロセッサ操作サポート [79](#)

結合データ・セット

管理 [113](#)

基本 CDS [113](#)

代替 CDS [113](#)

代替 CDS、リカバリー [113](#)

代替、指定 [125](#)

の連続可用性の保証 [113](#)

の連続可用性を使用可能にする [125](#)

ポリシー [113](#)

CFRM [125](#)

SYSPLEX [125](#)

構造

自動化プロシージャの [7](#)

構築

新規自動化定義 [77](#)

構文、バッチ・ジョブのコマンド・ステートメント [98](#)

コネクタ

アクティブ [115](#)

失敗した持続 [115](#)

コマンド

自動化プロシージャでの使用 [7](#)

プロセッサ操作 [14](#)

NetView [98](#)

コマンド、SDFCONF [240](#)

コマンド、長時間エンキューの定義 [130](#)

コマンド、モニター・リソース [25](#)

コマンド受信側 [95](#)

コマンド出力をリダイレクトするバッチ・ジョブ [99](#)

コマンド・フラディング・リカバリー [119](#)

## [サ行]

再構築

システム管理 [115](#)

ユーザー管理 [115](#)

再ロード・アクション出口 [165](#)

再ロード許可出口 [165](#)

削除、SDF からの処理済み WTO(R) の [169](#)

サブシステム

自動化への追加 [1](#)

初期化コマンドにおける UP [165](#)

システム (system)

定義 [125](#)

プロセッサへの接続 [125](#)

システム管理再構築 [115](#)

システム操作制御ファイル [77](#)

システムの除去

有効にする [127](#)

システム・ロガー

リカバリー、ディレクトリー不足 [114](#)

管理 [114](#)

ディレクトリー範囲 [114](#)

リカバリー、カスタマイズ [115](#)

ログ・ストリーム [114](#)

ログ・ストリーム・データ・セット [114](#)

LOGR 結合データ・セット [114](#)

システム・ログ障害のリカバリー [168](#)

シスプレックス機能

オン/オフの切り替え [131](#)

シスプレックス障害管理 [120](#)

シスプレックスの自動化

有効にする [113](#)

持続ストラクチャー [115](#)

持続接続 [115](#)

失敗した持続コネクタ [115](#)

自動化

Linux コンソール・メッセージ、大文字小文字の区別 [72](#)

Linux コンソール・メッセージ、セキュリティー考慮事項 [72](#)

エンキュー、長時間 [129](#)

拡張 [7](#)

拡張機能 [204](#)

シスプレックスの、使用可能にする [113](#)

長時間エンキュー [129](#)

へのアプリケーションの追加 [1](#)

補助記憶域不足のリカバリー [130](#)

メッセージ IXC102A [128](#)

メッセージ IXC402D [128](#)

IXC102A メッセージ [120](#)

IXC402D メッセージ [120](#)

Linux コンソール・メッセージ [72](#)

Linux コンソール・メッセージ、制約事項および制限事項 [72](#)

USS リソース [83](#)

自動化、プロセッサ操作制御リソースの [69](#)

自動化オペレーター ID

追加の [131](#)

自動化状況ファイル

コマンドの使用 [10](#)

ユーザー独自の情報のコーディング [20](#)

自動化処理

実行 [9](#)

自動化制御ファイル

再ロード・アクション出口 [165](#)

再ロード許可出口 [165](#)

SDF の定義 [244](#)

自動化セットアップ、定義 [84](#)

自動化セットアップの定義 [84](#)

自動化ネットワーク [133](#)

自動化の拡張 [7](#)

自動化のカスタマイズ

システム操作の [9](#)

プロセッサ操作の [11](#)

自動化フラグ出口

サンプル [160](#)

自動化プロシージャ

インストール [17](#)

外部コード [10](#)

グローバル変数名 [21](#)

構造 [7](#)

コマンドの使用 [7](#)

作成 [7](#)

初期化 [8](#)

説明 [7](#)

テスト [18](#)

デバッグ [18](#)

独自の作成 [7](#)

汎用化 [13](#)

プログラミングの推奨事項 [20](#)

メッセージの開発 [14](#)



自動化プロシージャー (続き)  
   呼び出し [7](#)  
   ルーチンの使用 [7](#)  
   例 [15](#)  
   AOCTRACE の使用 [18](#)  
   REXX コーディング例 [19](#)  
 自動化プロシージャーの作成 [7](#)  
 自動化プロシージャーの初期化 [8](#)  
 自動化プロシージャーのメッセージの開発 [14](#)  
 自動化プロシージャーの例 [15](#)  
 自動化リソース、z/OS UNIX 自動化 [85](#)  
 自動化ルーチン  
   概要 [167](#)  
   処理 [168](#)  
   ドレーン処理、JES2 シャットダウン前の [169](#)  
   AMRF バッファ不足の処理 [169](#)  
   AOFRSA01 [177](#)  
   AOFRSA02 [178](#)  
   AOFRSA03 [179](#)  
   AOFRSA08 [181](#)  
   AOFRSA0C [183](#)  
   AOFRSA0E [186](#)  
   AOFRSA0G [187](#)  
   AOFRSD07 [188](#)  
   AOFRSD09 [189](#)  
   AOFRSD0F [190](#)  
   AOFRSD0G [192](#)  
   AOFRSD0H [193](#)  
   HASP099 [195](#)  
   IMS トランザクション・リカバリー [169](#)  
   INGRCJSP [197](#)  
   INGRMJSP [195](#)  
   INGRTAPE [198](#)  
   INGRX740 [199](#)  
   LOGREC データ・セット処理 [167](#)  
   SMF データ・セット処理 [168](#)  
   SVC ダンプ処理 [168](#)  
 始動定義、z/OS UNIX リソースの [87](#)  
 始動についての考慮事項、プロキシー・リソースの自動化 [71](#)  
 自動リスタート・マネージャー  
   エレメント名の定義 [245](#)  
   MOVE グループ [246](#)  
 自動リスタート・マネージャーの MOVE グループ [246](#)  
 自動リスタート・マネージャーのエレメント名 [204](#)  
 シナリオ  
   OMEGAMON [36](#)  
   シャットダウンについての考慮事項、プロキシー・リソースの自動化 [71](#)  
   受動の、イベント・ベース・ヘルス・モニター [28](#)  
   使用  
     PROCESSOR INFO ポリシー項目 [124](#)  
   障害のあるシステム、の分離 [120](#)  
   使用可能化、コマンド受信側 [95](#)  
   使用可能化、リレーショナル・データ・サービスの [109](#)  
   状況記述子  
     状況コンポーネントへのチェーニング [232](#)  
     伝搬 [234](#)  
   状況情報 [9](#)  
   状況変更コマンド [156](#)  
   情報のコーディング、自動化状況ファイルでの [20](#)  
   ショートカット・キー [xvii](#)  
   初期化プロセス、AOFSERXINT [204](#)  
   ジョブ/ASID 定義、実施

