

**IBM Fault Analyzer for z/OS**  
**ユーザーズ・ガイドおよびリファレンス**  
**V15.1.5**

## 注

本書および本書で紹介する製品をご使用になる前に、[通知 ページ dcclix](#)に記載されている情報をお読みください。

## 本書に関する注意事項

本書(2024年6月発行)は、新しい版で明記されていない限り、IBM® Fault Analyzer for z/OS® のバージョン 15 リリース 1 モディフィケーション・レベル 5 (プログラム番号 5755-A02)、およびそ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適用されます。

改訂日: 2024-06-14

# 目次

本書について.....	xii	Fault Analyzer リアルタイム・レポートの結 合.....	37
本書の対象読者.....	xii	リアルタイム・レポートの SYSOUT クラスを制 御.....	37
本書の編成.....	xii	リアルタイム・レポートの抑止.....	38
構文図の見方.....	xii	SYSLOG 要約.....	38
変更の要約.....	xvi	プログラム SNAP インターフェース (IDISNAP) の使 用.....	39
2024 年 6 月 (V15R1M5 APAR PH60313).....	xvi	IDISNAP 呼び出し.....	39
2024 年 3 月 (V15R1M4 APAR PH57736).....	xvi	Java try-catch ブロックからの Fault Analyzer の起 動.....	47
2023 年 10 月 (V15R1M3 APAR PH55151).....	xvi	Java ダンプ・イベントからの Fault Analyzer の起 動.....	50
2023 年 6 月 (V15R1M2 APAR PH53394).....	xvi	Fault Analyzer Java ラッパー・ユーティリティーの使 用.....	50
2023 年 3 月 (V15R1M1 APAR PH50826).....	xvii	Fault Analyzer JVMTI エージェントの使用.....	52
2022 年 10 月 (V15R1M0).....	xviii	ダンプ登録処理.....	53
<b>第 1 部. Fault Analyzer の使用.....</b>	<b>19</b>	リアルタイム除外処理.....	54
<b>第 1 章. 概要.....</b>	<b>20</b>	重複障害処理.....	56
分析エンジン.....	20	リカバリー障害記録.....	57
分析プロセス.....	20	RFR ダンプ・タイトル.....	59
リアルタイム異常終了分析.....	20	Fault Analyzer における SLIP、COMP=0C4 の使 用.....	59
リアルタイム SNAP 分析.....	22	拡張ミニダンプ・データ・セット (XDUMP).....	59
障害再分析.....	23	異常終了トラップは障害分析を回避する可能性があり ます.....	60
バッチ再分析.....	24	異常終了ジョブ JCL のキャプチャーおよび表示.....	60
対話式再分析.....	25	<b>第 3 章. Fault Analyzer ISPF インターフェース.....</b>	<b>62</b>
障害履歴・ファイル.....	25	インターフェースの起動.....	63
関連付けられたダンプ・データ・セット.....	26	ISPF 分割画面のサポート.....	63
履歴・ファイル・データ・セット内の特殊 メンバー.....	26	Fault Entry List 画面.....	63
Fault Analyzer サポートされるアプリケーション環 境.....	27	ビューの使用.....	66
バインド関連の従属関係.....	30	表示される履歴・ファイルまたはビューの 変更.....	66
障害分析のための既存プログラムの設定.....	30	Fault Entry List の列構成.....	71
必要な追加の領域サイズ.....	30	障害項目のソートと突き合わせ.....	78
コンパイラ・リストまたはサイド・ファイルの選択 基準.....	31	障害を突き合わせて選択する追加の方法.....	83
言語環境プログラム CEEWUCHA 異常終了の特殊な処 理.....	31	特定の障害へのアクションの適用.....	87
Fault Analyzer で使用される WTO 宛先コードと記述子 コード.....	31	履歴・ファイルのプロパティ.....	89
CICS Storage Accounting Area (SAA) オーバーレイ・ア シスタンス.....	32	新しい履歴・ファイルの割り振り.....	93
<b>第 2 章. リアルタイム分析.....</b>	<b>33</b>	障害履歴・ファイル設定の変更.....	94
ダンプの抑止.....	33	履歴・ファイル・アクセス情報のリセッ ト.....	98
Fault Analyzer および CICS グローバル・ユーザ ー出口.....	34	障害項目情報の最新表示.....	98
障害履歴・ファイルの選択.....	34	障害項目の有効期限の制御.....	100
オプションによるリアルタイム分析の制御.....	35	アクション・バーのプルダウン・メニュー.....	100
JCL DD ステートメント.....	36	コマンド.....	104
JCL DD ステートメントによるリストのポインティ ング.....	36	CE.....	104
リアルタイム分析レポート.....	37	CICSD.....	105

CICSLINK.....	105	ヒストリー・ファイル項目のコピー.....	143
CICSSTG.....	105	ヒストリー・ファイル項目の移動.....	145
CMDS.....	105	ヒストリー・ファイル項目の送信.....	145
COLS.....	106	障害項目のパッケージ化.....	146
COPY.....	106	セキュリティ上の考慮事項.....	146
CUROPTS.....	106	64 ビット・アドレスの指定.....	147
DISASM.....	107	対話式 IDIS サブシステム・インターフェースの使 用.....	147
DSECT.....	107	管理されているヒストリー・ファイル.....	148
DUPS.....	107	除外されたヒストリー・ファイル.....	149
EDIT.....	107	選択されたヒストリー・ファイル.....	150
EXEC.....	108	現在有効なオプションの表示.....	152
FIND.....	108	ユーザー・ノートのリカバリー.....	152
IDISINFO.....	111	<b>第 4 章. バッチ再分析の実行.....</b>	<b>154</b>
INFO.....	111	バッチ再分析オプション.....	154
JCL.....	111	バッチ再分析の開始.....	159
LOOKUP.....	112	バッチ再分析に使用されるデータ・セット.....	159
MATCH.....	113	独自のバッチ再分析ジョブの作成.....	160
NEXT.....	113	<b>第 5 章. 対話式再分析の実行.....</b>	<b>162</b>
NOTE.....	113	対話式再分析オプション.....	162
NOTELIST.....	114	対話式再分析の開始.....	170
PREV.....	114	状況ポップアップ画面.....	170
QUIT.....	114	対話式レポートについての一般的な解説.....	171
REFRESH.....	115	対話式レポートの終了.....	173
RESET.....	115	基本オプション: Synopsis.....	174
RPTFIND.....	115	基本オプション: イベント・サマリー.....	175
RUNCHAIN.....	115	詳細なイベント情報.....	176
SHOW.....	116	基本オプション: ファイルを開く.....	179
SHOWFREE.....	116	基本オプション: CICS 情報.....	182
SIT.....	117	CICS Control Blocks.....	183
STCK.....	117	CICS Transaction Storage Summary.....	183
STGMAP.....	117	CICS Transaction Storage.....	183
VIEWS.....	117	Last CICS 3270 Screen Buffer.....	184
Fault Analyzer プリファレンス.....	117	Summarized CICS Trace.....	185
BatchOpts.....	120	CICS Trace Formatting.....	186
保管レポートの表示.....	125	CICS タスク・トレース・テーブル.....	192
ブランク行の追加または除去.....	127	CICS Recovery Manager.....	192
ヘルプ・テキストの追加または除去.....	127	CICS Levels, Commareas, and Channels.....	192
優先フォーマット幅の設定.....	128	CICS モニター・データ.....	193
ユーザー選択メッセージまたは異常終了コードの説明 の表示.....	128	CICS イベント・プログラム情報.....	193
ファイルへの対話式画面のコピー.....	130	CICS システム初期化パラメーター.....	194
製品の著作権、ライセンスおよびバージョン情報の表 示.....	131	基本オプション: メッセージ.....	194
ヒストリー・ファイル項目の削除.....	131	基本オプション: DB2 Information.....	194
障害項目のロック.....	132	基本オプション: IMS Information.....	197
障害項目情報の表示.....	134	基本オプション: ストレージ域.....	203
障害項目の重複ヒストリーの表示.....	139	基本オプション: Java Information.....	204
重複障害項目の展開と縮小.....	142	基本オプション: Language Environment Heap Analysis.....	204

基本オプション: MTRACE レコード.....	205	<b>第 7 章. CICS 補助トレース・データ・セットのフォーマット設定.....</b>	<b>260</b>
基本オプション: Abend Job Information.....	206	CICS 補助トレース・データ・セットの選択.....	260
基本オプション: ユーザー・ノート.....	207	CICS トレース選択パラメーターの指定.....	261
基本オプション: Fault Analyzer オプション.....	207	<b>第 8 章. Java 分析の実行.....</b>	<b>262</b>
関連したストレージ域の表示.....	208	Java ダンプ・データ・セットの選択.....	262
16 進値列の非表示.....	209	Java 障害項目の再分析.....	263
レベル 88 項目の縮小表示.....	210	対話式レポートへの Java 情報の表示.....	264
すべての COBOL ベース・ロケーターを表示.....	211	Fault Analyzer イベント・リストの Java イベント置換.....	269
メッセージおよび異常終了コードの拡張表示.....	212	Java レポートの制限の情報.....	271
ソース・コードの表示.....	212	Java アプリケーションのサンプル.....	271
据え置きブレークポイント機能.....	215	サンプル 1 (IDISJAV1): バッチで Java を起動する Enterprise PL/I.....	271
保管場所の表示.....	215	サンプル 2 (IDISJAV2): バッチで Java を起動する Enterprise COBOL.....	272
データ域の表示.....	217	サンプル 3 (IDISJAV3): バッチで Java を起動する 64 ビット Enterprise PL/I.....	272
ユーザー・ノートの作成と管理.....	218	サンプル 4 (IDISJAV4): IMS Java バッチ処理例.....	272
CICS トランジション・ストレージの表示 (CICSSTG).....	223	サンプル 5 (IDISJAV5): バッチで Java を起動する C++ プログラム.....	273
アドレス・スペース・ストレージ・マップの表示 (STGMAP).....	224	サンプル 6 (IDISJAV6): Fault Analyzer ラッパー・ユーティリティで Java バッチ・アプリケーションを呼び出し.....	273
PSW 情報の表示.....	225	<b>第 9 章. Fault Analyzer レポート.....</b>	<b>275</b>
DSECT 情報を使用したストレージ域のマッピング.....	227	汎用レポート情報.....	275
IDIDSECT 連結.....	230	最上位の異常終了コード.....	275
DSECT データ・セットの索引作成 (\$DINDEX メンバー).....	230	オープン・ファイルのレコード情報.....	275
チェーン・データ域の表示.....	231	COBOL で抑止されるコピーブック.....	277
オブジェクト・コードの逆アセンブル.....	234	主なレポート・セクション.....	278
STORE CLOCK 値の変換.....	236	Prolog セクション.....	278
ユーザー固有レポートのフォーマット.....	237	Synopsis セクション.....	278
コンパイラー・リストまたはサイド・ファイルのプロンプト.....	239	Summary セクション.....	278
プロンプトの制御.....	244	Event Details セクション.....	282
対話式再分析に使用されるデータ・セット.....	244	System-Wide Information セクション.....	284
リフレッシュ処理.....	244	Abend Job Information セクション.....	284
COBOL Explorer.....	246	Options セクション.....	284
COBOL Explorer の例.....	246	Epilog セクション.....	284
据え置きブレークポイント機能.....	250	レポート例.....	284
<b>第 6 章. CICS システム異常終了ダンプ分析の実行.....</b>	<b>251</b>	<b>第 10 章. Fault Analyzer ヒストリー・ファイルにアクセスするための非 ISPF インターフェースの使用.....</b>	<b>286</b>
CICS システム異常終了分析用のオプションの設定.....	251	IBM Fault Analyzer プラグイン (Eclipse 用).....	286
ユーザー出口.....	251	以下の環境での対話式再分析の実行 CICS®.....	286
CICS ダンプ・データ・セットの選択.....	251	<b>第 II 部. Fault Analyzer インストールおよび管理.....</b>	<b>287</b>
分析対象のアドレス・スペースの選択.....	253	<b>第 11 章. 障害分析の前のバージョンからのマイグレーション.....</b>	<b>288</b>
CICS システム異常終了対話式レポートの表示.....	254	V14.1 から V15.1 へのマイグレーション.....	288
ファースト・パス・ナビゲーション.....	255	V13.1 から V14.1 へのマイグレーション.....	289
オプション 1: Synopsis.....	255	V12.1 から V13.1 へのマイグレーション.....	289
オプション 2: 異常終了ジョブ情報.....	256	V11.1 から V12.1 へのマイグレーション.....	290
オプション 3: CICS System Information.....	256		
オプション 4: 有効になっているオプション.....	258		
ヒストリー・ファイル項目の作成.....	259		

V10.1 から V11.1 へのマイグレーション.....	290	Fault Analyzer IDISCMDS コマンド・テーブルを使用可能にする.....	328
V9.1 から V10.1 へのマイグレーション.....	291	ISPF 選択パネルの更新.....	329
<b>第 12 章. Fault Analyzer のカスタマイズの準備.....</b>	<b>292</b>	ISPF 3.4 の行コマンドを使用した Fault Analyzer の起動.....	329
Fault Analyzer のインストールおよびカスタマイズのチェックリスト.....	292	SDSF からの Fault Analyzer の起動 (SFA コマンド).....	331
インストール完了後のライブラリー名.....	296	カーソル選択を使用した LOOKUP コマンドの起動 (LOOKC コマンド).....	331
推奨されるストレージ.....	297	ISPF インターフェースのデフォルトを新規ユーザーに提供.....	331
Fault Analyzer を呼び出すための出口.....	299	インストール固有のバッチ再分析 JCL 制御ステートメントを提供.....	332
非 CICS トランザクション異常終了の場合の呼び出し.....	299	Fault Analyzer レポートの表示域の拡大.....	332
CICS トランザクション異常終了の場合の呼び出し.....	303	<b>第 16 章. USERMOD を使用した Fault Analyzer のカスタマイズ.....</b>	<b>334</b>
SVC ダンプ登録.....	303	呼び出される Fault Analyzer の有効化.....	334
以下の呼び出しに必要となる Language Environment オプション Fault Analyzer.....	304	MVS 変更オプション/抑止ダンプ出口のインストール IDIXDCAP.....	334
非 CICS 異常終了に必要な LE オプション.....	304	言語環境プログラム異常終了出口 (IDIXCEE または IDIXCE64) の有効化.....	334
CICS 異常終了に必要な LE オプション.....	304	非言語環境プログラム・ランタイムを使用するアプリケーションの操作.....	335
類似したサード・パーティー製品を使用した Fault Analyzer の実行.....	305	CA-Panexec からのロード・モジュールの取得.....	336
MVS ダンプ・データ・セットのサイズ.....	305	<b>第 17 章. Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ.....</b>	<b>338</b>
アプリケーション処理のエラー条件.....	305	代替 parmlib データ・セットを IDICNFxx(CNFDSN) に指定.....	340
Fault Analyzer SLIP トラップの設定.....	306	デフォルト・リカバリー障害記録 IEATDUMP データ・セット名 (RFRDSN) の変更.....	341
<b>第 13 章. Fault Analyzer の稼働環境のカスタマイズ.....</b>	<b>307</b>	CICS XDUREQ グローバル・ユーザー出口 (CICSNOA) を使用して起動された場合の Fault Analyzer のアクションの変更.....	341
Fault Analyzer モジュールの有効化.....	307	代替セキュリティ・サーバー・テスト・データ・セット名の指定 (SSCHKDSN).....	342
Fault Analyzer プログラムへのプログラム制御アクセスの定義.....	308	デフォルトの拡張ミニダンプ (XDUMP) データ・セット名パターン (XDUMPPDSN) の変更.....	342
履歴・ファイル設定の変更の制限.....	308	コピーされた SDUMP データ・セット名パターンの指定 (SDUMPPDSN).....	342
メッセージおよび異常終了コード説明リポジトリーのセットアップ.....	309	IBM Application Delivery Foundation for z/OS Common Components オプションを無視 (NOIPVOPT).....	343
リカバリー障害記録データ・セット・アクセスの管理.....	310	前の出口が RC=8 の場合に IDIXDCAP リアルタイム分析を抑止 (PREVEXIT).....	343
SDUMP リカバリー障害記録データ・セット.....	311	IDIOPTLM データ・セット名のシンボル置換.....	343
TDUMP リカバリー障害記録データ・セット.....	311	<b>第 18 章. ヒストリー・ファイルのセットアップ.....</b>	<b>345</b>
XDUMP データ・セット・アクセスの管理.....	313	割り振られるヒストリー・ファイルのサイズの決定.....	345
コピーされた SDUMP データ・セット・アクセスの管理.....	315	ヒストリー・ファイルへの PDS または PDSE の割り振り.....	345
Fault Analyzer の IFAPRDxx parmlib メンバーへの登録.....	317	ヒストリー・ファイルの 2 次スペースの割り振り.....	346
<b>第 14 章. Fault Analyzer IDIS サブシステムの使用.....</b>	<b>319</b>	PDS と PDSE の間でのヒストリー・ファイルの交換.....	347
シスプレックス全体にわたるサブシステムの相互通信.....	320	AUTO で管理される PDSE ヒストリー・ファイル.....	347
ヒストリー・ファイル \$\$INDEX データのキャッシュ.....	320	PDSE ヒストリー・ファイルのサイズの変更.....	349
IDIS サブシステムの始動.....	321		
IDIS サブシステム・ストレージ要件.....	324		
IDIS DB2 のサブシステム要件.....	324		
IDIS Java のサブシステム要件.....	325		
移行された密結合の関連ダンプ・データ・セットを削除するための IDIS サブシステム要件.....	326		
IDIS サブシステムの停止.....	326		
<b>第 15 章. ISPF 環境の変更.....</b>	<b>328</b>		
ISPF データ・セットの割り振り.....	328		