ジョブ/ASID 定義、実施 (続き)  
   長時間エンキューの [129](#)  
 ジョブ/ASID 定義の実施  
   長時間エンキューの [129](#)  
 ジョブの処理、補助記憶域不足のリカバリーの定義 [130](#)  
 ジョブ・ログ・モニター  
   セットアップ [47](#)  
   SPIN パラメーター [52](#)  
 処理、WTOR [145](#)  
 シリアライズ、コマンド処理の [12](#)  
 新規自動化定義  
   構築 [77](#)  
   身体障害 [xvii](#)  
   推奨事項  
     プログラミング、自動化プロシージャー [20](#)  
   ストラクチャー  
     再構築 [115](#)  
     システム管理再構築 [115](#)  
     持続 [115](#)  
     設定リスト [115](#)  
     二重化 [115](#)  
     ユーザー管理再構築 [115](#)  
     割り振り [115](#)  
     割り振り解除 [115](#)  
   スナップショット間隔、長時間エンキューの定義 [130](#)  
   スプールのモニター、JES2 [44](#)  
   制約事項および制限事項、Linux コンソール・メッセージ [72](#)  
   セキュリティ考慮事項、Linux コンソール・メッセージ [72](#)  
   セッション管理  
     OMEGAMON、INGMTRAP [35](#)  
     OMEGAMON、INGOMX [33](#)  
   接続  
     システムをプロセッサへ [125](#)  
   設定リスト [115](#)  
   前提事項、OMEGAMON を使用したヘルス・モニターの [32](#)

**[タ行]**  
 ターゲット・システム、カスタマイズ [80](#)  
 第 2 レベル・システム、VM サポート [79](#)  
 代替 CDS  
   基本 CDS への変更 [113](#)  
 代替 CDS リカバリー  
   カスタマイズ [114](#)  
 代替結合データ・セット  
   指定 [125](#)  
   タスク・グローバル変数 [10](#)  
   端末アクセス機能 [134](#)  
   長時間エンキュー  
     コマンドの定義 [130](#)  
     自動化 [129](#)  
     ジョブ/ASID 定義の実施 [129](#)  
     処理 [118](#)  
     スナップショット間隔の定義 [130](#)  
     リソースの定義 [129](#)  
     IEADMCxx シンボルの定義 [129](#)  
   追加  
     自動化へのアプリケーションの [1](#)  
     自動化へのプロセッサ操作メッセージ [73](#)  
   追加の SA z/OS 自動化プロシージャー、プログラミング [7](#)  
   追加の自動化オペレーター ID [131](#)  
   通信のフロー  
     アラート [55](#)  
   通知

通知 (続き)

[アラート 55](#)

定義

- [長時間エンキューのスナップショット間隔 130](#)
- [補助記憶域不足のリカバリー用ジョブの処理 130](#)
- [補助記憶域不足のリカバリー用のローカル・ページ・データ・セット 130](#)
- [アプリケーション・タイプ IMAGE 127](#)
- [一時データ・セット HLG 130](#)
- [開始済みタスク・ジョブ名 130](#)
- [共通自動化項目 130](#)
- [ゲートウェイ・セッション 134](#)
- [システム \(system\) 125](#)
- [自動化制御ファイル中の SDF 244](#)
- [長時間エンキューの IEADMCxx シンボル 129](#)
- [長時間エンキューのコマンド 130](#)
- [長時間エンキューの定義 129](#)
- [物理シスプレックス 125](#)
- [プロセッサ 124, 127](#)
- [論理区画 125](#)
- [論理シスプレックス 125](#)
- [IMAGE アプリケーション・タイプ 127](#)
- [SA z/OS への VTAM アプリケーションの 137](#)
- [SDF フォーカル・ポイント・システム 133](#)
- [SYSPLEX ポリシー項目 125](#)
- [TAF フルスクリーン・セッション 134](#)

停止定義、z/OS UNIX リソースの 87

[ディレクトリー範囲 114](#)

出口

- [カスタマイズ・ダイアログ 出口 149](#)
- [環境設定出口 155](#)
- [疑似出口 165](#)
- [サンプル自動化フラグ出口 160](#)
- [状況変更コマンド 156](#)
- [初期化コマンドにおけるサブシステム UP 165](#)
- [静的出口 156](#)
- [テスト 165](#)
- [フラグ出口 157](#)
- [AOFEXC00 160](#)
- [AOFEXC01 160](#)
- [AOFEXC02 161](#)
- [AOFEXC03 161](#)
- [AOFEXC04 161](#)
- [AOFEXC05 161](#)
- [AOFEXC06 161](#)
- [AOFEXC07 161](#)
- [AOFEXC08 161](#)
- [AOFEXC09 162](#)
- [AOFEXC11 162](#)
- [AOFEXC13 162](#)
- [AOFEXC14 162](#)
- [AOFEXC15 162](#)
- [AOFEXC16 162](#)
- [AOFEXC17 162](#)
- [AOFEXC18 163](#)
- [AOFEXC19 163](#)
- [AOFEXC20 163](#)
- [AOFEXC21 163](#)
- [AOFEXC22 163](#)
- [AOFEXC23 164](#)
- [AOFEXC24 164](#)
- [AOFEXC25 164](#)
- [AOFEXC26 164](#)
- [AOFEXC27 164](#)

出口 (続き)

- [AOFEXC28 164](#)
- [AOFEXC29 165](#)
- [AOFEXDEF 154](#)
- [AOFEXI01 154](#)
- [AOFEXI02 154](#)
- [AOFEXI03 154](#)
- [AOFEXI04 154](#)
- [AOFEXI05 154](#)
- [AOFEXI06 155](#)
- [AOFEXINT 155, 165](#)
- [AOFEXSTA 156](#)
- [AOFEXX02 157](#)
- [AOFEXX04 157](#)
- [AOFEXX05 157](#)
- [BUILD 処理 149](#)
- [CONVERT 処理 151](#)
- [COPY 処理 150](#)
- [DELETE 処理 151](#)
- [IMPORT 機能 151](#)
- [INGEEXIT 157](#)
- [INGEX01 149](#)
- [INGEX02 149](#)
- [INGEX03 150](#)
- [INGEX04 150](#)
- [INGEX05 151](#)
- [INGEX06 151](#)
- [INGEX07 151](#)
- [INGEX08 151](#)
- [INGEX09 151](#)
- [INGEX12 151](#)
- [INGEX14 151](#)
- [INGEX16 151](#)
- [INGEX17 151](#)
- [INGEX18 151](#)
- [RENAME 機能 151](#)

テスト

- [自動化プロシージャ 18](#)
- [詳細 20](#)
- [メッセージ 76](#)
- [NetView 機能 19](#)

テスト、出口の 165

デバッグ

- [自動化プロシージャ 18](#)
- [NetView 機能 19](#)
- [z/OS UNIX 自動化 94](#)

同期ハードウェア・コマンド、パイプおよび ISQCCMD の使用 77

トランザクション・リカバリー

[IMS 169](#)

ドレーン処理、JES2 シャットダウン前の [169](#)

[ナ行]

ネットワーク自動化  
定義処理 [133](#)

[ハ行]

ハードウェア・コマンド

同期、パイプおよび ISQCCMD の使用 [77](#)

非同期、パイプおよび ISQCCMD の使用 [78](#)

パイプ

パイプ (続き)  
同期ハードウェア・コマンドへの使用 [77](#)  
非同期ハードウェア・コマンドへの使用 [78](#)  
バックアップ・フォーカル・ポイント [236](#)  
バッチ・コマンド・インターフェース  
JCL [99](#)  
バッチ・ジョブ  
関数 INGRCRPC [102](#)  
コマンド受信側の外部 TSO API [102](#)  
コマンド出力のリダイレクト [99](#)  
コマンド情報の取得 [99](#)  
コマンド・ステートメントの構文 [98](#)  
コマンドの継続 [98](#)  
サンプル JCL [98](#)  
セキュリティ考慮事項 [107](#)  
有効なコマンド・タイプ [98](#)  
例 [106](#)  
NetView コマンドの実行依頼 [98](#)  
TSO から SA NetView Agent へのリモート・プロシージャ  
ー呼び出し [102](#)

パネル  
SYSLOG [181](#)  
「ハング」コマンド・リカバリー [118](#)  
ハング・コマンド・リカバリー、カスタマイズ [120](#)  
汎用  
自動化 [204](#)  
汎用自動化プロシージャ [13](#)  
汎用自動化プロシージャの作成 [13](#)  
非同期ハードウェア・コマンド、パイプおよび ISQCCMD の  
使用 [78](#)  
ファイル・マネージャ・コマンド [10](#)  
ファイル・モニター、z/OS UNIX 自動化 [87](#)  
フォーカル・ポイント・システム 定義 [133](#)  
物理シスプレックス、定義 [125](#)  
フラグ出口 [157](#)  
プロキシー・リソース  
始動についての考慮事項 [71](#)  
シャットダウンについての考慮事項 [71](#)  
のための自動化のカスタマイズ [70](#)  
プログラミング  
自動化プロシージャの推奨事項 [20](#)  
追加の SA z/OS 自動化プロシージャ [7](#)  
プログラミングの推奨事項  
自動化プロシージャ [20](#)  
プロセス・モニター、z/OS UNIX 自動化 [86](#)  
プロセッサ  
定義 [124](#), [127](#)  
プロセッサ操作  
ゲスト・マシン・サポート [79](#)  
プロセッサ操作コマンド [14](#)  
プロセッサ操作コマンド・メッセージ [74](#)  
プロセッサ操作制御リソース、自動化 [69](#)  
プロセッサ操作リソース [69](#)  
プロセッサ操作リソース・メッセージの自動化 [69](#)  
並行更新 [236](#)  
ヘルス状況の戻りコード [26](#)  
ヘルス・モニター  
アクティブ [26](#)  
イベント・ベース [28](#)  
概要 [24](#)  
受動の [28](#)  
ヘルス・モニター、OMEGAMON XE シチュエーション  
概要 [37](#)  
ヘルス・モニター、OMEGAMON 例外