以下へのヒストリー・ファイル名の指定: Fault Analyzer.....	349	IDIDATST サイド・ファイルの使用可能性テスト・ユーティリティー.....	393
ビューのセットアップ.....	350	ISPF パック・コンパイラー・リスト.....	394
デフォルトの列レイアウトの指定.....	351	Java ソース情報の提供 Fault Analyzer.....	395
最初の障害項目選択基準の指定.....	351	例: IDIJAVA DD および IDIJAVA IDI_OPTS の使用.....	396
XFACILIT リソース・クラスを使用しない場合のビューに関する考慮事項.....	352	最適なデバッグのための Java のコンパイル.....	397
共用 DASD のない MVS システム全体でのヒストリー・ファイルの管理.....	353	共通 Java ビルド・ツールを使用したデバッグ情報の生成.....	398
自動化インプリメンテーション.....	353	<b>第 21 章. CICS 環境のカスタマイズ.....</b>	<b>400</b>
オンデマンドの実装.....	361	Language Environment for CICS を呼び出すための構成 Fault Analyzer.....	401
シスプレックス全体でのヒストリー・ファイルの共用.....	364	必要なプログラムの CICS システムへの定義.....	401
ヒストリー・ファイル障害項目のアクセスの管理.....	365	必要なプログラムの始動 PLT への追加.....	402
ヒストリー・ファイル障害項目に対しての XFACILIT リソース・クラスの使用.....	366	必要なプログラムのシャットダウン PLT への追加.....	402
<b>第 19 章. サイトのデフォルト・オプションの設定および変更.....</b>	<b>372</b>	CICS トランザクション異常終了分析の動的制御の有効化.....	403
parmlib メンバー IDICNFxx.....	372	CFA を使用した現在分析中のタスクでの FORCEPURGE の実行.....	403
Exclude 処理による分析対象のジョブの制御.....	374	SVC ダンプ・スクリーニング.....	403
高速 Exclude オプション処理.....	375	定義ジョブの例.....	404
レポートの詳細レベルの制御.....	376	CICS トランザクション異常終了分析の制御.....	405
ミニダンプのサイズの制限.....	376	CICS® 端末からの CFA の使用.....	405
リストおよび REXX exec ライブラリーのポインティング.....	376	MVS コンソールからの CFA の使用.....	408
マイナーな SYSLOG メッセージの抑止.....	376	CICS Web サービス・トランザクション.....	410
Fault Entry List 画面のカスタマイズ.....	376	トランザクション異常終了分析が DUMP(NO) に抑止されないことを確認.....	414
国/地域別環境の指定.....	376	CICS NoDup(CICSFAST) オーバーライド・アセンブラー出口 (IDINDFUE).....	414
<b>第 20 章. コンパイラー・リストまたは Fault Analyzer サイド・ファイルの提供.....</b>	<b>377</b>	呼び出し.....	414
COBOL 報告書作成プログラム・プリコンパイラー.....	378	CICS トレース考慮事項.....	416
IDILANGX の必須コンパイラー・オプション.....	378	LE による CICS トレースの循環を回避.....	417
TEST オプションの考慮事項.....	381	IDI_OPTS DDname を使用した CICS オプションの指定.....	417
DEBUG オプションの考慮事項.....	381	言語環境プログラム異常終了の考慮事項.....	418
COBOL NOTEST(DWARF、SOURCE) 使用時の制限.....	382	CICS ユーザー・キー・オープン TCB (L9 TCB) での実行時の異常終了の取り込み.....	418
PL/I TEST (NOSEPARATE NOSOURCE) 使用時の制限.....	382	MVS ポスト・ダンプ出口のインストール IDIXTSEL.....	418
コンパイラー・リストまたはサイド・ファイルの命名.....	382	CICS トランザクション異常終了分析パフォーマンスの最大化.....	419
Fault Analyzer 用の CSECT の命名.....	383	XEIIN グローバル・ユーザー出口の実装.....	419
コンパイラー・リストまたはサイド・ファイルの検索.....	384	3270 画面バッファ・キャプチャーの無効化.....	421
IDITRACE 情報.....	387	<b>第 22 章. DB2 環境のカスタマイズ.....</b>	<b>422</b>
IDIXSFOR コンパイラー・リスト/サイド・ファイルの検索出口およびオーバーライド出口の使用.....	388	DB2 のバインディング.....	422
IDIXSFOR 呼び出し.....	388	DB2 および言語環境プログラム.....	422
例 (アセンブラー).....	391	Fault Analyzer DB2 パフォーマンスの向上.....	422
コンパイラー・リストおよびサイド・ファイルの属性.....	391	<b>第 23 章. IMS 環境のカスタマイズ.....</b>	<b>424</b>
CSECT マッピングに IDIRLOAD DDname の使用.....	392	IMS および言語環境プログラム.....	424
		<b>第 24 章. Fault Analyzer 日本語フィーチャーのカスタマイズ.....</b>	<b>425</b>



ISPF データ・セットの割り振り.....	425	<b>第 27 章. アプリケーション固有の説明の指定.....</b>	<b>455</b>
各国語サポートの設定.....	426	ユーザー定義メッセージの説明.....	455
<b>第 25 章. Fault Analyzer のカスタマイズの検証.....</b>	<b>427</b>	ユーザー定義の異常終了コードの説明.....	456
XDUMP の使用の検証.....	427	ユーザー定義プログラムの説明.....	457
アセンブラーによる Fault Analyzer の使用の検証.....	427	<b>第 28 章. Fault Analyzer の保守.....</b>	<b>459</b>
COBOL による Fault Analyzer の使用の検証.....	428	<b>第 29 章. Fault Analyzer の無効化.....</b>	<b>461</b>
PL/I による Fault Analyzer の使用の検証.....	429	Fault Analyzer を一時的にアンインストール.....	461
C による Fault Analyzer の使用の検証.....	430	IFAPRDxx parmlib メンバーを使用して Fault Analyzer	
IDIXCEE Language Environment 出口の有効化の検		をオフにする.....	462
証.....	431	JCL スイッチ (Fault Analyzer) を使用した IDIOFF の停	
CICS での Fault Analyzer のカスタマイズの検証.....	432	止.....	463
CICS IVP: 0C1 in program IDIXFA.....	432	環境変数 (Fault Analyzer) を使用した _IDL_OFF の停	
CICS IVP: EXEC CICS® DUMP		止.....	463
DUMPCODE(FAD1).....	433	<b>第 30 章. ユーザー出口を使用した Fault Analyzer のカスタ</b>	<b>マイズ.....</b>
CICS IVP: EXEC CICS® ABEND		起動パラメーター.....	467
ABCODE(FLT1).....	433	グローバル環境データ域 (ENV).....	467
CICS IVP: EXEC CICS ABEND ABCODE(FLT2).....	433	ユーザー出口タイプに固有のデータ域.....	467
DB2 による Fault Analyzer の使用の検証.....	433	サポートされる出口プログラム言語.....	467
C プログラムを使用する方法.....	434	共通のユーザー出口を呼び出している出口タイプの判	
COBOL プログラムを使用する方法.....	435	別.....	468
ISPF による Fault Analyzer の使用の検証.....	438	データ域のバージョンの検査.....	469
リカバリー障害記録セットアップの検証.....	438	診断トレース.....	469
ダンプ登録の検証 (IEAVTSEL).....	439	ユーザー出口パラメーター・リスト値のトレ	
<b>第 26 章. ヒストリー・ファイルの管理 (IDIUTIL ユーティ</b>	<b>リ</b>	ス.....	470
<b>ティ</b> ).....	<b>441</b>	REXX EXEC のトレース.....	474
IDIUTIL 制御ステートメント.....	442	障害分析ユーザー出口タイプの説明.....	474
FILES 制御ステートメント.....	442	分析制御ユーザー出口.....	475
LISTHF 制御ステートメント.....	442	分析制御ユーザー出口 (MVS SVC ダンプ登	
LISTHFDUP 制御ステートメント.....	443	録).....	479
DELETE 制御ステートメント.....	445	コンパイラー・リスト読み取りユーザー出	
SETFAULTPREFIX 制御ステートメント.....	446	口.....	480
SETMAXFAULTENTRIES 制御ステートメント.....	447	メッセージおよび異常終了コード説明ユーザー出	
SETMINFAULTENTRIES 制御ステートメント.....	447	口.....	485
IMPORT 制御ステートメント.....	448	フォーマット・ユーザー出口.....	491
EXPORT 制御ステートメント.....	449	終了処理ユーザー出口.....	494
EXITS 制御ステートメント.....	450	終了処理ユーザー出口 (障害項目リフレッ	
例.....	451	シュ).....	498
例 1. ヒストリー・ファイル項目のリスト.....	451	通知ユーザー出口.....	499
例 2. ヒストリー・ファイル異常終了インスタンス		通知ユーザー出口 (MVS SVC ダンプ登録).....	507
のリスト.....	452	IDIUTIL バッチ・ユーティリティー・ユーザー出口タイ	
例 3. 日付によるヒストリー・ファイル項目の削		プの説明.....	508
除.....	452	IDIUTIL インポート・ユーザー出口.....	508
例 4. 使用率によるヒストリー・ファイル項目の削		IDIUTIL 削除ユーザー出口.....	511
除.....	452	IDIUTIL ListHF ユーザー出口.....	512
例 5. ヒストリー・ファイル障害接頭部文字の変		IDIUTIL ListHFDUP ユーザー出口.....	517
更.....	452	ユーザー出口 REXX コマンド.....	519
例 6. 自己保持ヒストリー・ファイルの作成.....	453	EVALUATE コマンド.....	519
例 7. ヒストリー・ファイル項目のインポー		IDIALLOC コマンド.....	523
ト.....	453	IDIDTEST コマンド.....	527
IDIUTIL バッチ・ユーティリティー・ユーザー出口		IDIDSECTdsn コマンド.....	528
の例.....	454		

IDIDSNTEST コマンド.....	529	ユーザー・オプション・モジュール	
IDIEventInfo コマンド.....	530	IDICNFUM.....	570
IDIFREE コマンド.....	531	_IDI_OPTSFILE 環境変数.....	571
IDIGET コマンド.....	532	User options file IDIOPTS.....	571
IDIModQry コマンド.....	532	JCL EXEC ステートメント PARM フィールド.....	572
IDIPUT コマンド.....	533	_IDI_OPTS 環境変数.....	572
IDIRegisterFaultEntry コマンド.....	534	オプションに対する変更が有効な場合.....	572
IDIWRITE コマンド.....	535	オプションの説明.....	573
IDIWTO コマンド.....	537	AdditionalIDIOffDD.....	573
LIST コマンド.....	538	CallEqaueDat.....	573
NOTE コマンド.....	540	CICSDumpTableExclude.....	574
フォーマット・タグ.....	542	CICSTraceMax.....	575
ADDR (アドレス).....	544	CICSTranAnalysisUser.....	575
AREA (領域).....	544	DataSets.....	576
DD (定義説明).....	545	DeferredReport.....	584
DATA (データ).....	545	Detail.....	586
DL (定義リスト).....	545	DumpDSN.....	587
DT (定義語).....	546	DumpRegistrationExits.....	588
DUMP (EBCDIC ダンプ).....	547	ErrorHandler.....	589
DUMPA (ASCII ダンプ).....	548	Exclude/Include.....	590
HP (強調表示されたフレーズ).....	549	Exits.....	595
L (行).....	549	FAISPFopts.....	598
LI (リスト項目).....	549	FaultID.....	605
NOTEL (注釈リスト).....	550	GenerateSavedReport.....	606
P (パラグラフ).....	550	Include.....	607
TH (表の見出し).....	551	JclCapture.....	607
U (下線).....	551	Language.....	607
UL (順序なしリスト).....	552	LangxCapture.....	608
IDIXUFMT ロード・モジュール・フォーマット・ユーザー出口.....	552	Locale.....	609
IDIXUFMT 機能.....	553	LoopProtection.....	609
ADFzCC イベント処理ユーザー出口とのインターフェイス.....	562	MaxMinidumpPages.....	610
FA で提供されるサンプル.....	562	NoDup.....	611
<b>第 31 章. Fault Analyzer ヒストリー・ファイルにアクセスするための非 ISPF インターフェイスのインストール... 563</b>		PDTCCopts.....	620
IBM Fault Analyzer プラグイン (Eclipse 用) のインストール.....	563	PermitLangx.....	621
IBM Application Delivery Foundation for z/OS® Common Components サーバーのカスタマイズ.....	563	PreferredFormattingWidth.....	623
IBM Fault Analyzer プラグイン (Eclipse 用) のインストール.....	564	PrintInactiveCOBOL.....	624
CICS 環境での対話式再分析の使用可能化.....	564	Quiet.....	624
必要な CICS リソース定義の作成.....	565	RefreshExits.....	625
必要な CICS JCL の変更.....	565	RetainCICSDump.....	626
<b>第 III 部. Fault Analyzer 参照情報..... 566</b>		RetainDump.....	627
<b>第 32 章. オプション..... 567</b>		Snapdata.....	628
オプションの指定位置.....	569	Source.....	629
parmlib メンバー IDICNFxx.....	569	SpinIDIREPRT.....	629
		StoragePrintLimit.....	630
		StorageRange.....	631
		SystemWidePreferred.....	632
		UnreferencedPagesValid.....	634
		<b>第 33 章. データ域..... 635</b>	

非 REXX ユーザー出口のバッファ・データ・フォーマット.....	636
データ域の説明.....	636
CTL - 分析制御ユーザー出口パラメーター・リスト.....	636
ENV - 共通出口環境情報.....	650
EPC - End Processingユーザー出口パラメーター・リスト.....	667
LST - Compiler Listing Readユーザー出口パラメーター・リスト.....	668
NFY - 通知ユーザー出口パラメーター・リスト.....	672
UFM - ユーザー出口パラメーター・リストのフォーマット.....	673
UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト.....	687
XPL - Message and Abend Code Explanationユーザー出口パラメーター・リスト.....	689
<b>第 34 章. 戻りコード.....</b>	<b>693</b>
<b>第 35 章. メッセージ.....</b>	<b>695</b>
Fault Analyzer メッセージ.....	695
<b>カスタマイズ済み ISPF インターフェース・フロントエンドの例.....</b>	<b>741</b>
障害ヒストリー・ファイルまたはビュー名.....	742
プログラム名の MATCH.....	743
サンプル・アプリケーションのインストール.....	743
サンプルの機能方法.....	743
<b>Fault Analyzer レポートをダウンロードするための Java API.....</b>	<b>744</b>
<b>画面サイズ調整の技術的詳細.....</b>	<b>753</b>
<b>サポート・リソース.....</b>	<b>754</b>
<b>Fault Analyzer のアクセシビリティ機能.....</b>	<b>755</b>
通知.....	dcclix
用語集.....	dcclxiii
索引.....	767

## 本書について

本書は、ユーザー・アプリケーションの異常終了を分析するための Fault Analyzer のインストールおよび使用方法について説明します。

本書では、以下の名前が使用されています。

### **IBM® Application Delivery Foundation for z/OS (ADFz) 製品ファミリー**

以前は、IBM® Problem Determination Tools (PD Tools) 製品と呼ばれていました。

### **IBM® z/OS® Debugger (z/OS® Debugger)**

IBM® Debug Tool エンジンを含みます。

### **IBM® Developer for z/OS**

旧称は IBM® Rational® Developer for z Systems です。

### **IBM® Explorer for z/OS® and plug-ins for ADFz 製品ファミリー**

旧称は IBM® Problem Determination Tools Studio です。

## 本書の対象読者

本書は、いずれかのサポート言語環境で開発されたアプリケーションの開発または保守を担当するプログラマーを対象としています。Fault Analyzer は、アプリケーションの開発、テストまたは実稼働環境における問題判別に適しています。

Fault Analyzer のインストール方法を知りたいシステム・プログラマーも対象となります。

## 本書の編成

[Fault Analyzer の使用 ページ 19](#) には、Fault Analyzer の概要が説明されています。ここでは、ユーザー・アプリケーションが異常終了した場合の Fault Analyzer の起動方法、および Fault Analyzer のさまざまなオペレーション・モードについて説明しています。

[Fault Analyzer インストールおよび管理 ページ 287](#) は、異常終了を分析するための Fault Analyzer の設定方法について説明しています。また、Fault Analyzer 実行を制御するために使用できるユーザー出口についても解説します。

[Fault Analyzer 参照情報 ページ 566](#) は、オプション、戻りコード、データ域、および Fault Analyzer が発行するメッセージに関する情報を提供します。

サポート・リソースおよび問題解決情報は IBM® Web サイトに関する情報を含み、これは質問に答えて問題を解決するのに役立つことがあります。

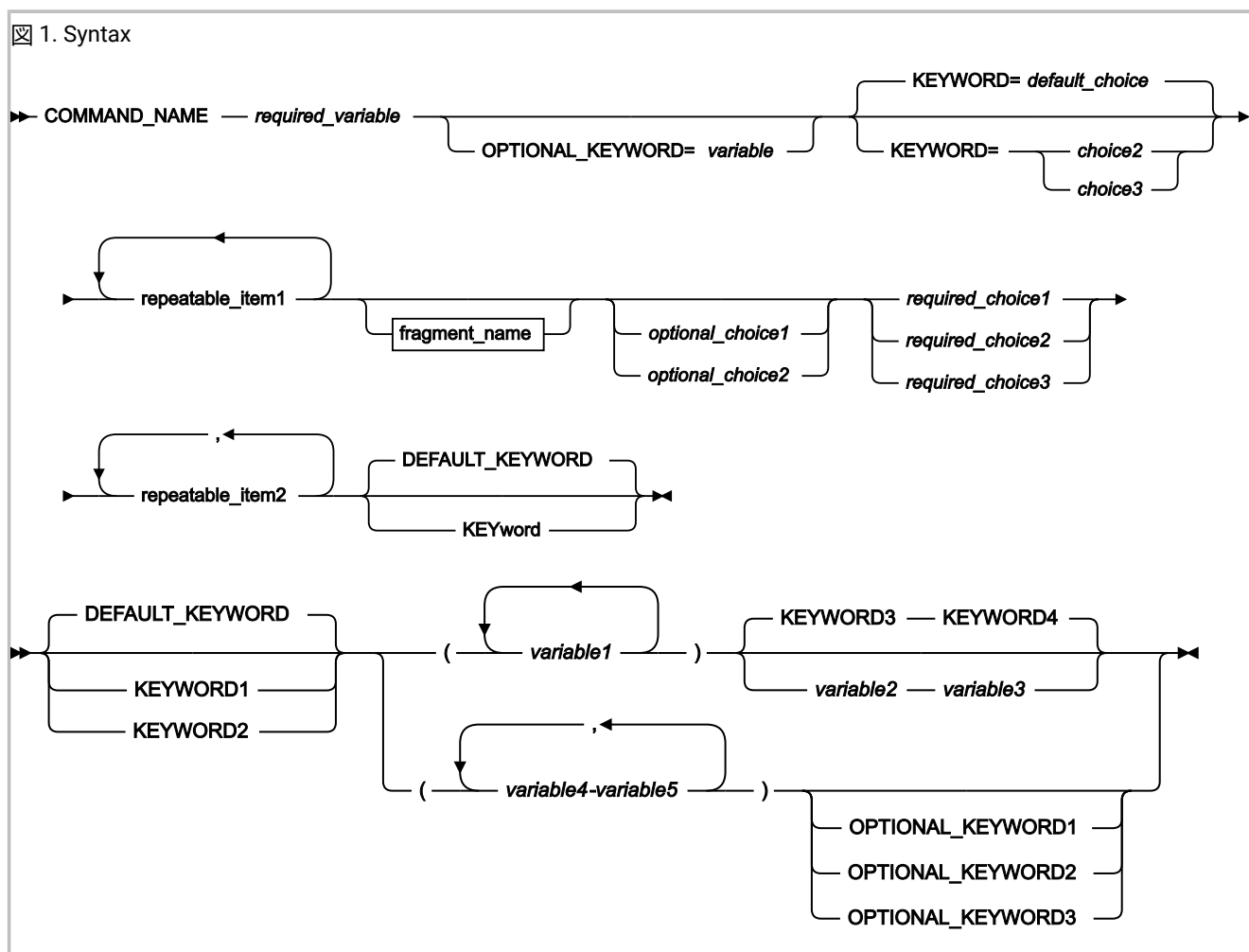
## 構文図の見方

本書で説明するコマンドの構文構造は、構文図で示されています。

[図 1: 構文図の例 ページ xiii](#) に、次のような項目を示すさまざまな表記が含まれた構文図の例を示します。

- 項目がキーワードであるか、変数であるか。
- 項目が必須であるか、オプションであるか。
- 選択が可能であるかどうか。
- 値を指定しない場合、デフォルトを適用するかどうか。
- 項目の反復が可能であるかどうか。