ヘルス・モニター、OMEGAMON 例外 (続き)  
概要 [36](#)  
ヘルスを基にした自動化、OMEGAMON を使用する  
推奨事項 [37](#)  
プログラミング手法 [29](#)  
リカバリー手法 [26](#)  
報告、可用性およびリカバリー時間の [61](#)  
補助記憶域不足のリカバリー  
自動化 [130](#)  
ジョブの処理の定義 [130](#)  
ローカル・ページ・データ・セットの定義 [130](#)  
補助モード  
概要 [18](#)  
自動化プロシージャをテストする場合 [18](#)  
ポリシー  
結合データ・セット [113](#)  
CFRM [115](#)

## [マ行]

マイナー・リソース  
リソース名 [159](#)  
メッセージ  
自動化プロシージャ用の開発 [14](#)  
UNIX syslogd のトラップ [92](#)  
メッセージの自動化  
プロセッサ操作リソースの準備 [71](#)  
Linux コンソール・メッセージ [72](#)  
Linux コンソール・メッセージ、大文字小文字の区別 [72](#)  
Linux コンソール・メッセージ、制約事項および制限事  
項 [72](#)  
Linux コンソール・メッセージ、セキュリティ考慮事項  
[72](#)  
メッセージの自動化、プロセッサ操作リソースの [69](#)  
メッセージのテスト [76](#)  
メッセージ表示 [248](#)  
メンバー、INGEI004 [78](#)  
戻りコード、ヘルス状況 [26](#)  
モニター  
アプリケーション [23](#)  
監視状況 [23](#)  
ヘルス、アクティブ [26](#)  
ヘルス、イベント・ベースの [28](#)  
ヘルス、概要 [24](#)  
ヘルス、受動の [28](#)  
CICS ヘルス [40](#)  
CICS リンク [40](#)  
CICSplex [40](#)  
IMS 自動化 [45](#)  
JES3 コンポーネント [41](#)  
OMEGAMON XE シチュエーションを使用 [37](#)  
OMEGAMON を使用したヘルスの [31](#)  
モニター・コマンド、INGMTRAP [35](#)  
モニター・リソース  
コマンド [25](#)  
CICS モニター用の定義 [40](#)  
OMEGAMON XE シチュエーション用の定義 [38](#)  
モニター・ルーチン  
独自の作成 [23](#)  
AOF RJ3MN [42](#)  
AOF RJ3RC [44](#)  
モニター・ルーチン、z/OS UNIX リソースの [84](#)  
問題判別モード  
MVS ゲスト・ターゲット・システム [79](#)

## [ヤ行]

有効にする  
アラート [56](#)  
アラート、INGCNTL による [57](#)  
結合データ・セットの連続可用性 [125](#)  
システムの除去 [127](#)  
シスプレックスの自動化 [113](#)  
使用可能化、通知リストでの [57](#)  
INGCNTL でのアラート送信 [58](#)  
WTOR(R) バッファ不足のリカバリー [126](#)  
ユーザー管理再構築 [115](#)  
ユーザー出口  
静的出口 [156](#)  
ユーザー・ログオン、LINUX ゲスト・ターゲット・システム [79](#)  
呼び出し  
自動化プロシージャ [7](#)

## [ラ行]

リカバリー  
コマンド・フラッディング [119](#)  
システム・ロガー、カスタマイズ [115](#)  
システム・ロガー、ディレクトリー不足 [114](#)  
システム・ログ障害 [168](#)  
代替 CDS [113](#)  
代替 CDS、カスタマイズ [114](#)  
長時間エンキュー、カスタマイズ [120](#)  
長時間エンキューの処理 [118](#)  
「ハング」コマンド [118](#)  
補助記憶域不足 [121](#)  
補助記憶域不足、自動化 [130](#)  
WTO(R) バッファ不足 [116](#)  
WTOR(R) バッファ不足、使用可能にする [126](#)  
リカバリー時間、報告  
概要 [61](#)  
リソースのライフ・サイクル [61](#)  
DB2 への書き込み [66](#)  
INGPUSMF ユーティリティー [64](#)  
INGPUSMF ユーティリティー [JCL 65](#)  
INGPUSMF ユーティリティー JCL、ユーザー・オプション [65](#)  
INGPUSMF ユーティリティーの出力 [64](#)  
INGPUSMF ユーティリティーの戻りコード [66](#)  
SMF レコード・レイアウト [62](#)  
リソース、長時間エンキューの定義 [129](#)  
リソースのライフ・サイクル、イベント [61](#)  
リダイレクト、バッチ・ジョブ・コマンドの出力 [99](#)  
リレーショナル・データ・サービス [109](#)  
ルーチン  
自動化プロシージャでの使用 [7](#)  
レイアウト、SMF レコード [62](#)  
連続可用性、結合データ・セット  
保証 [113](#)  
有効にする [125](#)  
ローカル・ページ・データ・セット、定義  
補助記憶域不足のリカバリー [130](#)  
ログ・ストリーム [114](#)  
ログ・ストリーム・データ・セット [114](#)  
論理区画  
定義 [125](#)  
論理シスプレックス、定義 [125](#)

## [数字]

1 次フォール・ポイント [236](#)