図 1. 構文図の例



構文図を読んで理解するためのヒントをいくつか示します。

#### 読む順序

構文図は、直線のパスをたどって左から右、上から下に読みます。

▶— 記号は、ステートメントの開始を示しています。

▶— 記号は、ステートメントの開始を示しています。

—▶ 記号は、ステートメントが次の行に継続していることを示しています。

▶—記号は、ステートメントが前の行から継続していることを示しています。

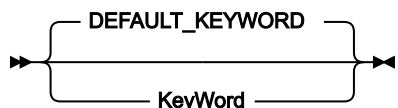
—▶記号は、ステートメントの終了を示しています。

## キーワード

キーワードは、大文字で示します。

▶▶ **COMMAND\_NAME** ▶▶

キーワードの一部を入力するだけで済む場合もあります。キーワードの必須部分は大文字で示されています。



この例では、「KW」または「KEYWORD」と入力できます。

短縮したキーワードまたはキーワード全体を入力する場合、示されたとおりに正確に入力する必要があります。

## 変数

変数は、小文字で示します。変数はユーザーが指定する名前または値を表します。

▶▶ *required\_variable* ▶▶

## 必須項目

必須項目は、水平線 (メインパス) 上に示されます。

▶▶ **COMMAND\_NAME** — *required\_variable* ▶▶

## オプション項目

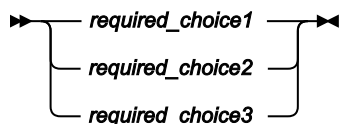
オプション項目は、メインパスの下に示されます。



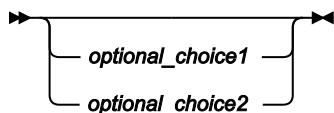
## 項目の選択

複数の項目から選択可能な場合、これらの項目は縦方向に重ねて示されます。

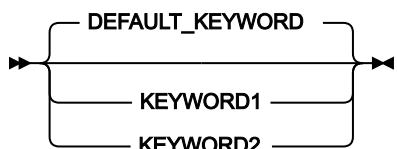
複数の項目から 1 項目を選択する必要がある場合には、項目のいずれかがメインパス上に表示されます。



複数の項目から任意で 1 項目を選択する場合は、縦にスタックされた選択項目全体がメインパスの線よりも下に表示されます。

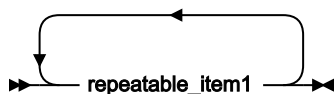


いずれの項目も選択しなかったときにデフォルト値が適用される場合は、デフォルト値をメインパスの上方に示します。

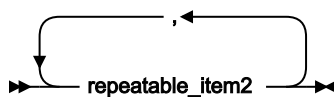


### 反復可能な項目

メインパスの上にある左向きの矢印は、繰り返し可能な項目を示します。

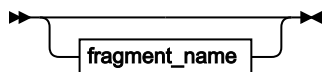


反復可能な項目の間に分離文字 (コンマなど) を指定する必要がある場合は、左に戻る矢印付きの線の中で、指定する必要がある分離文字を示します。

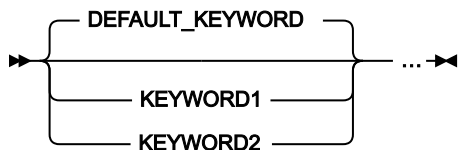


### フラグメント

構文図を読みやすくするために、構文のセクションまたはフラグメントを分けて示すことがあります。



fragment\_name



## 変更の要約

このバージョンの変更について、以下のトピックで説明します。

### 2024年6月 (V15R1M5 APAR PH60313)

本書のこの版には、軽微な補足説明と訂正のほか、前の版に対する以下のような変更が含まれています。

- 新しいメッセージ **IDI0196W** が追加されました。 [IDI0196W ページ 740](#) を参照してください。

### 2024年3月 (V15R1M4 APAR PH57736)

本書のこの版には、軽微な補足説明と訂正のほか、前の版に対する以下のような重要な変更が含まれています:

- 新しいセクション「[2GB から 4GB のアドレスに関する特別な考慮事項](#)」が追加されました。 [ページ 147](#)
- 新しいオプション・フィールド PREVEXIT が、 IDIOPTLM 構成オプション・モジュールに追加されました。詳しくは、 [前の出口が RC=8 の場合に IDIXDCAP リアルタイム分析を抑制 \(PREVEXIT\) ページ 343](#) を参照してください。
- DSECT コマンドと IDIPDSCU ユーティリティーを備えた新しい DDname IDISYSLB の使用方法に関するサポートが追加されました。詳しくは、 [IDISYSLB 連結 ページ 229](#) を参照してください。
- セクション [V14.1 から V15.1 へのマイグレーション ページ 288](#) が更新され、 IDIOPT1 および IDIOPT2 SMP/E USERMOD に関する情報が追加されました。

### 2023年10月 (V15R1M3 APAR PH55151)

本書のこの版には、軽微な補足説明と訂正のほか、前の版に対する以下のような重要な変更が含まれています:

- 障害項目リフレッシュ出口プロンプト画面が変更されました。詳しくは、 [リフレッシュ処理 ページ 244](#) を参照してください。
- IDIHUSRМ メンバーに CICS 異常終了コードを追加する方法の詳細が追加されました。詳しくは、 [ユーザー定義の異常終了コードの説明 ページ 456](#) を参照してください。

### 2023年6月 (V15R1M2 APAR PH53394)

本書のこの版には、軽微な補足説明と訂正のほか、前の版に対する以下のような重要な変更が含まれています:

- 新規オプション UnreferencedPagesValid が提供されました。詳しくは、 [UnreferencedPagesValid ページ 634](#) を参照してください。
- 最大 32 文字のプログラム名のサポートが追加されました。



- 新しいフィールド ENV.POF\_LONG\_PGM\_NAME が追加されました。詳しくは、[607 ページの「ENV - 共通出力環境情報」](#) [ページ 650](#)を参照してください。
- 新しいフィールド LST.LONG\_PROGRAM\_NAME が追加されました。詳しくは、[LST - Compiler Listing Read ユーザー出力パラメーター・リスト](#) [ページ 668](#)を参照してください。
- 新しい列 Program\_(long) が Fault Entry List 画面に追加されました。詳しくは、[選択可能な列](#) [ページ 73](#)を参照してください。
- 新しいセクション [移行された密結合の関連ダンプ・データ・セットを削除するための IDIS サブシステム要件](#) [ページ 326](#) が追加されました。

## 2023 年 3 月 (V15R1M1 APAR PH50826)

本書のこの版には、軽微な補足説明と訂正のほか、前の版に対する以下のような重要な変更が含まれています:

- デフォルトのリカバリー障害記録 IEATDUMP データ・セット名パターンの低位修飾子が '&JOBNAME.' から '&SEQ.' に変更されました。詳しくは、「[リカバリー障害記録データ・セット・アクセスの管理](#) [ページ 310](#)」を参照してください。
- Fault Analyzer が CICS Web サービス・トランザクションを提供するようになりました。[CICS Web サービス・トランザクション](#) [ページ 410](#)を参照してください。
- Fault Analyzer XDUMP はデフォルトで有効になりました。詳しくは、「[拡張ミニダンプ・データ・セット \(XDUMP\)](#) [ページ 59](#)」を参照してください。
- MaxMinidumpPages を超えたときの XDUMP データ・セットへのページのスピル処理のサポートが追加されました。詳しくは、「[XDUMP の使用の検証](#) [ページ 427](#)」を参照してください。
- 新しい列「スピル済み」が **Fault Entry List** 画面に追加されました。詳しくは、[選択可能な列](#) [ページ 73](#)を参照してください。
- XDUMP 構成をテストするための一般的な方法は、「[XDUMP の使用の検証](#) [ページ 427](#)」で提供されています。
- IDIUTIL IMPORT および EXPORT が、順次データ・セットの使用をサポートするようになりました。詳しくは、「[IMPORT 制御ステートメント](#) [ページ 448](#)」および「[EXPORT 制御ステートメント](#) [ページ 449](#)」を参照してください。
- Fault Analyzer Java SVC DUMP タイトルに関する情報が、45 ページの「[Fault Analyzer Java SVC ダンプ・タイトル](#)」に追加されました。
- 以前の非推奨のオプションが削除されました。
  - HistCols。FAISPFopts オプションの HistCols サブオプションを使用してください ([HistCols](#) [ページ 604](#))。
  - InteractiveExitPromptSeconds。この代わりに FAISPFopts オプションの InteractiveExitPromptSeconds サブオプションを使用してください ([InteractiveExitPromptSeconds](#) [ページ 605](#)を参照)。
  - RDZClient。置換なし。
  - UseIDISTime。置換なし。

これらのオプションは、IDICNFxx オプション・メンバーから削除して、IDI0020W メッセージが発行されないようにする必要があります。

- 新しいメッセージが追加されました:

[IDI0194W ページ 695](#)

[IDI0195W ページ 695](#)

## 2022 年 10 月 (V15R1M0)

本書のこのバージョンでは、Fault Analyzer バージョン 15 リリース 1 に適用可能な情報を説明します。このバージョンの変更には、以下が含まれます。

- Fault Analyzer V14.1 から V15.1 へのマイグレーションに関する情報が追加されました ([V14.1 から V15.1 へのマイグレーション ページ 288](#) を参照)。
- GPREGn\_VALID フィールド (n は 0 から 15) が UFM データ域に追加されました。 [UFM - ユーザー出口パラメーター・リストのフォーマット ページ 673](#) を参照してください。
- DUMP および DUMPA フォーマット・ユーザー出口レポート・フォーマット・タグ、および REXX LIST コマンドに対して、追加のオプション・パラメーター ALIGN が提供されます。 [DUMP \(EBCDIC ダンプ\) ページ 547](#)、 [DUMPA \(ASCII ダンプ\) ページ 548](#)、および [LIST コマンド ページ 538](#) を参照してください。
- 新しい IDIUTIL LISTHFDUP 制御ステートメントと、一致する LISTHFDUP ユーザー出口が提供されます。 [LISTHFDUP 制御ステートメント ページ 443](#) および [EXITS 制御ステートメント ページ 450](#) を参照してください。

## 第 I 部. Fault Analyzer の使用

# 第 1 章. 概要

本章は、分析プロセスの概要を示すとともに、Fault Analyzer for z/OS® の他の機能について説明します。

Fault Analyzer の目的は、アプリケーションが異常終了した原因を判別することです。Fault Analyzer はアプリケーションおよびその環境に関する情報を分析し、分析レポートを生成します。レポートには、アプリケーション・コードで問題が記述されており、アプリケーション開発者および保守管理者は低レベルのシステム・ダンプやシステム・レベルのエラー・メッセージを解釈する必要がなくなります。これにより、異常終了の原因を、迅速かつ労力をかけずに知ることができます。

## 分析エンジン

Fault Analyzer は、特定の目的のための分析エンジンをコアにしています。分析が必要な際には、このエンジンが起動されます。分析は、異常終了後に自動的に行われるか、プログラム SNAP インターフェースの場合はアプリケーションにより開始されるか、障害再分析の場合はユーザーにより開始されます。

この分析エンジンは、第一線のソフトウェア設計者、開発者およびテストのデバッグ経験を集合し、カプセル化したエキスパート・システムです。

## 分析プロセス

Fault Analyzer の分析プロセスは、リアルタイム分析から始まります。リアルタイム分析は、異常終了、または SNAP インターフェースへの明示呼び出しのどちらかによって行われ、その後、必要に応じて再分析が行われます。

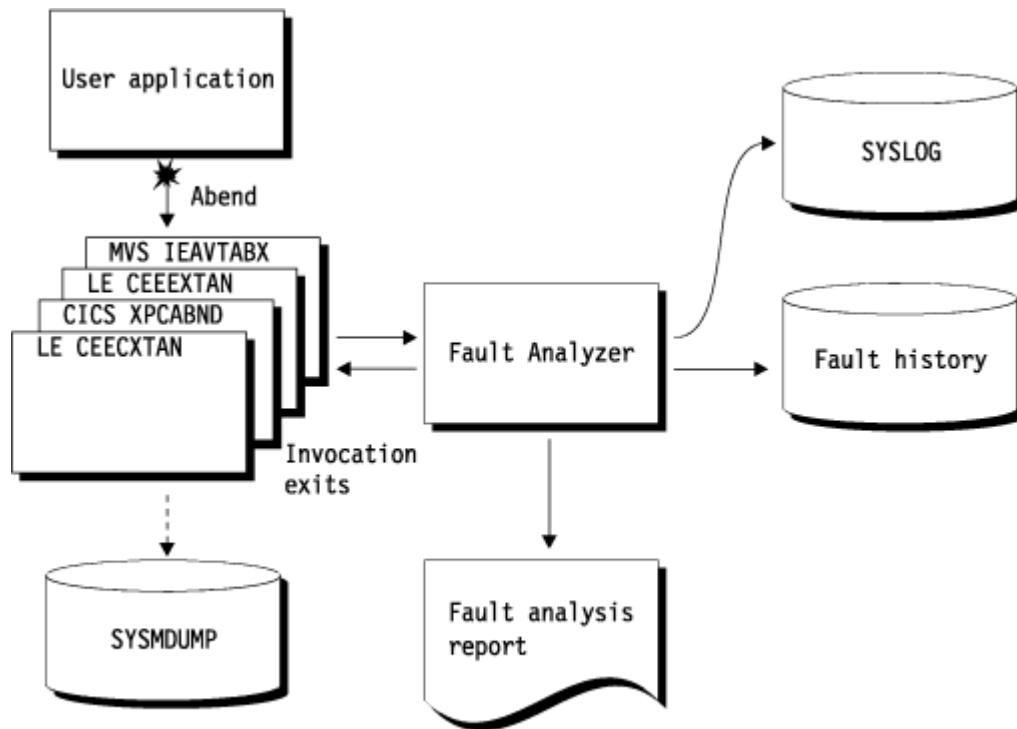
Fault Analyzer により実行可能なさまざまなタイプの分析機能に含まれるプロセスについて、以降のトピックで説明します。

### リアルタイム異常終了分析

プログラムが異常終了すると、異常終了処理 (MVS またはサブシステム) がインターセプトされ、その処理環境に適切な出口を経由して、Fault Analyzer が自動的に起動されます。使用可能な出口のタイプに関する詳細は、[Fault Analyzer を呼び出すための出口 ページ 299](#)を参照してください。

Fault Analyzer は、障害分析処理を実行して、異常終了に関する詳細を障害ヒストリー・ファイルに記録します。Fault Analyzer は、障害分析レポートをジョブに、要約を SYSLOG に書き込みます。また、分析レポートをミニダンプとともにヒストリー・ファイルに保管します。ミニダンプは、分析プロセス中に参照されたすべての仮想記憶域ページのコピーで構成されています。このオペレーション・モードは、「リアルタイム分析」と呼ばれます。 [図 3: リアルタイム異常終了分析 ページ 21](#) に、このプロセスを図示します。

図 3. リアルタイム異常終了分析



リアルタイム障害分析レポート例について、詳細は[レポート例 ページ 284](#)を参照してください。

Fault Analyzer は、指定された TCB での最初の異常終了で呼び出しのみがされるように意図されています。

分析が正常終了したと Fault Analyzer が判断し、異常終了ジョブ・ステップに対して SYSMDUMP、SYSABEND、または SYSUDUMP が指定された場合、Fault Analyzer は MVS にダンプを抑止するように指示を出します。

オプションの指定または JCL の変更を行わなくても、Fault Analyzer のリアルタイム起動について、ヒストリー・ファイルにミニダンプが書き込まれます。ミニダンプが抑止される例外には、次の 3 つがあります。

- ページ数が、インストールで定義された制限を超える場合 ([MaxMinidumpPages ページ 610](#)参照)。
- 障害が別の障害と重複している場合 ([NoDup ページ 611](#)参照)。
- End Processing ユーザー出口がミニダンプの抑止を要求する場合 ([終了処理ユーザー出口 ページ 494](#)参照)。

ミニダンプは、処理環境で発生した障害の再分析を、MVS ダンプも書き込まれたかどうかにかかわらず許可します。

アプリケーションが異常終了しなければ、Fault Analyzer は処理リソースを消費しません。このため、Fault Analyzer は、アプリケーション開発、テスト、または実稼働環境のいずれにも適しています。

アプリケーション開発者および保守管理者に対して低レベルのシステム・ダンプやシステム・レベルのエラー・メッセージを解釈させる代わりに、障害分析レポートには、アプリケーション・コードの形で障害が記述されます。可能な場合は、レ

ポートには異常終了が発生したソース・ステートメントや、(COBOL および PL/I の場合は) ステートメント内で使用されたデータの名前や値が引用されます。

一般的に、異常終了したプログラムの関連のコンパイラー・リスト (またはサイド・ファイル) がオンラインで入手可能な場合、アプリケーションの異常終了は、異常終了に関連したプログラムのソース・ステートメント・レベルまで分析されます (コンパイラー・リストまたはサイド・ファイルを選択する際に使用される基準については、[コンパイラー・リストまたはサイド・ファイルの選択基準 ページ 31](#)を参照してください)。リストが使用できなければ、マシン・インストラクションを逆アセンブルし、プログラム名とオフセットまで、問題が診断されます。

ほとんどの場合、Fault Analyzer が提供し、ジョブ出力に書き込まれる分析レポートで十分であり、問題判別を正しく行うためにこれ以上の障害情報は必要ありません。ただし、異常終了についてのさらに詳細な情報を取り出したい場合は、ISPF インターフェイスを使用してバッチまたは対話式の再分析を開始し、障害の再分析を要求できます。

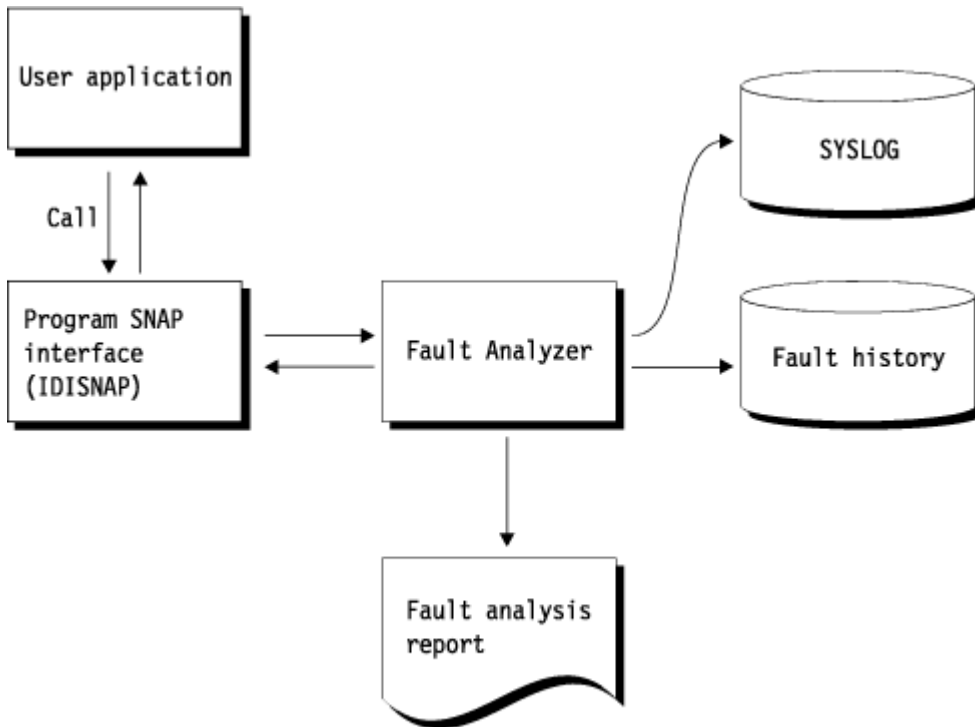
## リアルタイム SNAP 分析

Fault Analyzer が起動される方法を除いて、Fault Analyzer リアルタイム異常終了分析プロセスとリアルタイム SNAP 分析プロセスには、基本的に違いはありません。

プログラム SNAP インターフェイスにより、必要に応じて適切な呼び出しを組み込むことにより、アプリケーション・プログラムが Fault Analyzer を起動できるようになります。この方法で、アプリケーション・プログラマーは、アプリケーションが異常終了しない状態で、現行の環境の分析を取得できます。Fault Analyzer の呼び出しは、アプリケーション・プログラムを中断せず行われるため、アプリケーション・プログラムは、分析後も引き続き実行できます。

[図 4: リアルタイム SNAP 分析 ページ 23](#) は、リアルタイム SNAP 分析プロセスを図示しています。

図 4. リアルタイム SNAP 分析



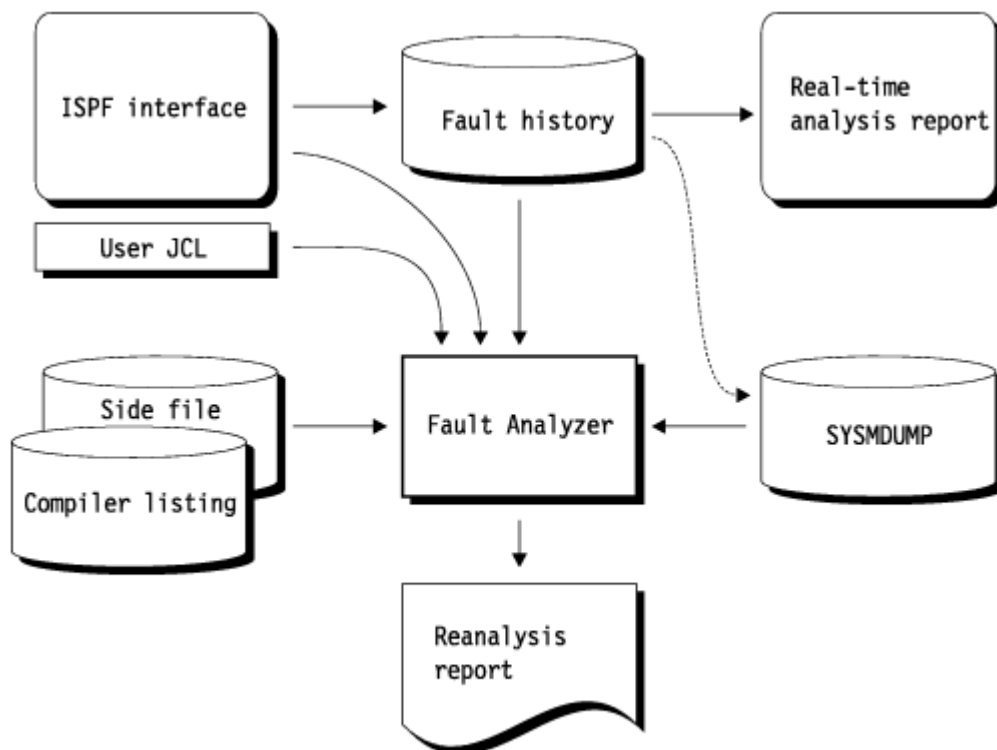
## 障害再分析

再分析プロセスは、次の点を除いて、リアルタイム分析プロセスと本質的に同じです。

- Fault Analyzer は、異常終了プログラムの仮想ストレージの代わりに、保管されたミニダンプ (または関連付けられた MVS ダンプ・データ・セット) から必須情報を取り出します。
- ヒストリー・ファイルは更新されません。
- 要約は SYSLOG に書き込まれません。

図 5: 再分析 ページ 24 は、再分析のプロセスを示しています。

図 5. 再分析



MVS ダンプ・データ・セットが障害項目に関連付けられている場合、再分析、またはミニダンプに含まれていない保管場所のユーザー選択表示で必要となるための、障害履歴項目はダンプ・データ・セット名を保持します。ダンプ・データ・セットを使用できない場合も、履歴・ファイル項目に含まれているミニダンプを使用して、再分析を実行できます。指定された障害についてミニダンプも関連付けられた MVS ダンプ・データ・セットも存在しない場合は再分析を開始できませんが、ISPF インターフェースからリアルタイム分析レポートを参照することはできます。

リアルタイム分析を実行したときにリスト (またはサイド・ファイル) が使用不可であった場合、再分析のために、そのリスト (またはサイド・ファイル) を使用可能にすることができます。Fault Analyzer は、異常終了に関連するソース・ステートメント情報を提供できるようになりました。

障害再分析は、バッチまたは対話式の 2 とおりの方法で実行できます。どちらの方法も、Fault Analyzer ISPF インターフェースを使用して開始できます。

## バッチ再分析

バッチ再分析レポートのフォーマットは、リアルタイム分析レポートと同じです。バッチ再分析レポートは、再分析ジョブの DD ステートメントに順次ファイルとして書き込まれますが、障害履歴・ファイル項目には保管されません。

ISPF インターフェースを使用してバッチ再分析を開始するだけでなく、独自の JCL を使用してバッチ再分析ジョブを実行することもできます。詳しくは、[Fault Analyzer ISPF インターフェース ページ 62](#)を参照してください。



## 対話式再分析

順次ファイルとして書き込まれるリアルタイム分析レポートやバッチ再分析レポートとは異なり、対話式分析レポートは、目的のセクションを選択できる一連のパネルとして提示されます。

また、対話式再分析を行うことにより、ミニダンプおよび関連付けられた MVS™ ダンプ・データ・セットに含まれるストレージ域の内容を表示できます。ただし、その内容は、必ずしもレポート内でフォーマットされるとは限りません。これは、CICS® システム異常終了分析を実行する唯一の方法でもあります。

対話式再分析は、Fault Analyzer ISPF インターフェースを使用する以外の方法では開始できません。

## 障害ヒストリー・ファイル

障害ヒストリー・ファイルは、Fault Analyzer が分析した障害に関する情報を含む PDS または PDS データ・セットです。ヒストリー・ファイル内に別個のメンバーとして保管された障害項目には、以下のタイプの情報が含まれます。

- リアルタイム分析の主な情報 (例えば、異常終了コードや、障害のあるプログラムの名前)
- 実行環境の詳細 (例えば、ジョブ名、システム ID、および障害が発生した日時)
- 関連したリアルタイム分析レポート (該当する場合)
- 保管されたミニダンプ (該当する場合)
- 関連した MVS ダンプ・データ・セットの名前 (該当する場合)

ヒストリー・ファイル内の新しい各障害項目には、ヒストリー・ファイルで固有の ID が付与されます。ID は 1 から 3 文字の接頭部と、5 桁のシーケンス番号で構成されます。デフォルトの接頭部は「F」ですが、このデフォルトは以下のいずれかの方法で変更できます。

- Fault Analyzer ISPF インターフェースを使用します。アクション・バーのプルダウン・メニュー・オプションで、**[File] > [Change Fault History File Settings]** をクリックします。詳しくは、[「障害ヒストリー・ファイル設定の変更 ページ 94」](#) を参照してください。
- IDIUTIL バッチ・ユーティリティーを使用します。詳しくは、[「ヒストリー・ファイルの管理 \(IDIUTIL ユーティリティー\) ページ 441」](#) を参照してください。

シーケンス番号は 1 で始まり、新しい障害項目が作成されるたびに 1 ずつ増分します。シーケンス番号が 99999 に達すると、1 に戻ります。ある特定の障害 ID が既に存在している場合は、次に使用可能な障害 ID が割り当てられます。

PDSE ヒストリー・ファイルに保持される障害項目の最小数、または PDS ヒストリー・ファイルに入れることができる障害項目の最大数を次のようにして設定できます。

- Fault Analyzer ISPF インターフェースを使用します。アクション・バーのプルダウン・メニュー・オプションで、**[File] > [Change Fault History File Settings]** をクリックします。詳しくは、[「障害ヒストリー・ファイル設定の変更 ページ 94」](#) を参照してください。
- IDIUTIL バッチ・ユーティリティーを使用します。詳しくは、[「ヒストリー・ファイルの管理 \(IDIUTIL ユーティリティー\) ページ 441」](#) を参照してください。

障害の最大限度に達し、すべての項目がロックされた場合は、障害項目の最小数を超えることがあります。(障害項目のロックの詳細については、[障害項目情報の表示 ページ 134](#)を参照してください。)この限度を設定しても、新しく作成された障害項目に割り当てられるシーケンス番号には影響しません。PDSE ヒストリー・ファイルの場合、デフォルトでは最小 25 の障害項目が保持されます。PDS ヒストリー・ファイルの場合、デフォルトでは障害項目数は制限されません。

ヒストリー・ファイルには、PDS データ・セットまたは PDSE データ・セットのいずれも使用できますが、PDS データ・セットでは同時メンバー書き込み機能が使用不可であるのに対して PDSE データ・セットではそれが可能なため、PDSE データ・セットが推奨されます。Fault Analyzer により、PDSE 履歴ファイルの使用時に共有機能とパフォーマンスが向上し、履歴ファイルが IDIS サブシステムで管理できるようにします(「[ヒストリー・ファイル \\$INDEX データのキャッシュ ページ 320](#)」を参照)。

障害項目の詳細の表示、リアルタイム分析レポートのブラウズ、バッチおよび対話式の障害再分析の開始、またはヒストリー・ファイルの対話式表示からの障害項目の削除を実行できます。これらのアクションについては、[Fault Analyzer ISPF インターフェース ページ 62](#)に説明があります。

ビュー・メンバーは、IDIVIEWS DDname で識別されるデータ・セットに作成できます。このデータ・セットには、Fault Analyzer ISPF インターフェースを使用して同時に表示する、任意の数のヒストリー・ファイルの名前が含まれています。詳しくは、[ビューの使用 ページ 66](#)を参照してください。

## 関連付けられたダンプ・データ・セット

ヒストリー・ファイルの障害項目と関連付けられるダンプ・データ・セットは 2 種類あります。

### 密結合

これらはリアルタイム処理中に Fault Analyzer により作成されたダンプ・データ・セットで、障害項目と一意にリンクされます。障害項目の削除時に、関連付けられたダンプ・データ・セットも自動的に削除されます。

密結合ダンプ・データ・セットの例:

- 拡張ミニダンプ・データ・セット (XDUMP)
- IEATDUMP RFR データ・セット (TDUMP)
- SVC ダンプ RFR または Java データ・セット (SDUMP)

### 疎結合

これらは、障害項目と一意にリンクされていないダンプ・データ・セットです。障害項目は既存のダンプ・データ・セットの分析の結果として作成されました。例: SLIP トラップを設定して作成した SVC ダンプが、**Fault Entry List** 画面の **File > Analyze MVS Dump Data Set** を使用して分析用に選択されました。

同じダンプ・データ・セットを複数の障害項目と関連付けることができます。障害項目の削除時に、ダンプ・データ・セットは削除されません。Fault Analyzer の管理対象外と見なされるからです。

## ヒストリー・ファイル・データ・セット内の特殊メンバー

ISPF などのヒストリー・ファイル・データ・セットのメンバーをリストすると、名前が「\$\$」で始まるメンバーが表示されることがあります。これらは、障害項目ではなく、次のように Fault Analyzer で使用される内部制御メンバーです。

## \$\$INDEX

このメンバーには、履歴・ファイル内のすべての障害項目の索引が含まれ、各障害の基本情報に迅速にアクセスするために使用されます。また、履歴・ファイル内の障害に対するすべての重複情報を保持する単独のリポジトリでもあります。

### \$\$INDEX

メンバーが何らかの理由で欠落している場合は、次に履歴・ファイルが更新されるときに再ビルドされます。このような状態は、例えば、リアルタイム分析によって新規障害項目が作成されるとき、または既存障害項目のユーザー情報が更新されるときに発生する場合があります。



**注:** 再ビルドされた \$\$INDEX メンバーには、重複障害の発生に関する情報は含まれません。

サンプル・アセンブラー・プログラムは、データ・セット IDI.SIDISAM1 のメンバー IDIS\$NDX として提供されています。このプログラムは、履歴・ファイルの \$\$INDEX メンバーのユーザー固有のレポートまたは統計分析を作成する際のベースとして使用することができます。

## \$\$BACKUP

このメンバーには、IDIUTIL バッチ・ユーティリティ SetFaultPrefix、SetMaxFaultEntries、または or SetMinFaultEntries 制御ステートメント (詳細は、[履歴・ファイルの管理 \(IDIUTIL ユーティリティ\)](#) ページ 441 を参照) を使用して、または Fault Analyzer ISPF インターフェース (詳細は、[障害履歴・ファイル設定の変更](#) ページ 94 を参照) を使用して \$\$INDEX メンバーに設定された履歴・ファイル固有の設定のコピーが含まれます。何らかの理由で \$\$INDEX メンバー情報が失われた場合は、これらの設定が \$\$BACKUP メンバーからリカバリーされます。

## Fault Analyzer サポートされるアプリケーション環境

Fault Analyzerは、z/OS® において、以下の言語環境およびアプリケーション環境で実行されるアプリケーションをサポートします。

- COBOL
- PL/I
- アセンブラー
- C/C++
- Language Environment
- UNIX® システム・サービス
- CICS®
- IMS™
- DB2®
- MQSeries®
- Java™

ホーム・スペース・モードでの実行か、または 1 次アドレス・スペースがホーム・アドレス・スペースに等しい 1 次スペース・モードでの実行のみをサポートしています。

Fault Analyzer は、z/OS の現在のバージョンでのみ正式にサポートされていますが、以前のバージョンでも正常に動作する可能性があります。z/OS 環境での Fault Analyzer:

- 31 ビット・アドレッシング・モードで動作します。
- 24 ビット、31 ビット、または 64 ビットのアドレッシング・モード・アプリケーションで分析を実行します。
- マルチスレッド、DLL、および XPLink アプリケーションをサポートしています。

アセンブラー、COBOL、および Enterprise PL/I は、データが 64 ビット・ストレージ内にある場合に関連ストレージ域のフォーマットとソース行データ・フィールド値が提供される、唯一のアプリケーション・プログラミング言語です。

C++ のサポートでは、クラス情報は提供されません。

リアルタイム分析は、次の TCB 保護キーで実行されているタスクに制限されています。

- CICS の場合: キー 8 または 9
- CICS 以外の場合: キー 8

その他の保護キーでタスクが実行されている場合、Fault Analyzer は [IDIO123S ページ 722](#) メッセージを発行します。別の保護キーで実行されているタスクで異常終了を分析するには、IDIO123S メッセージで SLIP トラップを設定して SVC ダンプを取り込みます。**Fault Entry List** 画面で **File > Analyze MVS Dump Data Set** を選択すると、SVC ダンプを分析できます。

アプリケーション環境に対する基本サポートに直接は関係ないと考えられる問題を解決するには、最新の Fault Analyzer メンテナンスをインストールしてください。詳細については、[Fault Analyzer の保守 ページ 459](#)を参照してください。

Fault Analyzer は、以下のいずれかのアプリケーション環境をサポートしています。

- 適用される保守がない一般出荷 (GA) リリース
- 指定した PTF (UInnnnn) または APAR (PHnnnnn) が適用されている GA リリース

以下の表は、サポートされている各環境バージョンについて、Fault Analyzer V14.1 に適用される最小メンテナンス・レベルを示しています。

**表 1. サポートされるアプリケーション環境**

サポート対象の環境	サポート対象の環境バージョン	Fault Analyzer V15.1 の最小メンテナンス・レベル
CICS Transaction Server	V6R1 (740)	GA
	V5R6 (730)	GA
	V5R5 (720)	GA
	V5R4 (710) およびこれ以前	GA
Enterprise COBOL	V6R4	GA

表 1. サポートされるアプリケーション環境

(続く)

サポート対象の環境	サポート対象の環境バージョン	Fault Analyzer V15.1 の最小メンテナンス・レベル
	V6R3	GA
	V6R2	GA
	V6R1 以前	GA
Enterprise PL/I	V6R1	GA
	V5R3	GA
	V5R2	GA
	V5R1 およびこれ以前	GA
:NONE.	V13 以前	GA
IMS	V15 以前	GA
IBM Java for z/OS	Version 8 およびこれ以前	GA
z/OS	V3R1 以前	UI94574

## コンパイラー・サポート

最小メンテナンス・レベルが適用されている場合、Fault Analyzer V15.1 は、指定された環境バージョンでコンパイルされたプログラムに対して、その環境の以前のバージョンでコンパイルされた同じプログラムに対して生成されたのと同じ結果を生成します。

Fault Analyzer の最小メンテナンス・レベルは、コンパイラー・バージョンの新機能をすべてサポートしているとは限りません。最新の Fault Analyzer の機能拡張を取得するには、メンテナンス・レベルを最新の状態に保つ必要があります。

## MQSeries のサポート

MQSeries サポートは以下のいずれかで構成されます。

- MQSeries 呼び出しで発生する異常終了

異常終了自体に関する通常の情報に加えて、MQSeries API の説明が提供されます。

- MQSeries に対する前の呼び出しに関する情報

この情報は、ゼロ以外の理由コードが生成される結果となった最後の MQSeries 呼び出しの識別番号 (ソース行の昇順) で構成され、呼び出しで使用された理由コード・データ・フィールドの現行内容が提供されます。理由コードがその説明とともに示されます。

この情報を収集できるようにするには、以下の条件が満たされている必要があります。

- MQSeries 静的リンケージが使用されている。
- MQSeries 呼び出しを発行するアプリケーションが COBOL で書かれている。
- コンパイラー・リストまたはサイド・ファイルが提供されている。

## Java サポート

Java サポートには、以下の制限があります。

- Fault Analyzer は現在サポートされている Java のバージョンで動作します。Fault Analyzer は以前のバージョンの Java では正常に動作する可能性があります、正式にはサポートされていません。
- z/OS 2.3 での Java サポートには z/OS PTF UA96120 が必要です。この PTF が適用されていない場合、ABEND EC6 が理由コード 0B26 C04A で Java 分析中に発生します。

## バインド関連の従属関係

ロード・モジュール内の CSECT をマップするために、Fault Analyzer は IBM® バインド・プログラムをアプリケーション・プログラミング・インターフェース (API) から呼び出します。通常、この呼び出しはリアルタイム分析時に行われませんが、CICS® システム・ダンプの対話式分析時にも行われる場合があります。バインドは PDS または PDSE データ・セットのみに常駐するロード・モジュールをサポートするため、Fault Analyzer は、他のタイプのストレージからロードされた CSECT をロード・モジュール内で識別できません。

## 障害分析のための既存プログラムの設定

Fault Analyzer に障害の分析を出力させるために既存のプログラムを変更する必要はありません。また、プログラムの再コンパイルも必要ありません。ただし、コンパイラー・リストまたはサイド・ファイルを適切なリポジトリに保管すると、Fault Analyzer は、異常終了プログラムのソース・ステートメントを識別できるようになります。(リストまたはサイド・ファイルを保管しないことを選択した場合でも、異常終了後にリストまたはサイド・ファイルを作成できます。これにより、再分析の際に Fault Analyzer はより詳細な情報を抽出できます。)