## A

AMRF バッファ不足の処理 [169](#)  
AOCMSG 汎用ルーチン [10](#)  
AOCMSG 呼び出し [14](#)  
AOCQRY 共通ルーチン  
自動化の可用性 [9](#)  
メッセージの自動化 [20](#)  
AOCTRACE  
テストでの使用 [18](#)  
トレースでの使用 [18](#)  
AOCUPDT 共通ルーチン  
状況情報を更新するには [9](#)  
AOCUPDT コマンド  
と AOFEXSTA 出口 [156](#)  
AOF\_AAO\_AAREQUEST\_MAX\_WAIT [204](#)  
AOF\_AAO\_AOCUPDT\_LOGGING [204](#)  
AOF\_AAO\_AOCUPDT\_PRESERVE\_CASE [204](#)  
AOF\_AAO\_GWPPING\_RETRY [204](#)  
AOF\_AAO\_HW\_VALIDATION [204](#)  
AOF\_AAO\_INJECT\_NOFORCE\_REQ [204](#)  
AOF\_AAO\_IPL\_COMPLETE\_MSG [204](#)  
AOF\_AAO\_ISQ\_DYNTGT [204](#)  
AOF\_AAO\_ISQ\_KALIST [204](#)  
AOF\_AAO\_JLM\_RECYCLE [204](#)  
AOF\_AAO\_MSG\_EHKVAR [204](#)  
AOF\_AAO\_MVSTAPEMON [204](#)  
AOF\_AAO\_OMVS\_SHUTDOWN [204](#)  
AOF\_AAO\_RDS\_TSO\_DSN [204](#)  
AOF\_AAO\_RDS\_TSO\_RACFHLQ [204](#)  
AOF\_AAO\_RETENTIONPERIOD [204](#)  
AOF\_AAO\_SDF\_MSGDEL\_OPERID [204](#)  
AOF\_AAO\_SDFCSAPLEX [204](#)  
AOF\_AAO\_SDFCxxx.\* [204](#)  
AOF\_AAO\_SDFPFP\_ROOT.\* [204](#)  
AOF\_AAO\_SDFROOT\_LISTn [204](#)  
AOF\_AAO\_SDFROOT.\* [204](#)  
AOF\_AAO\_SEC\_PPIAUTH [204](#)  
AOF\_AAO\_SHUTDOWN\_BOOST [204](#)  
AOF\_AAO\_SHUTDOWN\_STOPAPPL [204](#)  
AOF\_AAO\_SHUTSYS\_OLD [204](#)  
AOF\_AAO\_TRANRERUN [204](#)  
AOF\_AAO\_TWS\_CHK\_CONDDEP [204](#)  
AOF\_AAO\_TWS\_CHK\_OUTPUT\_NETLOG [204](#)  
AOF\_AAO\_TWS\_CMD\_OUTPUT\_NETLOG [204](#)  
AOF\_AAO\_TWS\_ERRMSG [204](#)  
AOF\_AAO\_TWS\_MAX\_WAIT\_TIME [204](#)  
AOF\_AAO\_TWS\_RESYSPLX [204](#)  
AOF\_ASSIGN\_JOBNAME [204](#)  
AOF\_E2E\_EAS\_PPI [204](#)  
AOF\_E2E\_EVT\_RETRY [204](#)  
AOF\_E2E\_EXREQ\_NETLOG [204](#)  
AOF\_E2E\_TKOVN\_TIMEOUT [204](#)  
AOF\_EMCS\_AUTOTASK\_ASSIGNMENT [204](#)  
AOF\_EMCS\_CN\_ASSIGNMENT [204](#)  
AOF\_FDBK\_CS\_HONOR [204](#)  
AOF\_INIT\_MCSFLAG [204](#)  
AOF\_INIT\_ROUTCDE [204](#)  
AOF\_INIT\_SYSCONID [204](#)  
AOF\_PRODLVL [203](#)

AOF.ODEBUG [203](#)  
 AOF.OTRACE [203](#)  
 AOFACFINIT [204](#)  
 AOFAOCCLONE [203](#)  
 AOFARMQUERYRETRY [204](#)  
 AOFARMQUERYWAIT [204](#)  
 AOFBFP [203](#)  
 AOFCFP [203](#)  
 AOFCNMASK [204](#)  
 AOFCNMASK\_EXTENDED [204](#)  
 AOFCOMPL [203](#)  
 AOFCONFIRM グローバル変数 [165](#)  
 AOFDEBUG [203](#)  
 AOFDEBUG グローバル変数 [18](#)  
 AOFDEFAULT\_TARGET [204](#)  
 AOFDOM [247, 248](#)  
 AOFEXC00 出口 [160](#)  
 AOFEXC01 出口 [160](#)  
 AOFEXC02 出口 [161](#)  
 AOFEXC03 出口 [161](#)  
 AOFEXC04 出口 [161](#)  
 AOFEXC05 出口 [161](#)  
 AOFEXC06 出口 [161](#)  
 AOFEXC07 出口 [161](#)  
 AOFEXC08 出口 [161](#)  
 AOFEXC09 出口 [162](#)  
 AOFEXC11 出口 [162](#)  
 AOFEXC13 出口 [162](#)  
 AOFEXC14 出口 [162](#)  
 AOFEXC15 出口 [162](#)  
 AOFEXC16 出口 [162](#)  
 AOFEXC17 出口 [162](#)  
 AOFEXC18 出口 [163](#)  
 AOFEXC19 出口 [163](#)  
 AOFEXC20 出口 [163](#)  
 AOFEXC21 出口 [163](#)  
 AOFEXC22 出口 [163](#)  
 AOFEXC23 出口 [164](#)  
 AOFEXC24 [164](#)  
 AOFEXC25 [164](#)  
 AOFEXC26 [164](#)  
 AOFEXC27 [164](#)  
 AOFEXC28 [164](#)  
 AOFEXC29 [165](#)  
 AOFEXDEF 出口 [154](#)  
 AOFEXI01 出口 [154](#)  
 AOFEXI02 出口 [154](#)  
 AOFEXI03 出口 [154](#)  
 AOFEXI04 出口 [154](#)  
 AOFEXI05 出口 [154](#)  
 AOFEXI06 出口 [155](#)  
 AOFEXINT 出口 [155, 165, 204](#)  
 AOFEXPLAIN\_USER [204](#)  
 AOFEXSTA 出口 [156](#)  
 AOFEXX02 出口 [157](#)  
 AOFEXX04 出口 [157](#)  
 AOFEXX05 [157](#)  
 AOFINITIALSTARTTYP [203](#)  
 AOFINITREPLY [204](#)  
 AOFJESPREFX [203](#)  
 AOFJRYCMD の説明 [99](#)  
 AOFLOCALHOLD [204](#)  
 AOFMATLISTING [204](#)  
 AOFMSGSY [247](#)  
 AOFOPCCMDMSG [204](#)  
 AOFRESTARTALWAYS [204](#)  
 AOFRJ3MN モニター・ルーチン [42](#)  
 AOFRJ3RC モニター・ルーチン [44](#)  
 AOFRMTCMDWAIT [204](#)  
 AOFRPCWAIT [204](#)  
 AOFRSA01 自動化ルーチン [177](#)  
 AOFRSA02 自動化ルーチン [178](#)  
 AOFRSA03 自動化ルーチン [179](#)  
 AOFRSA08 自動化ルーチン [181](#)  
 AOFRSA0C 自動化ルーチン [183](#)  
 AOFRSA0E 自動化ルーチン [186](#)  
 AOFRSA0G 自動化ルーチン [187](#)  
 AOFRSD01 自動化ルーチン [197](#)  
 AOFRSD07 自動化ルーチン [188](#)  
 AOFRSD09 自動化ルーチン [189](#)  
 AOFRSD0F 自動化ルーチン [190](#)  
 AOFRSD0G 自動化ルーチン [192](#)  
 AOFRSD0H 自動化ルーチン [193](#)  
 AOFSENDALERT [204](#)  
 AOFSERXINT [204](#)  
 AOFSETSTATESCOPE [220](#)  
 AOFSHUTDELAY [204](#)  
 AOFSMARTMAT [204](#)  
 AOFSPoolFULLCMD [204](#)  
 AOFSPoolSHORTCMD [204](#)  
 AOFSTATUSCMDSEL [204](#)  
 AOFSUBSYS [203, 204](#)  
 AOFSYS [247, 248](#)  
 AOFSYSNAME [203](#)  
 AOFSYSPLEX [203](#)  
 AOFSYSPLEXGROUP [203](#)  
 AOFSYSTEM [203](#)  
 AOFUSSWAIT [204](#)  
 ARM (Automatic restart management) [246](#)  
 ASCB チェーニング  
     グローバル変数 [212](#)  
 ASFUSER コマンド [20](#)

**B**

BASEOPER [204](#)

**C**

CDEMATCH 共通ルーチン [20](#)  
 CDS [113](#)  
 CF [115](#)  
 CFRM 結合データ・セット [115, 125](#)  
 CFRM ポリシー [115](#)  
 CHKTHRES 自動化プロシージャ [10](#)  
 CICS ヘルス・モニター [37, 40](#)  
 CICS モニター  
     コンポーネントの概要 [40](#)  
     モニター・リソースの定義 [40](#)  
     VOST の管理 [40](#)  
 CICS リンク・モニター [40](#)  
 CICSplex モニター [40](#)  
 CNMCMdu メンバー [17](#)

**D**

DB2、SMF レポートの書き込み [66](#)

[DISPEVT\\_WAIT 220](#)  
[DISPEVTS\\_WAIT 220](#)  
[DISPGW\\_COLUMNS 220](#)  
[DISPMTR\\_COLUMNS 220](#)  
[DISPSTAT\\_COLUMNS 220](#)  
[DISPTRG\\_WAIT 220](#)  
[DISIPARM データ・セット 17](#)

## E

[ENQ 118](#)  
[EVIRYDPO\\_COLUMNS 220](#)  
[EXPLAIN 215](#)  
[EXTSTART 状況 204, 245](#)

## G

[GDPS 環境、z/OS システムのシャットダウン 141](#)  
[GDPS 環境での z/OS システムのシャットダウン 141](#)

## H

[HASP099 自動化ルーチン 195](#)

## I

[IDENT 20](#)  
[IEADMCxx シンボル、定義  
長時間エンキューの 129](#)  
[IMAGE アプリケーション・タイプ、定義 127](#)  
[IMS 自動化、モニター 45](#)  
[IMS トランザクション・リカバリー 169](#)  
[INCLUDE ステートメント 243](#)  
[INGAMS\\_COLUMNS 220](#)  
[INGAUTO\\_INTERVAL 220](#)  
[INGCF コマンド 115](#)  
[INGCICS\\_CORRWAIT 204](#)  
[INGDLG 152](#)  
[INGEAXIT 出口 157](#)  
[INGEI004 メンバー 78](#)  
[INGEVENT\\_WAIT 220](#)  
[INGEX01 149](#)  
[INGEX02 149](#)  
[INGEX03 150](#)  
[INGEX04 150](#)  
[INGEX05 151](#)  
[INGEX06 151](#)  
[INGEX07 151](#)  
[INGEX08 151](#)  
[INGEXEC\\_RESP 220](#)  
[INGEXEC\\_SELECT 220](#)  
[INGEXEC\\_TIMEOUT 220](#)  
[INGEXEC\\_WAIT 220](#)  
[INGGROUP\\_WAIT 220](#)  
[INGHIST\\_MAX 220](#)  
[INGHIST\\_WIMAX 220](#)  
[INGIMS\\_CMDWAIT 220](#)  
[INGIMS\\_CORRWAIT 204](#)  
[INGIMS\\_REQ 220](#)  
[INGINFO\\_RSTAT 220](#)  
[INGINFO\\_WAIT 220](#)  
[INGKLUP\\_WAIT 220](#)  
[INGLIST\\_COLUMNS 220](#)