サイド・ファイルを提供するには、プログラムを再コンパイルする必要があります。これは、特定のコンパイラー・オプションが要求された場合にのみ、適切なサイド・ファイルが作成されるためです。正しいコンパイラー・オプションを使用して生成されたコンパイラー・リストが既にある場合は、再コンパイルしなくてもサイド・ファイルを作成できます。サイド・ファイルの利点としては、リストよりも簡潔であることがあげられます。詳しくは、[コンパイラー・リストまたは Fault Analyzer サイド・ファイルの提供 ページ 377](#)を参照してください。

## 必要な追加の領域サイズ

Fault Analyzer は、異常終了時に、異常終了したプログラムと同じ領域で動作します。したがって、Fault Analyzer が実行し、異常終了状態のプログラム・ストレージを分析するために、アプリケーションが使用していない予備の GETMAIN ストレージが必要です。最初は、実行環境に応じて最大 16 MB のストレージが必要な場合があります。異常終了プログラムのサイズと複雑さが増すにつれて、この追加の領域サイズは大きくなります。

ストレージ要件の詳細については、[推奨されるストレージ ページ 297](#)を参照してください。

Fault Analyzer が障害のリアルタイム分析を行うのに十分なストレージを取得できない状況では、通常、Recovery Fault Recording (リカバリー障害記録) 障害項目が作成されます (リカバリー障害記録 ページ 57 を参照)。ただし、Recovery Fault Recording がまだ有効になっていないと、後続のバッチ再分析または対話式再分析のために SYSMDUMP が取得される可能性があります。

## コンパイラー・リストまたはサイド・ファイルの選択基準

Fault Analyzer は、基本的に、ソース・レベル分析に使用されるコンパイラー・リストまたはサイド・ファイルを選択する際に、以下の 2 つのタイプの検査を実行します。

1. 言語ごとに変わるサイズ検査が行われます。サイズ検査では、ロード・モジュールのサイズと内容をコンパイラー・リストまたはサイド・ファイルと突き合わせるものが試みられます。例えば、COBOL コンパイラーの LIST オプションが使用される場合、サイズ検査には、CSECT 内の最後の 12 個のアセンブラー命令のオフセットと内容の突き合わせが含まれます。また、現行 COBOL コンパイラーの場合、作業用ストレージ・サイズと TGT サイズも検査されます。
2. ロード・モジュールとコンパイラー・リストまたはサイド・ファイルの間で、日時の検査が行われます。ロード・モジュールの日時以降に作成されたコンパイラー・リストに対して設定が行われます。

特定のコンパイラー・リストまたはサイド・ファイルが選択またはリジェクトされた理由についての詳細情報を入手するには、IDITRACE 機能が使用できます。この機能の使用法の詳細は、[IDITRACE 情報 ページ 387](#) を参照してください。

## 言語環境プログラム CEEWUCHA 異常終了の特殊な処理

Language Environment® サンプルのユーザー条件処理ルーチンである CEEWUCHA が用意されていて、これを使用すると、LE のデフォルト動作を変更して VS COBOL II と同様の動作を取得できます。この条件処理ルーチンは、LE USRHDLR(CEEWUCHA) オプションを使用して指定されます。

Fault Analyzer は初期 U4038 LE 異常終了を抑止して、CEEWUCHA 異常終了コードを重要な異常終了コードとしてイベント要約に出力します (以下の 2 つの条件が満たされた場合)。

- Fault Analyzer によって分析されるアプリケーション異常終了に対して LE USRHDLR(CEEWUCHA) オプションが有効になっている。
- LE から当該条件処理ルーチンに制御が渡された。

CEEWUCHA 以外の名前を持つ条件処理ルーチンが LE USRHDLR オプションで指定されると、初期 LE ユーザー異常終了は抑止されません。

## Fault Analyzer で使用される WTO 宛先コードと記述子コード

Fault Analyzer によって発行されるすべてのオペレーター宛メッセージ (WTO) は、宛先コード 11 (プログラマー情報) および記述子コード 7 (タスク関連) を指定します。

## CICS Storage Accounting Area (SAA) オーバーレイ・アシスタンス

Fault Analyzer には、CICS® SAA (Storage Accounting Area) オーバーレイに役立つことを目的に設計された、以下の2つの機能が用意されています。

- 並行 CICS® タスク・ストレージ情報

対話式再分析レポートによって、現在のトランザクションのストレージにおけるオーバーレイを識別するために使用できるだけでなく、オーバーレイに関係していた可能性がある他のトランザクションによって所有されている関連ストレージ域も入った画面を使用できます。

詳しくは、[CICS Transaction Storage Summary ページ 183](#)を参照してください。

- XEIN グローバル・ユーザー出口の実装による、SAA オーバーレイの早期検出

XEIN 出口は、ストレージ・オーバーレイが CICS® によって検出されるかどうか、標準リアルタイム分析を実行するために Fault Analyzer が起動されるか、または後で再分析できる SVC ダンプが取られるかを確認するために使用されます。

詳しくは、[XEIN グローバル・ユーザー出口の実装 ページ 419](#)を参照してください。



## 第2章. リアルタイム分析

リアルタイム分析が行われるのは、アプリケーションが異常終了するときに、Fault Analyzer が提供されている呼び出し出口のいずれかから起動されるか (Fault Analyzer を呼び出すための出口 ページ 299参照)、プログラム SNAP インターフェースへの呼び出しが行われ、さらにジョブが分析から除外されていない場合です。

一般的に、リアルタイム分析では、次の2つが生成されます。

- レポート。デフォルトで JES に書き込まれます (リアルタイム分析レポート ページ 37 を参照してください)。
- ヒストリー・ファイル内の障害項目。障害の再分析を実行する機能が提供されます。

JES に書き込まれたレポートのコピーは、障害項目にも組み込まれるため、ISPF インターフェースで表示できます。レポートを表示している間は、オプションを異なる値に設定してレポートを変更できません。より詳細に (または簡潔に) 表示するには、調整したオプション、または提供のリストやサイド・ファイルを使用して障害を再分析する必要があります。

このステップは、障害分析プロセスの最初のステップです。ほとんどの場合は、この分析で十分と思われるため、障害を再分析する必要はありません。

特定のジョブについて、ジョブを実行する前にいくつかのオプションを調整できます。

異常終了タスクのアドレス・スペースで分析中に参照された仮想記憶域ページはすべて、ストレージ・コンテンツに応じて、ミニダンプとしてヒストリー・ファイルに書き込まれるか、関連する XDUMP に書き込まれます。ミニダンプを宛先とするストレージ・ページの数がある MaxMinidumpPages オプションを超える場合、ミニダンプと XDUMP のどちらにも書き込まれません。

**LOADER の制限:** LOADER (IEWBLDGO) を使用すると、Fault Analyzer は正常に動作しません。これは、リンク・エディット・モジュールのロード実行手法では、リンク・エディット・モジュールがデータ・セットに書き込まれないからです。CSECT の名前、長さ、および開始オフセットを判別するには、ロード・モジュールのデータ・セット・コピーが必要です。

### ダンプの抑止

抑止できるダンプのタイプは次のとおりです。

- SYSABEND、SYSUDUMP、または SYSDUMP (IEAVTABX MVS™ 変更オプション/抑止ダンプ出口 IDIXDCAP を使用して Fault Analyzer が呼び出された場合)。
- CICS® トランザクション・ダンプ。



**注:** CICS® トランザクション・ダンプを抑止するには、Fault Analyzer を XPCABND かまたは XDUREQ CICS® グローバル・ユーザー・出口にインストールする必要があります (CICS 環境のカスタマイズ ページ 400 を参照してください)。

分析の正常終了時のダンプの抑止はデフォルトです。(正常終了とは、Fault Analyzer が異常終了や S レベルのメッセージを発行することなく、通常の分析終了に達したことを指します。)ただし、CICS® で以下のステートメントを使用する場合、

トランザクション・ダンプが抑制されるのは、Fault Analyzer が呼び出される前 (例えば、それ以前の終了) に要求された場合のみです。

```
EXEC CICS DUMP TRANSACTION DUMPCODE(xxxx)
```

これは、発行するアプリケーション・プログラムに SUPPRESSED 応答が戻されるためです。正しく処理されないと、AEXW が異常終了することがあります。トランザクション・ダンプ抑止と CICS® グローバル・ユーザー出口について詳しくは、[Fault Analyzer および CICS グローバル・ユーザー出口 ページ 34](#) を参照してください。

デフォルトのダンプ抑止を指定変更するには、次のいずれかで行います。

- CICS® トランザクション障害の場合は RetainCICSDump(ALL) オプション ([RetainCICSDump ページ 626](#) を参照)。
- 非 CICS トランザクション障害の場合は RetainDump(ALL) オプション ([RetainDump ページ 627](#) を参照)。
- End Processing ユーザー出口 ([終了処理ユーザー出口 ページ 494](#) を参照)。

ダンプ抑止を、分析の抑止、または障害項目作成の抑止と混同しないでください ([リアルタイム除外処理 ページ 54](#) を参照してください)。

## Fault Analyzer および CICS グローバル・ユーザー出口

Fault Analyzer が CICS® XPCABND グローバル・ユーザー出口 (GLUE) にインストールされていて、トランザクション異常終了分析が正常終了した場合 (LE または非 LE アプリケーションの両方)、Fault Analyzer は該当する戻りコードを CICS® に渡してトランザクション・ダンプを抑止します。この引き渡しは、これ以降のダンプ関連 CICS® GLUE (例えば XDUREQ) が CICS® によって呼び出されないことを意味します。

このような環境のために、その後の CICS® ダンプ GLUE が常に呼び出されるようにする必要がある場合、または CICS® トランザクション・ダンプが常に書き込まれるようにする必要がある場合は、次のいずれかを行います。

- Fault Analyzer End Processing ユーザー出口をコーディングして、ENV.SUPPRESS\_DUMP フラグに「N」を設定します。
- RetainCICSDump オプションを ALL に設定します。詳細については、[「RetainCICSDump ページ 626」](#) を参照してください。

## 障害ヒストリー・ファイルの選択

Fault Analyzer によってリアルタイムで障害を分析する場合は、分析の詳細を記録できるヒストリー・ファイルが使用可能でなければなりません。

ヒストリー・ファイルの名前を Fault Analyzer に指定するには、複数の方法があります。以下に、これらの方法をオーバーライド優先度の低い順に (リスト内の各項目が、以前の項目をすべてオーバーライドする) リストします。

1. 製品のデフォルト名 IDI.HIST。
2. parmlib 構成メンバー IDIHIST で指定された DataSets オプションの IDICNFxx サブオプション。この情報には、論理 parmlib 連結、または IDIOPTLM 構成オプション・モジュールの CNFDSN 設定で提供されるインストール済み環境全体の代替 parmlib データ・セット名のいずれかが含まれています。詳しくは、[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#) を参照してください。

3. IDIHIST ユーザー・オプション・モジュールにより識別される構成メンバーで指定された DataSets オプションの IDICNFUM サブオプション。



**注:** ユーザー・オプション・モジュールが使用される場合、このモジュールは、デフォルトの IDICNFxx parmlib 構成メンバーを置き換えます。したがって、ユーザー・オプション・モジュールが指定する構成メンバーに、DataSets オプションの IDIHIST サブオプションが含まれていなかった場合でも、デフォルトの IDIHIST parmlib 構成メンバーにおける IDICNFxx の指定は認識されません。

4. 異常終了ジョブ・ステップの IDIHIST DDname により提供される DataSets オプションの IDIOPTS サブオプション。
5. 異常終了ジョブ・ステップ内で明示的にコード化された IDIHIST DD ステートメント。
6. 分析制御または End Processing ユーザー出口によって ENV データ域の IDIHIST フィールドに提供されるデータ・セット名。

## オプションによるリアルタイム分析の制御

オプションをグローバルに設定することにより、すべてのジョブの出力を制御できるようにします。ただし、1つのジョブについてのみオプションを設定することもできます。この場合、IDIOPTS DDname を使用してユーザー・オプション・ファイルにオプションを設定してください。

使用可能なすべてのオプションと、そのオプションを指定できるさまざまな方法については、[オプション ページ 567](#) を参照してください。

リアルタイム分析で比較的良好に使用されるオプションは、以下のとおりです。

### RetainDump(ALL)

SYSABEND、SYSUDUMP、または SYSMDUMP

を無条件に保存するには、このオプションを指定します。このオプションを指定しない場合、Fault Analyzer が単独で行った分析が適切であると見なしたとき、多くのダンプは抑止されます。このオプションは、ミニダンプをヒストリー・ファイルに書き込む処理に影響を与えません。詳しくは、[RetainDump ページ 627](#) を参照してください。

このオプションのダンプ後処理部分は、MVS™ IEAVTABX 変更オプション/抑止ダンプ出口 IDIXDCAP の使用にのみ適用されます。

### Detail

リアルタイム分析レポートに含める詳細のレベルを調整するときに、このオプションを指定します。(ダンプが作成される場合は、再分析を実行する際にこのオプションを変更できます。)詳細については、[Detail ページ 586](#) を参照してください。

### Exclude

ジョブを分析から除外するときに、このオプションを指定します。詳細については、[Exclude/Include ページ 590](#) を参照してください。

**NoDup**

重複障害がデフォルトで扱われる方法を変更したい場合は、このオプションを指定します。詳細については、[NoDup ページ 611](#) を参照してください。

**CICSDumpTableExclude**

CICS® トランザクション・ダンプ・コード・テーブルを使用して、CICS® トランザクション障害分析を除外したい場合は、このオプションを指定します。詳細については、[CICSDumpTableExclude ページ 574](#) を参照してください。

DataSets オプションを使用して、リストおよびサイド・ファイルをポイントすることもできます。詳細については、[DataSets ページ 576](#) を参照してください。

**JCL DD ステートメント**

Fault Analyzer をバッチまたはリアルタイムで実行する場合には、DD ステートメントは必要ありません。

**JCL DD ステートメントによるリストのポインティング**

必要に応じて、以下の DD ステートメントを JCL に指定できます。これらのステートメントを指定しないと、以下のデータ・セットの識別に、PARMLIB 構成メンバー IDICNFxx、IDIOPTS ユーザー・オプション・ファイル、または分析制御ユーザー出口の中の定義が使用されます。

**IDILC**

C コンパイラー・リストを含む PDS または PDSE データ・セット

**IDILCOB**

COBOL コンパイラー・リスト (OS/VS COBOL 以外) を含む PDS または PDSE データ・セット

**IDILCOBO**

OS/VS COBOL コンパイラー・リストを含む PDS または PDSE データ・セット

**IDISYSDB**

COBOL または Enterprise PL/I SYSDEBUG か、あるいは XL C/C++ MDBG サイド・ファイルを含む PDS または PDSE データ・セット。

**IDILPLI**

PL/I コンパイラー・リスト (Enterprise PL/I 以外) を含む PDS または PDSE データ・セット

**IDILPLIE**

Enterprise PL/I コンパイラー・リストを含む PDS または PDSE データ・セット

**IDIADATA**

アセンブラー・コンパイルの SYSADATA を含む PDS または PDSE データ・セット

**IDILANGX**

すべての言語の LANGX サイド・ファイルを含む PDS または PDSE データ・セット

上記の DD ステートメントでメンバー名を指定しないでください。これらのファイルの使用方法の詳細については、[コンパイル・リスト](#)または [Fault Analyzer サイド・ファイルの提供 ページ 377](#)を参照してください。

## リアルタイム分析レポート

Fault Analyzer が異常終了を分析するか、IDISNAP または同等の `com.ibm.faultanalyzer.Snap.dump Java™` クラスによって呼び出されると、常にリアルタイム分析レポートが作成されます。ただし、`DeferredReport` オプションが使用されている場合 ([DeferredReport ページ 584](#)を参照) またはレポートが抑止されている場合 ([リアルタイム・レポートの抑止 ページ 38](#)を参照) を除きます。レポートは IDIREPRT DDname に書き込まれ、事前の割り振りが存在しない場合は、`SYSOUT=class` に動的に割り振られるため、JES スプール上の通常ジョブ出力の一部として組み込まれます。

使用される `SYSOUT` クラス (`class`) は、デフォルト・ジョブ出力クラス (`SYSOUT=*`) です。異常終了ジョブ・ステップの `SYSUDUMP` DD ステートメントが JES `SYSOUT` クラスを指定している場合は、同じ出力クラスとフォーム名が Fault Analyzer リアルタイム・レポートに使用されます。

リアルタイム分析レポートを他のファイルに出力する場合は、必要に応じて DD カードを調整してください。例:

```
//IDIREPRT DD DISP=(,CATLG),DSN=MY.REPORT.DS,
//          DCB=(RECFM=VB,LRECL=137),SPACE=(CYL,(1,1))
```

また、ユーザー出口を使用して IDIREPRT を別の出力クラスに割り振ることもできます。詳しくは、[リアルタイム・レポートの SYSOUT クラスを制御 ページ 37](#)を参照してください。

IDIREPRT DDname は LRECL=137 でオープンされます。既存のデータ・セット属性はすべて、この論理レコード長と互換性がある必要があります。

CICS® トランザクション異常終了の IDIREPRT 割り振りは、他のタイプの異常終了の場合と同じです。

Fault Analyzer レポートの内容に関する一般情報、およびレポート例については、[Fault Analyzer レポート ページ 275](#)を参照してください。

## Fault Analyzer リアルタイム・レポートの結合

デフォルトでは、すべてのリアルタイム・レポートが個別の JES スプール・ファイルに書き込まれます。このプロセスは一般的に、サブシステムの再始動前に複数のレポートを書き込めるという点で、サブシステム (CICS®, IMS™ メッセージ処理領域、または WLM 管理 DB2® など) にとって有効であると考えられます。

何らかの理由でレポートを 1 つのスプール・ファイルに書き込みたい場合は、IDIREPRT DD ステートメントを次のように、ジョブまたは始動プロシージャに追加します。

```
//IDIREPRT DD SYSOUT=*
```

## リアルタイム・レポートの SYSOUT クラスを制御

IDIREPRT 割り振りがまだ存在していない場合は、Fault Analyzer が IDIREPRT を `SYSOUT=*` に、または `SYSUDUMP` DDname と同じ `SYSOUT` クラスに動的に割り振ります。次の例のように、DD ステートメントをジョブまたは始動プロシージャに追加することによって、別の `SYSOUT` クラスに変更します。

```
//IDIREPRT DD SYSOUT=sysout-class
```

または、分析制御ユーザー出口を使用して、IDIREPRT を必要なクラスに (次の REXX の例のように) 割り振ることもできます。

```
/* REXX */
/*****/
/* Sample Analysis Control user exit to          */
/* allocate IDIREPRT to SYSOUT class F.          */
/*****/
"IDIALLOC DD(IDIREPRT) SYSOUT(F)"
exit 0
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))
Exits(CONTROL(REXX(ABC)))
```

## リアルタイム・レポートの抑止

Fault Analyzer レポートの JES spool への書き込みを抑止するため、以下の DD ステートメントを ジョブまたは始動プロシージャーに追加できます。

```
//IDIREPRT DD DUMMY
```

または、分析制御ユーザー出口を使用して、IDIREPRT を DUMMY に (次の REXX の例のように) 割り振ることもできます。

```
/* REXX */
/*****/
/* Sample Analysis Control user exit to suppress */
/* the analysis report.                          */
/*****/
"IDIALLOC DD(IDIREPRT) DUMMY"
exit 0
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))
Exits(CONTROL(REXX(ABC)))
```