[INGLIST\\_WAIT 220](#)  
[INGLKUP\\_TIMEOUT 220](#)  
[INGMON\\_WAIT 220](#)  
[INGMON、プログラミング手法 29](#)  
[INGMOVE\\_WAIT 220](#)  
[INGMTRAP モニター・コマンド 35](#)  
[INGOMX API 33](#)  
[INGOPC\\_MULTIPLIER 204](#)  
[INGPAC\\_SHOWNOLIMIT 204](#)  
[INGPAC\\_WAIT 220](#)  
[INGPUSMF ユーティリティ  
出力 64  
導入済み 64  
戻りコード 66  
JCL 65  
JCL、ユーザー・オプション 65](#)  
[INGRCJSP 自動化ルーチン 197](#)  
[INGRELS\\_SHOW 220](#)  
[INGRELS\\_WAIT 220](#)  
[INGREQ\\_BOOST 220](#)  
[INGREQ\\_EXPIRE 220](#)  
[INGREQ\\_INTERRUPT 220](#)  
[INGREQ\\_ORIGINATOR 204](#)  
[INGREQ\\_OVERRIDE 220](#)  
[INGREQ\\_PRECHECK 220](#)  
[INGREQ\\_PRI 220](#)  
[INGREQ\\_PRI.E2EMGR 220](#)  
[INGREQ\\_REMOVE 220](#)  
[INGREQ\\_REMOVE.START 220](#)  
[INGREQ\\_REMOVE.STOP 220](#)  
[INGREQ\\_RESTART 220](#)  
[INGREQ\\_SCOPE 220](#)  
[INGREQ\\_SOURCE 220](#)  
[INGREQ\\_TIMEOUT 220](#)  
[INGREQ\\_TYPE 220](#)  
[INGREQ\\_VERIFY 220](#)  
[INGREQ\\_WAIT 220](#)  
[INGRMJSP 自動化ルーチン 195](#)  
[INGRPT\\_WAIT 220](#)  
[INGRTAPE 自動化ルーチン 198](#)  
[INGRUN\\_CMT 220](#)  
[INGRUN\\_MULT 220](#)  
[INGRUN\\_OVERRIDE 220](#)  
[INGRUN\\_PERSISTENT 220](#)  
[INGRUN\\_PRI 220](#)  
[INGRUN\\_REQ 220](#)  
[INGRUN\\_RUNMODE 220](#)  
[INGRUN\\_RUNRES 220](#)  
[INGRUN\\_SYSTEM 220](#)  
[INGRUN\\_TARGET 220](#)  
[INGRUN\\_TYPE 220](#)  
[INGRUN\\_VERIFY 220](#)  
[INGRUN\\_WAIT 220](#)  
[INGRX740 自動化ルーチン 199](#)  
[INGSCHED\\_WAIT 220](#)  
[INGSET\\_VERIFY 220](#)  
[INGSET\\_WAIT 220](#)  
[INGSTX\\_WAIT 220](#)  
[INGTRIG\\_WAIT 220](#)  
[INGUSS コマンド  
例 87](#)  
[INGUSS コマンドの例 87](#)  
[INGVOTE\\_EXCLUDE 220](#)  
[INGVOTE\\_SOURCE 220](#)



INGVOTE\_STATUS [220](#)  
INGVOTE\_VERIFY [220](#)  
ISQCCMD  
同期ハードウェア・コマンドへの使用 [77](#)  
非同期ハードウェア・コマンドへの使用 [78](#)  
ISQEXEC コマンド [12, 73](#)  
ISQOVRD [74](#)  
ISQOVRD コマンド [12](#)  
ISQXLOC コマンド [12](#)  
ISQXMON コマンド [73](#)  
ISQXUNL コマンド [12](#)  
IXC102A メッセージ  
自動化 [120, 128](#)  
自動化のカスタマイズ [121](#)  
IXC402D メッセージ  
自動化 [120, 128](#)  
自動化のカスタマイズ [121](#)  
IXCARM マクロ呼び出し [245](#)  
IXCMIAPU [245](#)

## J

JES2 スプールのモニター [44](#)  
JES2 スプール・ファイル  
ジョブ・ログ・モニター [47](#)  
JES3 モニター [41](#)

## L

LINUX ゲスト・ターゲット・システム、ユーザー・ログオン [79](#)  
Linux コンソール・メッセージ  
大文字小文字の区別 [72](#)  
自動化 [72](#)  
制約事項および制限事項 [72](#)  
セキュリティ考慮事項 [72](#)  
LINUX ターゲット・システム、カスタマイズ [80](#)  
LOGR 結合データ・セット [114](#)  
LOGREC データ・セット処理 [167](#)

## M

message  
テスト [73, 76](#)  
転送 [73](#)  
転送パス、定義 [133](#)  
ISQ900I [73](#)  
ISQ901I [73](#)  
IXC102A、自動化 [120](#)  
IXC402D、自動化 [120](#)  
MOVED 状況  
自動化 [245](#)  
自動リスタート・マネージャー [245](#)  
MPF リスト [77](#)  
MTR [23](#)  
MVS ゲスト・ターゲット・システム  
問題判別モード [79](#)  
NIP コンソール [79](#)  
NIP メッセージ [79](#)  
MVS 自動リスタート・マネージャー  
エレメント名 [204](#)  
グローバル変数 [204](#)  
クローン ID [204](#)

MVS ターゲット・システム、カスタマイズ [80](#)  
MVSESA.RELOAD.ACTION マイナー・リソース [165](#)  
MVSESA.RELOAD.CONFIRM フラグ [165](#)  
MVSESA.RELOAD.CONFIRM マイナー・リソース [165](#)

## N

NetView  
テストおよびデバッグ機能 [19](#)  
Linux コンソール接続 [72](#)  
NetView コマンド  
バッチ・ジョブからの実行依頼 [98](#)  
別の NetView に対する実行 [99](#)  
NetView 自動化テーブル  
項目の組み合わせ [76](#)  
サンプル項目 [74](#)  
実動 [76](#)  
テーブルの再ロード [77](#)  
フラグメント [247](#)  
プロセッサ操作メッセージの追加 [73](#)  
AOFMSGSY [247](#)  
ISQEXEC [12, 73](#)  
ISQOVRD [12](#)  
ISQXLOC [12](#)  
ISQXMON [73](#)  
ISQXUNL [12](#)  
NetView 自動化テーブルの再ロード [77](#)  
NetView への Linux コンソール接続 [72](#)

## O

OMEGAMON  
使用したヘルス・モニター [31](#)  
使用法シナリオ [36](#)  
セッション管理、INGMTRAP [35](#)  
セッション管理、INGOMX [33](#)  
前提事項 [32](#)  
対話 [32](#)  
ヘルス・モニター [37](#)  
ヘルスを基にした自動化、推奨事項 [37](#)  
ヘルスを基にした自動化、プログラミング手法 [29](#)  
ヘルスを基にした自動化、リカバリー手法 [26](#)  
モニター、概要 [31](#)  
例外、ヘルス・モニター [36](#)  
例外分析 [31](#)  
OMEGAMON XE シチュエーション、モニター [37](#)  
OMEGAMON XE シチュエーションのモニター  
概要 [37](#)  
モニター・リソースの定義 [38](#)

## P

PIPE ラベル [99](#)  
PROCESSOR INFO ポリシー項目  
使用 [124](#)  
ProcOps サービス・マシン  
ゲスト・ターゲット・システム [79](#)  
PSM [79](#)

## R

RDS テーブル・エディター [111](#)  
RELOAD.ACTION フラグ [165](#)



RELOAD.CONFIRM フラグ [165](#)  
REXX PARSE [20](#)  
REXX コーディング例 [19](#)  
REXX トレース・タイプ [18](#)

## S

SA IOM [55](#)  
SA z/OS  
    コマンド ISQXIPM および ISQCMMT [11](#)  
SA z/OS、への VTAM アプリケーションの定義 [137](#)  
SDF  
    階層の定義 [239](#)  
    カスタマイズ [229](#)  
    カスタマイズ・ダイアログでの定義 [244](#)  
    機能 [229](#)  
    コンポーネント [237](#)  
    自動化制御ファイルでの定義 [244](#)  
    始動と停止 [238](#)  
    状況記述子 [230](#)  
    初期設定パラメーターのカスタマイズ [243](#)  
    ツリー構造 [231](#)  
    定義処理 [238](#)  
    特定の問題 [234](#)  
    パネル  
        タイプ [229](#)  
        定義 [235, 240](#)  
    パネルの定義 [240](#)  
    複数システムでの [236](#)  
SDFCONF コマンド [240](#)  
SFM [120](#)  
SMF データ・セット処理 [168](#)  
SMF レポート、DB2 への書き込み [66](#)  
SPIN パラメーター [52](#)  
SUBSAPPL [20](#)  
SUBSJOB [20](#)  
SUBSTYPE [20](#)  
SVC ダンプ処理 [168](#)  
SYSLOG 処理 [168](#)  
syslogd メッセージ、トラップ [92](#)  
SYSPLEX 結合データ・セット [125](#)  
SYSPLEX ポリシー項目  
    定義 [125](#)

## T

TAF フルスクリーン・セッション  
    定義 [134](#)  
TCP ポート・モニター、z/OS UNIX 自動化 [87](#)  
TRAP AND WAIT 処理 [78](#)

## U

UNIX syslogd メッセージ、トラップ [92](#)  
UNIX syslogd メッセージのトラップ [92](#)  
UNIX システム・サービス、統合 [83](#)  
UNIX 自動化  
    自動化リソース [85](#)  
    セットアップ [83](#)  
    セットアップ例 [88](#)  
    デバッグ [94](#)  
    ヒント [92](#)  
    ファイル・モニター [87](#)

UNIX 自動化 (続き)  
    プロセス・モニター [86](#)  
    TCP ポート・モニター [87](#)

UNIX リソース  
    始動定義と停止定義 [87](#)  
    定義 [84](#)  
    のカスタマイズ [83](#)  
    モニター・ルーチン [84](#)

USS リソース  
    自動化 [83](#)  
USS リソースを自動化する方法 [83](#)

## V

VM ターゲット・システム、カスタマイズ [81](#)  
VM 第 2 レベル・システム・サポート [79](#)  
VOST の管理、CICS モニター [40](#)  
VSE ゲスト・ターゲット・システム [79](#)  
VSE ターゲット・システム、カスタマイズ [81](#)  
VTAM アプリケーション、SA z/OS への定義 [137](#)

## W

WTO(R)  
    自動化ルーチン  
        削除、SDF からの処理済み WTO(R) の [169](#)  
        処理済み、SDF からの削除 [169](#)  
WTO(R) バッファ [116](#)  
WTOR 処理 [145](#)  
WTOR(R) バッファ不足  
    リカバリー、使用可能にする [126](#)

## Z

z/OS UNIX アプリケーション  
    インフラストラクチャーの概要 [83](#)  
z/OS UNIX システム・サービス、統合 [83](#)  
z/OS UNIX システム・サービスの統合 [83](#)  
z/OS UNIX 自動化  
    自動化リソース [85](#)  
    セットアップ [83](#)  
    セットアップ例 [88](#)  
    デバッグ [94](#)  
    ヒント [92](#)  
    ファイル・モニター [87](#)  
    プロセス・モニター [86](#)  
    TCP ポート・モニター [87](#)  
z/OS UNIX 自動化のセットアップ  
    例 [88](#)  
z/OS UNIX リソース  
    始動定義と停止定義 [87](#)  
    定義 [84](#)  
    のカスタマイズ [83](#)  
    モニター・ルーチン [84](#)  
z/OS UNIX リソースのカスタマイズ [83](#)  
z/OS UNIX リソースの定義 [84](#)  
z/OS システム、GDPS 環境でのシャットダウン [141](#)





SC43-1378-02