上記の JES スプール・レポートの抑止に関係なく、リアルタイム・レポートをヒストリー・ファイルに書き込んだり、Fault Analyzer ISPF インターフェースを使用してそこから表示したりできます。

CICS® 環境では、代わりに DeferredReport オプション (こちらがデフォルトです) を使用することをお勧めします。詳しくは、[DeferredReport ページ 584](#)を参照してください。

## SYSLOG 要約

リアルタイム分析の過程で、オペレーター・コンソールにメッセージが書き込まれ、障害の原因が 1 行の要約で示されません。

このメッセージの例を以下に示します。

```
IDI0002I There was an unsuccessful REWRITE of file MYFILE01 (file status 44)
        in program COBFERRD at line # 21
```

このメッセージは、Quiet オプションが有効な場合、SYSLOG に書き込まれません。

## プログラム SNAP インターフェース (IDISNAP) の使用

プログラム SNAP インターフェースは、異常終了しないアプリケーションの問題、または何らかの理由により [Fault Analyzer を呼び出すための出口 ページ 299](#) で説明されている通常の異常終了呼び出し出口のいずれかを使用して Fault Analyzer が分析できないアプリケーションの問題をデバッグするユーザーを支援するために提供されます。このインターフェースを使用すると、アプリケーション・プログラム内の任意の場所からの Fault Analyzer の呼び出しにより、現行環境の分析を要求できます。24 ビット・プログラムと 31 ビット・プログラムにおけるプログラム SNAP インターフェース・モジュール名は IDISNAP です。64 ビット・プログラムにおける同モジュール名は IDISNAP6 です。

IDISNAP の呼び出しが使用される例は、SQL ステートメントの実行後に否定 SQLCODE になる DB2® アプリケーションです。

Fault Analyzer の起動方法を除いて、このタイプの分析と、異常終了により行われる他のリアルタイム分析とに違いはありません。

この出口では、現行のジョブに関連した WTO コンソール・メッセージをメインのトレース・テーブルから抽出し、これらのメッセージを分析レポートに含めすることができます (CICS で実行している場合を除く)。

常に最新のバージョンが使用されていることを確認するため、IDISNAP を動的に呼び出すことが推奨されます。

- 呼び出し側プログラムが DLL オプション付きでコンパイルされている場合、IDISNAP の呼び出しは、DYNAMIC ではなく STATIC でなければなりません。
- C で書かれたプログラムの場合、IDISNAP は動的にのみ呼び出すことができます。

## IDISNAP 呼び出し

IDISNAP のエントリーおよび戻りの指定について、以下で説明します。

### エントリーの指定

IDISNAP へのエントリーについて、レジスターの内容は次のとおりである必要があります。

#### レジスター

##### 内容

1

ゼロ、または入力パラメーター・リストのアドレス (以下参照)

13

72 バイトのレジスター保管域のアドレス

14

戻りアドレス

## 15

IDISNAP のエントリー・ポイント・アドレスです。

## 入力パラメーター・リスト

R1 がゼロ以外の場合、レジスター 1 は、アドレスのリストから構成されるパラメーター・リストのアドレスになります (OS リンケージ)。パラメーター・リスト内の各アドレスは 1 つのパラメーターを指します。

表 2. IDISNAP 入力パラメーター

パラメーター	バイト数	説明
パラメーター 1	4	<p><i>t</i><i>nnn</i> のフォーマットに従ってオプション・パラメーターの数とタイプを指定します。</p> <p>各部の意味は以下のとおりです。</p> <p><b>t</b></p> <p>パラメーター 3 のフォーマットを次のように指定します。</p> <p><b>0</b></p> <p>これは、固定長 140 バイトのパラメーター領域を使用するように指示します。</p> <p><b>N</b></p> <p>これは、可変長のヌル終了パラメーター領域を使用するように指示します。</p> <p><b>nnn</b></p> <p>パラメーター 1 に続くパラメーターの数を次のように指定します。</p> <p><b>000</b></p> <p>パラメーター 1 のみを指定します。</p> <p><b>001</b></p> <p>パラメーター 1 と 2 を指定します。</p> <p><b>002</b></p> <p>パラメーター 1、2、および 3 を指定します。</p> <p><b>00V</b></p> <p>パラメーター 1、2、および 3 に加えて、後続のパラメーターに 1</p>



表 2. IDISNAP 入力パラメーター

(続く)

パラメーター	バイト数	説明
		つ以上のストレージ範囲アドレスの対を指定します。
		 <b>注:</b> このパラメーターに 0000 を指定するのは、レジスター 1 をゼロに設定して IDISNAP を呼び出すのと同じです。
パラメーター 2	140	<p>最初の 40 バイトはユーザー・タイトルで、残りは Fault Analyzer が使用するために予約されます。ユーザー・タイトルの未使用部分は、ブランクに設定します。</p> <p>指定されたユーザー・タイトルはリアルタイム・レポートの見出しに追加され、Fault Analyzer ISPF インターフェース Fault Entry List 画面から表示、更新できます。</p> <p>このパラメーターを処理するには、Parameter 1 が 0001 または 0002 である必要があります。</p>
パラメーター 3	パラメーター 1 が 0002 である場合は 140、パラメーター 1 が N002 である場合は可変です。	<p>Fault Analyzer オプション。例:</p> <ul style="list-style-type: none"> <li>• ヒストリー・ファイルの障害項目が作成されないようにするには、DATASETS(IDIHIST(NULLFILE)) オプションを指定します。</li> <li>• 分析レポートに特定のストレージ域を表示するように要求するには、StorageRange オプションを指定します (<a href="#">StorageRange ページ 631</a>を参照してください)。</li> <li>• ENV データ域情報を戻すように要求するには、SNAPDATA オプションを指定します (<a href="#">SNAPDATA オプションの使用 ページ 42</a>を参照してください)。</li> </ul> <p>このパラメーターを処理するには、パラメーター 1 が 0002、000V、N002、または N00V である必要があります。</p>

表 2. IDISNAP 入力パラメーター

(続く)

パラメーター	バイト数	説明
		<ul style="list-style-type: none"> <li>パラメーター 1 が 0002 である場合、このパラメーターが指している領域は 140 バイトである必要があり、未使用部分はブランクに設定されます。</li> <li>パラメーター 1 が N002 である場合、このパラメーターが指している領域は、可変長の文字ストリングです。これは、ヌル文字 (X'00') で区切る必要があります。</li> </ul>

次のパラメーターは、ストレージ範囲の開始アドレスと終了アドレスの対に使用します。指定可能なアドレス範囲の最大数は 160 です。

アドレス範囲パラメーターを使用する方法は、パラメーター 3 で StorageRange オプションを指定する方法の代替手段で、前者が後者をオーバーライドします。

これらの追加パラメーターを使用するには、パラメーター 1 が 000V または N00V でなければなりません。

- パラメーター 1 が 000V である場合、パラメーター 2 が指している領域は 140 バイトである必要があり、未使用部分はブランクに設定されます。
- パラメーター 1 が N00V である場合、パラメーター 2 が指している領域は 1 から 1024 バイトである必要があり、オプション・ストリングの最後の文字の直後にヌル文字 (X'00') が続きます。

パラメーター $n$	4	ストレージ範囲開始アドレス。
パラメーター $n+1$	4	ストレージ範囲終了アドレス。

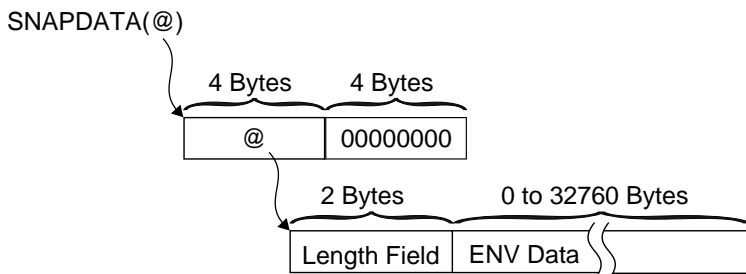
パラメーター 3 で指定されたオプション、またはパラメーター 4 で始まるストレージ範囲アドレスの対によって暗黙的に指定されたオプションは、PARM フィールド・オプションとして、メインの Fault Analyzer 分析モジュール IDIDA に受け渡されます。

### SNAPDATA オプションの使用

IDISNAP は SNAPDATA オプションも処理します ([Snapdata ページ 628](#)を参照してください)。SNAPDATA はポインターのリストを指すストレージのアドレスを 8 バイトの文字形式で指定します。このリストはゼロのアドレス・フィールドで終了します。

図 6: SNAPDATA パラメーター・リスト ページ 43 に、SNAPDATA パラメーター・リストの例を示します。

図 6. SNAPDATA パラメーター・リスト



パラメーター・リストの最初のポインターのみが現在使用されています。2 バイト長の入力フィールドとそれに続くバッファのアドレスである必要があります。そこで IDISNAP はすべての出口が実行した後で ENV データ域の共通出口環境情報の最終バージョンを保存できます。ゼロ長のフィールドでは、Fault Analyzer はパラメーターを無視します。ENV データ域の長さよりも短いフィールドの長さを使用できます。

この機能の使用方法の例で、サンプル・メンバー IDISNAPB (COBOL) および IDISNAPP (PL/1) をレビューします。

## 戻り指定

IDISNAP から戻るときのレジスターの内容は次のとおりです。

### レジスター

#### 内容

0 ~ 1

未定義。

2 ~ 14

未変更。

15

ゼロ。

## 例 1 (COBOL)

以下は、それぞれ異なる呼び出しスタイルで IDISNAP を 5 回呼び出す COBOL プログラムの例です。COBOL ソースの CBL ステートメント内で DYNAM オプションが示すとおり、この例の IDISNAP は動的に呼び出されています。

```
CBL APOST,NOOPT,DYNAM,XREF,LIST,SSRANGE,RENT,MAP
IDENTIFICATION DIVISION.
PROGRAM-ID. COBMST4X
ENVIRONMENT DIVISION.
```

```

INPUT-OUTPUT SECTION.
FILE-CONTROL.
DATA DIVISION.
FILE SECTION.

WORKING-STORAGE SECTION.
01 FILLER          PIC X(20) VALUE 'WORKING-STORAGE'.
01 PARM1           PIC X(4) .
01 PARM2.
  02 PARM2MSG      PIC X(40) VALUE 'HEADING FOR IDIXSNAP'.
  02 PARM2WORK     PIC X(100) .
01 PARM3           PIC X(140) VALUE 'DATASETS(IDIHIST(NULLFILE))'.
01 DATAA         PIC X(200) VALUE 'DATAA'.
01 DATAB          PIC X(200) VALUE 'DATAB'.
01 DATAC          PIC X(200) VALUE 'DATAC'.
01 DATAD          PIC X(200) VALUE 'DATAD'.
01 DATAE         PIC X(200) VALUE 'DATAE'.
01 DATAF         PIC X(200) VALUE 'DATAF'.
01 DATAG          PIC X(200) VALUE 'DATAG'.
PROCEDURE DIVISION.
MAIN SECTION.
START000.
***** 5 CALLS TO IDISNAP
      CALL "IDISNAP".
      MOVE "0000" TO PARM1.
      CALL "IDISNAP" USING PARM1.
      MOVE "0001" TO PARM1.
      CALL "IDISNAP" USING PARM1 PARM2.
      MOVE "0002" TO PARM1.
      CALL "IDISNAP" USING PARM1 PARM2 PARM3.
      MOVE "000V" TO PARM1.
      CALL "IDISNAP" USING PARM1 PARM2 PARM3 PARM1 PARM2WORK
          DATAA DATAB DATAC DATAD.
      GOBACK.
END PROGRAM COBMST4X.

```

## 例 2 (PL/I: 非 DLL)

それぞれ異なる呼び出しスタイルで、IDISNAP を 4 回呼び出す PL/I プログラムの例は、次のとおりです。

```

*PROCESS COMPILE,ATTRIBUTES,AGGREGATE,MAP,LIST,ESD,NEST;
@960IDI:PROC OPTIONS(MAIN) REORDER;
  DCL WKPTR          PTR ;
  DCL WORK           CHAR(4) INIT('0001') ;
  DCL WORK140        CHAR(140) INIT(' ');
  DCL WORK1402       CHAR(140) INIT(' ');
  DCL NUMWK          FIXED DEC(9) INIT(0) ;
  DCL NUMWK2         FIXED DEC(9) INIT(0) ;
  DCL PICWK          PIC'999' INIT(0);
  DCL IDISNAP EXTERNAL ENTRY;
/* ON ERROR CALL PLIDUMP(' F B ') */
/* ON ERROR CALL IDISNAP(WORK,WORK140) */
  FETCH IDISNAP;
  CALL SUBA;
SUBA: PROCEDURE ;
  CALL SUBB;
END SUBA;
SUBB: PROCEDURE ;

```

```

/* THIS WILL CALL IDISNAP 4 TIMES THEN ABEND FOR CALL 5 */
CALL IDISNAP;
DISPLAY ('ZZZ RETURNED FROM IDISNAP TO SUBB');
CALL IDISNAP('0000');
DISPLAY ('ZZZ RETURNED FROM IDISNAP(0000) TO SUBB');
WORK140 = 'USER TITLE DATA.';
CALL IDISNAP(WORK,WORK140);
WORK = '0002';
WORK140 = 'USER TITLE DATA.';
WORK1402 = 'DATASETS(IDIHIST(NULLFILE))';
CALL IDISNAP(WORK,WORK140,WORK1402);
PICWK = NUMWK2      ;
PICWK = NUMWK2      ;
PICWK = NUMWK2/NUMWK ;
END SUBB;
END @960IDI;

```

### 例 3 (PL/I: 31 ビット DLL)

IDISNAP を DLL として呼び出す 31 ビット PL/I プログラムの例:

```

*PROCESS AGGREGATE,ATTRIBUTES(FULL),LIST,MAP,NEST,SOURCE,STMT,
          NONUMBER,OFFSET,XREF(FULL),OPTIONS,NOBLKOFF,RENT;
*PROCESS LIMITS(EXTNAME(8));
IDISPLI1: PROC          OPTIONS(MAIN) ;

DECLARE
  IDISNPD      External Entry Options(asm),
  Work         Char(4),
  Work140      Char(140),
  Work1402     Char(140);

Work = '0002';
Work140 = 'User Title Data';
Work1402 = 'Datasets(IDIHIST(MY.HIST))';
Call IDISNPD(Work,Work140,Work1402);

End IDISPLI1;

```

それ以外のユーザー出口の場合、このフィールドは IDI により初期化され、IDI.SIDISAM1 DDname で指定されるヒストリー・ファイル名になります(この名前は事前割り振りされているか、または DataSets オプションを介して指定します)。

### 例 4 (PL/I: 64 ビット DLL)

IDISNAP を DLL として呼び出す 64 ビット PL/I プログラムの例:

```

*PROCESS AGGREGATE,ATTRIBUTES(FULL),LIST,MAP,NEST,SOURCE,STMT,
          NONUMBER,OFFSET,XREF(FULL),OPTIONS,NOBLKOFF,RENT;
*PROCESS LIMITS(EXTNAME(8)),LP(64);
IDISPLI2: PROC          OPTIONS(MAIN) ;

DECLARE
  IDISNPD      External Entry Options(asm),
  plist_pointer pointer,
  1 snap_list based(plist_pointer),
  3 args_list(3) pointer(32),

```

```

    3 Work      Char(4),
    3 Work140   Char(140),
    3 Work1402  Char(140),
    lastargflag bit(1) based;

plist_pointer = alloc31(stg(snap_list));
Work = '0002';
Work140 = 'User Title Data';
Work1402 = 'Datasets(IDIHIST(MY.HIST))';
args_list(1) = addr(work);
args_list(2) = addr(work140);
args_list(3) = addr(work1402);
addr( args_list(3 ))->lastargflag = '1'b;

Call IDISNPD(args_list);

Call plifree(plist_pointer);
End IDISPLI2;

```

それ以外のユーザー出口の場合、このフィールドは IDI により初期化され、IDI.SIDISAM1 DDname で指定されるヒストリー・ファイル名になります (この名前は事前割り振りされているか、または DataSets オプションを介して指定します)。

## 例 5 (アセンブラー)

IDISNAP を呼び出すアセンブラー・プログラムの例は、次のとおりです。

```

        TITLE 'HLASM EXAMPLE'
R0      EQU    0
R1      EQU    1
R3      EQU    3
R13     EQU    13
R14     EQU    14
R15     EQU    15
ASMSNAP CSECT
ASMSNAP AMODE 31
ASMSNAP RMODE ANY
        PRINT  GEN
        STM   14,12,12(R13)
        LR   R3,R15
        USING ASMSNAP,R3
        LA   R1,REGSAVE
        ST   R13,4(,R1)
        LR   R13,R1
        WTO  'START OF ASMSNAP'
        LOAD EP=IDISNAP
        LTR  R15,R15
        BNZ  ERROR
        LR   R15,R0
        LA   R1,0
        CALL (15)          CALL IDISNAP
        WTO  'END OF ASMSNAP'
        SR   R15,R15      RC=0
        B    RETURN
ERROR   WTO  'ERROR LOADING IDISNAP'
RETURN L   R13,4(,R13)
        L   14,12(,R13)
        LM  R0,12,20(R13)

```

```

BR      R14          RETURN TO CALLER
DROPP   ,
REGSAVE DS  18F
LTOrg   ,
END     ASMSNAP

```

## Java try-catch ブロックからの Fault Analyzer の起動

Java™ プログラムから現在の Java™ の状態を取り込むために、Fault Analyzer を起動して、関連する MVS™ SVC ダンプとともにヒストリー・ファイルの障害項目を作成します。

使用される Fault Analyzer ヒストリー・ファイルはデフォルトのヒストリー・ファイルです。別のヒストリー・ファイルを使用するには、[オプション ページ 567](#)で説明されている `_IDI_OPTS` または `_IDI_OPTSFILE` 環境変数で指定します。作成される障害項目は、続いて Fault Analyzer ISPF インターフェースを使用して再分析され、Fault Analyzer が呼び出されたときに実行中の Java™ コードやネイティブ・コードを検査することができます。

Java™ ダンプ取り込みを容易にするには、以下のことが必要となります。

- Fault Analyzer `IDI_SDUMP_ACCESS` XFACILIT プロファイルに対して ALTER アクセス権限を付与する必要があります。

名前 `IDI_SDUMP_ACCESS` の XFACILIT クラス・プロファイル(リカバリー障害記録 SDUMP アクセスに使用されるプロファイルと同じもの) をセットアップして、Java™ ダンプ取り込みが必要となるユーザー ID またはグループに対して ALTER アクセス権限を付与します。以下のように定義すれば、ユーザーの Java™ アプリケーションで例外が発生した場合に、Fault Analyzer は JDEV グループ内のすべてのユーザーに対して Java™ 取り込み SDUMP を作成できます。

```

RDEF XFACILIT IDI_SDUMP_ACCESS UACC(NONE)
PERMIT IDI_SDUMP_ACCESS CLASS(XFACILIT) ID(JDEV) ACCESS(ALTER)

```

ALTER アクセス権は XFACILIT `IDI_SDUMP_ACCESS` プロファイルに対するものであり、実際の SDUMP データ・セットに対するものではありません。Fault Analyzer は、許可状態を使用して Java™ 取り込み SDUMP へのアクセスを許可します。`IDI_SDUMP_ACCESS` プロファイルはスイッチとして機能し、Fault Analyzer はそのスイッチを検査して、特定のユーザーに対して SDUMP が作成可能かどうかを確認できます。

- MVS™ ポスト・ダンプ出口 `IDIXTSEL` をインストールする必要があります。詳しくは、[MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#)を参照してください。
- IDIS サブシステムを始動する必要があります。詳しくは、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照してください。

Fault Analyzer の呼び出しは try-catch ブロックの中や、ユーザー・プログラムの任意の場所に置くことができ、`com.ibm.faultanalyzer.Snap.dump` メソッドを使用して実行されます。

図 7. Syntax

```

▶▶ com.ibm.faultanalyzer.Snap.Dump — (" — comment — "); ▶▶

```

Fault Analyzer 障害項目のユーザー・タイトル・フィールドを初期化するための、オプションの `comment` 文字ストリングを指定できます。

次の例は、Fault Analyzer が Java™ try-catch ブロックの中からどのように呼び出されるかを示します。

```
public class JavaTest {
    public static void main() {
        ...
        try {
            ...
        }
        catch() {
            ...
            com.ibm.faultanalyzer.Snap.dump("Java error"); // Call Fault Analyzer
        }
    }
}
```

## 代替スナップ・ダンプ呼び出しメソッド

Fault Analyzer を起動するために使用できる Java™ 「Snap.dump」メソッドには、多重定義されたバージョンがいくつかあります。

```
/**
 * Create a Fault Entry
 *
 * @param dumpTitle The user title to use for the Fault Entry. (non-null & non-empty).
 */
public static void dump(String dumpTitle)

/**
 * Create a Fault Entry
 *
 * @param dumpTitle The user title to use for the Fault Entry. (non-null & non-empty).
 *
 * @return The ID of the created Fault Entry, in the form <history file dataset>(<fault id>).
 * E.g. IDI.HIST(F00001)
 */
public static String dump(String dumpTitle, Throwable t)

/**
 * Create a Fault Entry.
 * (This should only be used when gathering diagnostic information requested by IBM Support.)
 *
 * @param dumpTitle The user title to use for the Fault Entry. (non-null & non-empty).
 * @param debug Specifying 'true' will enable verbose debug information.
 * Provide the output to IBM Support.
 *
 * @return The ID of the created Fault Entry, in the form <history file dataset>(<fault id>).
 * E.g. IDI.HIST(F00001)
 */
public static String dump(String dumpTitle, Throwable e, boolean debug)
```

## アプリケーション・クラスパスへのスナップ・クラスの追加

com.ibm.faultanalyzer.Snap.dump メソッドの呼び出しを容易にするには、Fault Analyzer IDIXJAVA Java™ ライブラリーがアプリケーション・プログラム・ビルド・パスに存在しているか、または現行クラス・パス経由で使用可能になっていなければなりません。

IDIXJAVA Java™ ライブラリーは以下の手順で取得できます。



## 1. 以下のいずれかを実行します。

- 次のような JCL を使用して、z/OS 上のアプリケーション・ディレクトリーに jar ファイルをコピーします。

```
/* --- Copy IDIXJAVA to an HFS directory:
//CPYXJAVA EXEC PGM=BPXBATCH
//STDPARM DD *
SH cp "'/IDI.SIDIDOC1(IDIXJAVA)'" /u/hunter2/idixjava.jar
/*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
```

- ご使用のプロジェクト開発ディレクトリーにファイル IDI.SIDIDOC1 として IDIXJAVA(IDIXJAVA) をバイナリー FTP 転送します。



**注:** IDI.SIDIDOC1(IDIXJAVA) データ・セットおよびメンバーは、Fault Analyzer SMP/E インストールの一部として作成されているので、別の高位修飾子が付いている可能性があります。

## 2. 以下のいずれかを実行します。

- プロジェクト・ビルド・パスを構成して、外部 JAR ライブラリー依存として IDIXJAVA を組み込みます。
- ディレクトリー名およびファイル名を現行の ClassPath に追加します。

## Fault Analyzer Java SVC ダンプ・タイトル

Fault Analyzer Java ダンプ・タイトルは、IDISNAP インターフェースまたは BPXBATCH インターフェースのどちらを使用してダンプが書き込まれるかによって異なります。

### • Java IDISNAP インターフェース SVC ダンプ・タイトル。

```
history-file-name(fault-id)^SVCDUMP(0x asid).JAVA IDISNAP^
```



**注:** ^ は非印字文字 x'00' を表します。

次は Java IDISNAP インターフェース SVC ダンプ・タイトルの例です。

```
TESTFA1.TEMP.DA5E1SP.DAJAVAZZ.JC83201.HIST(F00027). .SVCDUMP(0x0138).JAVA IDISNAP.
```

### • Java BPXBATCH インターフェース SVC ダンプ・タイトル。

```
history-file-name(fault-id)^SVCDUMP(0x asid).JAVA IDI0162I^
```



**注:** ^ は非印字文字 x'00' を表します。



次は Java BPXBATCH インターフェース SVC ダンプ・タイトルの例です。

```
TESTFA1.TEMP.DA4E1SP.DAJAVAZZ.JZJ2C02K.HIST(F00700). .SVCDUMP(0x00DB).JAVA IDI0162I.
```

## Java ダンプ・イベントからの Fault Analyzer の起動

Java™ -Xdump 環境スイッチを使用して、例外の発生時に TDUMP を取り、その TDUMP を Fault Analyzer に渡して分析レポートを生成することができます。-Xdump 設定の使用について詳しくは、<https://www.ibm.com/docs/en/sdk-java-technology/8?topic=options-xdump> にアクセスしてください。この解説書では、ダンプを起動できる Java™ 内部イベントおよび適用できるフィルターについて説明しています。

以下のセットアップは、NullPointerException または SocketException に関する Fault Analyzer レポートを取得するための -Xdump の使用例です。

```
java -Xdump:system:events=throw+catch+uncaught,filter=*NullPointerException*,opts=IEATDUMP
-Xdump:system:events=throw+catch+uncaught,filter=*SocketException*,opts=IEATDUMP
-Xdump:tool:events=throw+catch+uncaught,filter=*NullPointerException*,
  exec="tso 'idida dsn(%last) datasets(idihist(da.test.gui.hist))' "
-Xdump:tool:events=throw+catch+uncaught,filter=*SocketException*,
  exec="tso 'idida dsn(%last) datasets(idihist(da.test.gui.hist))' " Java-program-name
```

opts=IEATDUMP を指定した [system] ダンプ・エージェントを使用して、ロー TDUMP を取り込みます。次に、同じイベントに対する [tool] ダンプ・エージェントが exec= オプションを使用して Fault Analyzer を呼び出し、dsn(%last) トークンを介して TDUMP を渡します。

Xdump:tool オプション内の [exec] は、要求されたイベントが起動されたときに呼び出すコマンドを Java™ に提供します。[exec] スtring は、二重引用符で囲む必要があり、その結果 -Xdump スtring が長くなるため、Java™ プログラムの実行に使用される BPXBATCH ジョブの STDPARM DD を使用してシーケンス全体を提供したほうが簡単な場合があります。

range サブオプションを使用すると、特定の例外について生成されるダンプの数を制限できます。上の例では、range 1..4 を指定すると、4 つの NullPointerException スロー・イベントまたは SocketException スロー・イベントについてのみダンプが生成され、それ以上は無視されます。このオプションは、例外がキャッチされて再スローされる場合に特に便利ですが、それ以降の例外は処理されないこととなります。

priority サブオプションを使用すると、特定の順序でダンプを作成することができます。したがって、Java™ ダンプまたはヒープ・ダンプも必要な場合には、これらのダンプ時に Fault Analyzer が呼び出されるのを防止するために、TDUMP 以外のものは、Xdump system と Xdump tool の両方を合わせた優先順位より低い、高いかのいずれかに設定してください。

## Fault Analyzer Java ラッパー・ユーティリティーの使用

Fault Analyzer Java ラッパー・ユーティリティーは、FA Java クラスを使用して、try-catch ブロックの呼び出し可能 Java アプリケーションをラップします。

try-catch ブロックは Java™ アプリケーションからの未処理の例外をトラップし、Fault Analyzer メソッドを使用して com.ibm.faultanalyzer.Snap.dump を呼び出します。

## 使用

IDIXJAVA jar ファイルを HFS または zFS ディレクトリーにバイナリーとしてコピーします。

```
/* --- Copy IDIXJAVA to an HFS or zFS directory:
//CPYXJAVA EXEC PGM=BPXBATCH
//STDPARM DD *
SH cp "'/IDI.SIDIDOC1(IDIXJAVA)'" /u/hunter2/idixjava.jar
/*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
```

## Fault Analyzer ラッパーの起動

ラッパー・ユーティリティーは、コマンド・ラインから、またはバッチ・ジョブ (JCL) の一部として呼び出すことができます。

次のいずれかのコマンドを使用して、コマンド・ライン・アプリケーションからラッパー・ユーティリティーを起動します。

- `java -jar idixjava.jar <clsName> <mainArgs>`

または

- `java -cp idixjava.jar FA <clsName> <mainArgs>`

BPXBATCH を使用して起動される Java™ バッチ・アプリケーションの場合:

```
/* --- Launch a batch Java application with
/* --- Fault Analyzer wrapper
// EXPORT SYMLIST=*
// SET HFSDIR=<target directory>
//CPYXJAVA EXEC PGM=BPXBATCH
//STDENV DD *
JAVA_HOME=/usr/lpp/java800/31bit/J8.0/ &&
PATH=${PATH}:${JAVA_HOME}/bin
CLASSPATH=/u/hunter2/classes
/*
//STDPARM DD *
SH java -jar /u/hunter2/idixjava JavaApp arg1 arg2
/*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
```

Fault Analyzer は指定された引数をユーザー・アプリケーションの main メソッドに渡します。specified Java™ プロパティーおよび環境変数はユーザー・アプリケーションで通常どおり使用できます。

未処理の例外が発生した場合、結果として生じた障害項目には Java™ イベントに関する情報が含まれます。

図 8. Java ラッパー・ユーティリティーの使用法を示すサンプルのイベント・サマリー

```
<H1> E V E N T   S U M M A R Y
```

The following events are presented in chronological order.

Event #	Type	Fail Point	Module Name	Program Name	EP Name	Event Location (*)	Description
1	JavaExc		n/a	n/a	Driver.main	L#4	From file:/u/testfa1/Driver.class

## Fault Analyzer JVMTI エージェントの使用

Fault Analyzer は JVMTI (JVM Tool Interface) エージェントを提供します。このエージェントは Java™ 仮想マシン (JVM) の内部で実行し、実行中の Java™ アプリケーションの状態を検査することができます。

Fault Analyzer エージェントは次のコマンドライン・オプションを使用して起動時に追加されます。

```
java -agentpath:<path-to-agent> <java Class Name>
```

例えば、エージェントを使用して、「HelloWorld」を実行する次のコマンドを指定できます。

```
java -agentpath:IDIJAGNT.so HelloWorld
```

### Java™ インストルメンテーション API を使用して、起動後にエージェントを追加できますか。

いいえ、Fault Analyzer エージェントは JVM の初期化前のみ使用可能な特定の JVMTI 機能を要求します。

### Fault Analyzer エージェントは何を行いますか。

JVM が起動すると、エージェントは JVM

により使用可能になるローカル変数についての情報を要求します。これは、JVM がフルスピード・デバッグ (FSD) モードで実行することを示しています。

すべての例外に対し、エージェントは既知の catch

ブロックがあるかどうかをチェックします。ない場合、エージェントは現在のスレッド・スタック・フレームのローカル変数を含む障害項目を作成します。

### エージェントが抽出するのは何の情報ですか。

エージェントはすべてのアクセス可能なローカル変数を現在のスレッドとともに取得します。収集される具体的な情報は、現在実行しているスタック・フレームの種類によって異なります。

- Java インスタンス・フレーム:
  - 実行中の行およびファイル名が抽出されます。
  - すべてのローカル変数のタイプ、名前、および値が抽出されます。
  - ローカル・インスタンスの、修飾子、タイプ、名前、および値を含む、すべてのフィールドが抽出されます。

- Java 静的フレーム:
  - 実行中の行およびファイル名が抽出されます。
  - すべてのローカル変数のタイプ、名前、および値が抽出されます。
  - メソッドのクラスを宣言する、静的フィールド修飾子、タイプ、名前。および値を含む、静的フィールド。
- ネイティブ・フレーム:
  - 「ローカル・インスタンス」の、修飾子、タイプ、名前、および値を含む、フィールドが抽出されます。
- 静的ネイティブ・フレーム:
  - メソッドのクラスを宣言する、静的フィールド修飾子、タイプ、名前。および値を含む、静的フィールド。

**Java プログラムが JNI を使用して COBOL プログラムを呼び出して、異常終了しました。エージェントはこれを報告しますか。**

いいえ、JVMTI API は Java™ 例外へのアクセスのみを許可します。異常終了が発生すると、JVM はアクティブではなくなり、ローカル変数名に関する情報は使用できなくなります。

**Fault Analyzer は JNI 異常終了に対して何をレポートできますか。**

Fault Analyzer は DTFJ API を使用した事後分析を実行します。

DTFJ API はローカル変数情報のサブセットへのアクセスを許可します。具体的には、JavaStackFrame インターフェースにより戻されるすべてのオブジェクト・リファレンスです。この API は変数名を提供しません。オブジェクト値のみです。

JavaStackFrame および DTFJ API に関する詳細は、「IBM® SDK, Java™ Technology Edition」を参照してください。

## ダンプ登録処理

ユーザー・アドレス・スペースで実行される SYSABEND、SYSMDUMP、および SYSUDUMP 処理とは異なり、MVS™ の SVC ダンプ処理は DUMPSRV アドレス・スペースから実行されます。このため、Fault Analyzer を呼び出す通常の手段の 1 つである MVS™ 変更オプション/抑止ダンプ出口は、SVC ダンプでは使用できません。SVC ダンプについては、Fault Analyzer には MVS™ ポスト・ダンプ出口 IDIXTSEL が用意されています。SVC ダンプはシステム異常終了時に行われるほか、CICS® のシステム・ダンプにも使用されます。

MVS™ ポスト・ダンプ出口 IDIXTSEL をインストールした場合、SVC ダンプが書き込まれるたびに「スケルトン」障害項目が作成されます。MVS™ ポスト・ダンプ出口 IDIXTSEL の詳細については、[MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#)を参照してください。この処理は、分析が実行されない時点で通常のリアルタイム処理とは異なるため、レポートもミニダンプも生成されません。この Fault Analyzer 処理が「ダンプ登録」です。

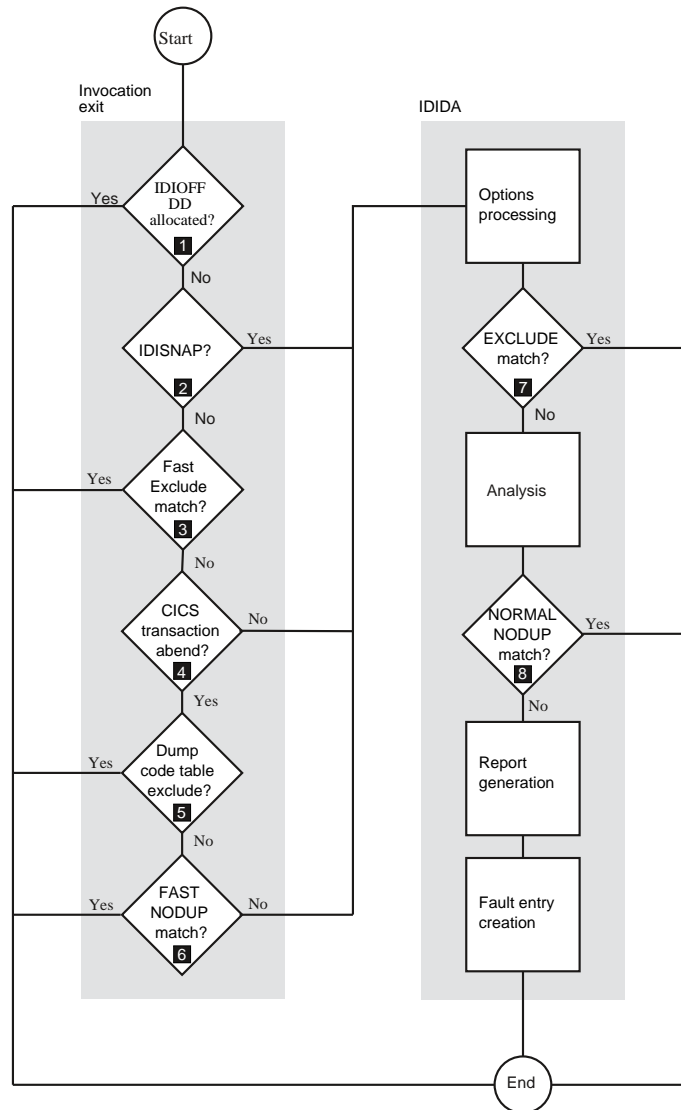
ダンプ登録処理により、通常の分析制御ユーザー出口および通知ユーザー出口と実際には同等である、2 つのユーザー出口が使用可能になります。これらは、DumpRegistrationExits オプションを使用して指定されます ([DumpRegistrationExits ページ 588](#)参照)。

ダンプ登録障害項目には、作成時刻、システム名、SVC ダンプを書き込むジョブの名前など、限られた情報のみ含まれます。異常終了コードおよび異常終了するプログラム名が存在する場合、それらも示されます。ただし、ダンプ登録障害項目の最初の再分析では、障害項目をリフレッシュし、レポートおよびミニダンプとともに保存します。詳しくは、[リフレッシュ処理 ページ 244](#)を参照してください。

## リアルタイム除外処理

基本サブシステム (JES) が使用不可であればリアルタイム分析が実行されることはありませんが、それ以外にも [図 9: リアルタイム除外処理の概要 ページ 54](#) に示すように、Fault Analyzer 処理のさまざまなエレメントを選択的に除外する方法はいくつかあります。

図 9. リアルタイム除外処理の概要





①

異常終了するジョブ・ステップに対して DDname IDIOFF の割り振りを提供することによって、Fault Analyzer

処理は、分析レポートを生成したり、ヒストリー・ファイル障害項目を書き込んだりせずに、即時に終了します。これは、例えば、JCL の中に以下の JCL

ステートメントを追加することによって行うことができます。

```
//IDIOFF DD DUMMY
```

IDIOFF を割り振ることが、Fault Analyzer で特定のジョブ・ステップが実行されるのを防止する最も速い方法であり、オーバーヘッドが最も少ないと認識されている方法です。



**Notice:** z/OS Unix System Services 環境では、環境変

数 `_IDI_OFF` を「Y」に設定する方法は、IDIOFF DDname

スイッチを使用する方法と同じです。詳しくは、「[環境変数 \(Fault Analyzer\) を使用した `\_IDI\_OFF` の停止 ページ 463](#)」を参照してください。

②

アプリケーション内から IDISNAP を呼び出している場合 ([プログラム SNAP インターフェース \(IDISNAP\) の使用 ページ 39](#)を参照)、オプション処理が実行される前には、それ以上の除外は有効ではありません。

③

高速 Exclude オプション処理が有効で、ジョブが適格の場合 ([高速 Exclude オプション処理 ページ 375](#)を参照)、一致する Exclude オプション ([Exclude/Include ページ 590](#)を参照) を使用して、Fault Analyzer 処理を早期に終了することができます。

④

追加の呼び出し出口の除外は、CICS® トランザクションの異常終了でのみ提供されます。

⑤

CICSDumpTableExclude オプションが有効であり ([CICSDumpTableExclude ページ 574](#)を参照)、CICS® トランザクション・ダンプ・コード・テーブルで、障害に関連する CICS® トランザクション異常終了で CICS® ダンプを必要としないとして指定されている場合は、それ以上の処理は行われません。つまり、分析レポートは作成されず、障害項目も書き込まれません。

⑥

次のいずれかが TRUE の場合、現行の障害に対する処理は終了します。

- NODUP(CICSFAST(...)) オプションで分単位のゼロ以外の数値が指定され、重複 CICS® トランザクションの異常終了障害項目の判定に使用される基準と一致する。
- NODUP(IMAGEFAST(...)) オプションで分単位のゼロ以外の数値が指定され、重複 IMS™ トランザクションの異常終了障害項目の判定に使用される基準と一致する。



分析レポートは作成されず、ヒストリー・ファイル障害項目も書き込まれませんが、次の重複しない障害項目が書き込まれるときに、障害項目に対するヒストリー・ファイル・キャッシュの中の重複カウントが引き続き更新されます。

これらの高速重複検出オプションについての詳細は、[NoDup ページ 611](#) を参照してください。

**7**

EXCLUDE オプション ([Exclude/Include ページ 590](#)を参照)

を使用して、オプションがメインライン・コードに読み取られたときに、Fault Analyzer の処理を終了することができます。

**8**

ヒストリー・ファイル障害項目を書き込む前に、NODUP(NORMAL(...))

オプションが検査されます。オプションがゼロ以外の時間数を指定し、障害項目の重複の判別を使用される基準に一致する場合 ([NoDup ページ 611](#)を参照)、現在の障害についての処理は終了します。リアルタイム・レポートが IDIREPRT に書き込まれますが、ヒストリー・ファイル障害項目は作成されません。

NODUP(NORMAL(...)) オプションは、CICS® トランザクション異常終了障害を含む、すべての障害項目に適用されることに注意してください。

Fault Analyzer 処理が IDIOFF DDname スイッチを使用して除外されていない限り、Fault Analyzer 使用を示す SMF タイプ 89 レコードが書き込まれます。

## 重複障害処理

Fault Analyzer には、「高速」および「通常」の2つの異なるタイプの重複障害処理が用意されています。「高速」は事前分析検出を意味し、「通常」は事後分析検出を意味します。

「高速」重複障害タイプはさらに、CICS® 領域レベルのサブタイプと MVS™ イメージ全体を包含するサブタイプの、2つのサブタイプに分割されます。後者は IMS™ にのみ適用されます。

「高速」重複と見なされない障害にも、「通常」重複処理が行われます。

さまざまなタイプの重複処理は、NoDup オプションを使用して制御されます ([NoDup ページ 611](#)を参照)。NoDup オプションの説明に、各タイプが詳細に説明されていますが、簡単に比較できるように、以下に一般的な概要を示します。

**表 3. 重複処理タイプの比較**

処理のアスペクト	ImageFast	CICSfast	通常
オプションを使用して制御	NoDup(ImageFast( IMS NoDup(CICSfast(...)) (...)))	NoDup(CICSfast(...))	NoDup(Normal(...))
適用可能先	IMS™	CICS® トランザクシ ョン障害のみ	すべての障害



表 3. 重複処理タイプの比較 (続く)

処理のアスペクト	ImageFast	CICSfast	通常
	害 (CICS® トランザクション障害を除く)		
処理順	1	1	2
障害分析の抑止によるパフォーマンスの改善 (つまり、IDIDA TCB を付加しない)	あり	あり	なし
ヒストリー・ファイル障害項目の抑止によるディスク・スペースの節約	あり	あり	あり
IDIS サブシステム始動を必要とする	あり	なし	なし
重複したシグニチャー・リポジトリの場所	IDIS サブシステム・ストレージ	CICS® 領域ストレージ	ヒストリー・ファイル索引
デフォルトの設定値	使用可能、5分	使用可能、5分	使用可能、24時間

ある特定のヒストリー・ファイル障害項目に対して発生した重複障害の数は、Fault Entry List 画面の「DUPS」列に表示されます (Fault Entry List 画面 ページ 63 を参照)。

## リカバリー障害記録

Fault Analyzer のリカバリー障害記録機能は、リアルタイム分析時の異常終了の問題によって通常の障害項目の作成が妨げられる条件を減らすために用意されています。この方法は、例えば以下の状況で行われます。

- ストレージ不足。(IDI0005S ページ 696 を参照。)
- Fault Analyzer が異常終了。(IDI0047S ページ 705 を参照。)
- Fault Analyzer がタイムアウト。(IDI0092S ページ 715 を参照。)
- 無効な負のストレージ長の要求。(IDI0105S ページ 718 を参照。)

終了条件がリカバリー障害記録処理に従う場合、スケルトン障害項目が作成され、関連 SDUMP (SVC ダンプ) または IEATDUMP (トランザクション・ダンプ) が書き込まれます。

まず、パフォーマンス上の理由により SDUMP がダンプ・タイプとして優先されるため、リカバリー障害記録ダンプ・タイプとして SDUMP を使用するためのセキュリティー・アクセスが付与されているかどうかのチェックが行われます。(詳しくは、SDUMP RFR データ・セットに対する XFACILIT リソース・クラスの使用 ページ 311 を参照してください。)



**注:** SDUMP の使用時にリカバリー障害記録機能をサポートするには、MVS™ ポスト・ダンプ出口 IDIXTSEL が必要です。この出口の詳細については、SVC ダンプ登録 ページ 303 を参照してください。

SDUMP を使用できない場合は、リカバリー障害記録ダンプ・タイプとして代わりに IEATDUMP が使用されます。

使用されているダンプ・タイプにかかわらず、RFR ダンプという用語は、リカバリー障害記録ダンプ・データ・セットを指します。

RFR ダンプにより追加データ・セットが作成され、このデータ・セットに MVS™ がアドレス・スペースのダンプを書き込みます。このデータ・セットはミニダンプよりもかなり多くの DASD スペースを使用しますが、この状態では、Fault Analyzer はミニダンプを収集できません。したがって、スケルトン障害項目の再分析にはミニダンプの代わりに RFR ダンプ・データ・セットが使用されます。



**注:** リカバリー障害記録処理を有効にするには、IDIS サブシステムが始動済みであり、`UPDINDEX` パラメーターが有効になっている (これがデフォルトです) 必要があります。

障害項目が作成される履歴・ファイルは、異常分析終了時に判別された異常終了ジョブの現行履歴・ファイル、または IDIS サブシステムのデフォルトの履歴・ファイルです。異常終了ジョブに判別された現行履歴・ファイルが PDSE の場合は、これが最初に使用されます。それ以外の場合は、IDIS サブシステムのデフォルトの履歴・ファイルが使用されます。

メッセージ [IDI0126I ページ 724](#) が発行されて、障害項目が作成された先の履歴・ファイルを示します。

RFR ダンプが IEATDUMP の場合、異常終了領域から作成されます。ただし、SDUMP の場合、IDIS サブシステムにより作成されます。スケルトン障害項目は、常に IDIS サブシステムにより作成されます。

リカバリー障害記録プロセスが開始されると、IDIS サブシステムで使用可能なオプションで指定された通知ユーザー出口を除き、このプロセスに対してユーザー出口は起動されません。ユーザー出口は、スケルトン・リカバリー障害記録障害項目の作成時に起動されます。リカバリー障害記録イベントと通知ユーザー出口の他の呼び出しを区別するために、NFY.NFYTYPE フィールドが「R」に設定されます。通知ユーザー出口の詳細については、[通知ユーザー出口 ページ 499](#)を参照してください。

RFR ダンプが IEATDUMP の場合、デフォルトのデータ・セット名パターンが使用されます。`IDIRFRHQ.IDIRFR.&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.`

この名前を変更する (または IEATDUMP RFR ダンプを無効にする) には、「[デフォルト・リカバリー障害記録 IEATDUMP データ・セット名 \(RFRDSN\) の変更 ページ 341](#)」を参照してください。関連付けされた障害項目が削除されるときに IEATDUMP データ・セットの自動削除を許可するには、ご使用のシステム固有のセキュリティー規則に従って、デフォルトの高位修飾子の変更が必要な場合があります。詳しくは、「[リカバリー障害記録データ・セット・アクセスの管理 ページ 310](#)」を参照してください。

リアルタイム分析処理で問題が発生した場所に応じて、リカバリー障害記録障害項目の再分析で再分析レポートを作成できます。これは、リアルタイム分析が正常終了した場合に作成されるレポートと実質的には同じです。通常のリアルタイム障害項目の代わりにリカバリー障害記録障害項目が作成されたという事実は、多くのリカバリー状態では、ユーザーにほとんど認識されません。

リカバリー障害記録障害項目が削除されると、関連 RFR ダンプ・データ・セットも自動的に削除されて、このデータ・セットが不必要にディスク・スペースを使用しないようにします。RFR ダンプ・データ・セットの削除に失敗すると、メッセージ [IDI0187I ページ 737](#) が発行されます。

## RFR ダンプ・タイトル

Fault Analyzer リカバリー障害記録ダンプ・タイトルは、ダンプが IEATDUMP かまたは SVCDUMP かに依存します。

### IEATDUMP ダンプ・タイトル

▶ `history-file-name (fault-id) ^ — dump-data-set-name ^` ▶

 **注:** 上記の ^ は非印字文字 X'00' を表します。

次は Fault Analyzer IEATDUMP リカバリー障害記録ダンプ・タイトルの例です。

```
MY.HIST(F32752). IDIRFRHQ.TDUMP.FAE1.D110216.T003630.S00405.
```

### SVCDUMP ダンプ・タイトル

▶ `history-file-name (fault-id) ^ — ^SVCDUMP(0x asid)^RFR^` ▶

 **注:** 上記の ^ は非印字文字 X'00' を表します。

次は Fault Analyzer SVCDUMP リカバリー障害記録ダンプ・タイトルの例です。

```
MY.HIST(F32753). .SVCDUMP(0x0080).RFR.
```

## Fault Analyzer における SLIP、COMP=0C4 の使用

他の多くの製品と同様に Fault Analyzer は、ESTAE 処理を使用して、通常処理中に発生が予想される S0C4 異常終了から保護し、復旧します。システムで S0C4 異常終了を取り込むように SLIP トラップが設定されている場合は、これらの結果として不必要な一致が発生することがあります。このような不必要な一致を防ぐには、DATA や PVTMOD などの他のパラメーターを使用して SLIP トラップを修飾するか、次のように SLIP トラップをさらに追加します。

```
SLIP SET, ID=xxxx, COMP=0C4, ACTION=IGNORE, location=IDIDA, END
```

*location* は、IDIDA ロード・モジュールが LPA にある場合は LPAMOD です。それ以外の場合は PVTMOD です。

## 拡張ミニダンプ・データ・セット (XDUMP)

XDUMP は、データ・セット名パターンを使用してデフォルトで有効になっています。

```
IDIXDPHQ.XDUMP.&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.
```

このデータ・セット名パターンを変更するには、[「拡張ミニダンプ・データ・セット名パターンの指定 \(XDUMPDSN\) ページ 342」](#) を参照してください。Fault Analyzer は RECFM=FBS LRECL=4096 を使用してデータ・セットを割り振り、[「XDUMP データ・セット・アクセスの管理 ページ 313」](#) で説明されているアクセス許可メカニズムを使用します。XDUMP データ・セットの割り振りが失敗すると、Fault Analyzer により [メッセージ IDI0184W ページ 737](#) が発行されます。

拡張ミニダンプ・データ・セット (XDUMP) は Fault Analyzer 固有のデータ・セットで、特定のタイプのストレージ・ページを保持するために作成されることがあり、障害項目と密接に結合されます。XDUMP データ・セットの使用とその最終処分は、通常、ユーザーに対して透過的です。

XDUMP データ・セットは、以下を含むように設計されています。

- 64 ビット・ストレージ・ページ
- プログラム・リンケージや障害点などの、分析の不可欠な面とは直接関係ないアプリケーション・ストレージ
- 「スピルされた」ミニダンプ・ページ

スピルされたミニダンプ・ページでは、XDUMP データ・セットを使用して 64 ビット・ストレージ・ページを保存し、それらを障害項目と関連付ける必要があります。64 ビット・ストレージ・ページはミニダンプに含まれません。

24 ビットおよび 31 ビット・アプリケーションの場合、XDUMP データ・セットは再分析ストレージ制約を軽減することができます。

リアルタイム処理中に有効な MaxMinidumpPages オプション値を超えると、MaxMinidumpPages 値を超えたページ数が XDUMP に書き込まれ (「スピルされ」) ます。これにより、問題が回避されなければ、メッセージ [IDIO133W ページ 725](#) が発行されます。ただし、XDUMP データ・セットに対する依存関係も作成されます。これは、正しい再分析が行われるために、引き続き存在する必要があります。通常、これは問題ではありません。障害項目が削除されるときに、XDUMP を Fault Analyzer が自動的に削除する必要があるためです。

リアルタイム分析中にページが XDUMP データ・セットにスピルされると、メッセージ [IDIO194W ページ 739](#) が発行されます。

再分析中に、スピルされたページを含む XDUMP データ・セットが使用できない場合、警告が出されます。バッチ再分析の場合、メッセージ [IDIO195W ページ 740](#) も発行されます。

XDUMP はデフォルトで有効になっています。XDUMP が正しく構成されていることをテストするには、[「XDMP の使用の検証 ページ 427」](#) を参照してください。

## 異常終了トラップは障害分析を回避する可能性があります

プログラム内で異常終了トラップ・メソッド (CICS の ESTAE/ESPIE ルーチンや EXEC CICS HANDLE ルーチンなど) を使用すると、異常終了の発生後に Fault Analyzer が制御を受け取ることができなくなる可能性があります。その結果、Fault Analyzer は異常終了を分析できません。

## 異常終了ジョブ JCL のキャプチャーおよび表示

リアルタイム処理中に Jclcapture オプションが有効な場合、Fault Analyzer は JCL を取得し、それを障害項目に保存しようとします。この機能は、JES2 でのみ使用できます。

[JclCapture ページ 607](#) を参照してください。

保存された JCL を取得するには、次のようにします。

- 対話式再分析レポート内から JCL コマンドを発行する。[JCL ページ 111](#) を参照してください。
- 対話式再分析レポートの「Abend Job Information」セクションから「Abend Job JCL」ポイント・アンド・シュート・フィールドを選択します。
- REXX フォーマット・ユーザー出口内から IDIGET コマンドを使用して、保存された \_JCL ステムを取得します。例:

```
/* REXX */
"IDIGET _JCL"
if rc = 0 then do
  "IDIWRITE '<L>Abend job JCL:'"
  "IDIWRITE '<L>'"
  do i = 1 to _jcl.0
    "IDIWRITE '<L>"_jcl.i"'"
  end
end
else do
  "IDIWRITE '<L>JCL not available'"
end
exit 0
```

## 第3章. Fault Analyzer ISPF インターフェース

異常終了後はいつでも、TSO ユーザーとして、Fault Analyzer ISPF インターフェースを開始して、障害を検討できます。このインターフェースを使用すると、以下を実行できます。

- 保管されたリアルタイム分析レポートの表示。
- バッチ再分析の開始 (詳細については、[バッチ再分析の実行 ページ 154](#) を参照してください)。
- 対話式再分析の開始 (詳細については、[対話式再分析の実行 ページ 162](#) を参照してください)。
- 障害に関する情報の表示。
- 発生した障害のうち、現行障害と重複していると見なされたものの詳細表示。
- 障害項目の削除。

ISPF インターフェースにより、以下も実行できます。

- CICS® システム異常終了ダンプの分析 (詳細については、[CICS システム異常終了ダンプ分析の実行 ページ 251](#) を参照してください)。
- Java™ ダンプの分析 (詳細については、[Java 分析の実行 ページ 262](#) を参照してください)。



**注:** 本章の情報は、Fault Analyzer ISPF インターフェースが ISPF から起動されることを想定して書かれていますが、このインターフェースは CICS® から起動することもできます。その場合は、いくつかの制約事項が適用されることがあります。これらの制約事項は、[以下の環境での対話式再分析の実行 CICS ページ 286](#) で説明します。



**再分析:** 障害項目の再分析は、ミニダンプが含まれている場合、あるいは MVS ダンプ・データ・セットに関連付けられている場合にのみ、実行できます。

リアルタイム分析を実行したときに割り振られたか、または DataSets オプションを介して指定されたコンパイラー・リストまたはサイド・ファイルのデータ・セットは、再分析を実行する場合、再使用されます (これらのデータ・セットが、再分析環境で使用可能な場合)。

再分析を、最初のリアルタイム分析と異なるものにするには、以下のいずれかを行います。

- 異常終了に関与するプログラムのコンパイラー・リスト (またはサイド・ファイル) を提供する (最初のリアルタイム分析でこれらが使用可能でなかった場合)。
- 分析オプションを変更する。
- 対話式再分析をして、ダンプ・ストレージを検討する。

バッチ再分析と対話式再分析のステップの主な違いは、以下のとおりです。

- 対話式再分析は、常にすべての詳細を提供します。これを使用すると、分析レポートに含まれない可能性がある保管場所を確認できます。一方、バッチ再分析は Detail オプションで指定したレベルの詳細を提供します。これを使用しても、保管場所を確認することはできません。
- 対話式再分析では、ISPF セッションを占有しますが、バッチ再分析ジョブでは、一度実行依頼すれば、他のジョブを継続できます。

リストまたはサイド・ファイルを提供して、Fault Analyzer が障害再分析の実行時にソース行情報を提示できるようにするには、プログラムをコンパイルし、そのコンパイラー・リストまたはサイド・ファイルを保管する必要があります。このプロセスの詳細については、[コンパイラー・リストまたは Fault Analyzer サイド・ファイルの提供 ページ 377](#)を参照してください。

リストまたはサイド・ファイルが作成済みであり、これを標準外の保管場所に保持している場合は、JCL DD ステートメントを使用してその場所をポイントできます。[JCL DD ステートメントによるリストのポインティング ページ 36](#) 可能な値が示されています。

## インターフェースの起動

Fault Analyzer ISPF インターフェースの起動方法は、そのインターフェースがサイトでどのようにカスタマイズされたかによって異なりますが、一般的に、ISPF 選択パネルからオプションを選択するか、コマンドを発行することによって起動されます。

推奨される Fault Analyzer のさまざまな起動方法については、[ISPF 環境の変更 ページ 328](#)に説明があります。

サイトでの Fault Analyzer ISPF インターフェースの起動方法が分からない場合は、システム・プログラマーまたは Fault Analyzer をカスタマイズしたユーザーにお尋ねください。

インターフェースからオンライン・ヘルプを表示するには、「ヘルプ」ファンクション・キー (PF1) を押します。

## ISPF 分割画面のサポート

1 人の TSO/ISPF ユーザーが、例えば ISPF 分割画面を使用して、複数の Fault Analyzer を同時に起動することは、サポートされてはいますが、多少の制限があります。例:

- 複数の ISPF 分割画面セッションで同一の障害項目の対話式再分析を並行して実行する場合:
  - 単一のセッションのユーザー・ノートに対する変更は、別のセッションには反映されません。
  - 最後に終了した分析からのユーザー・ノートだけが保存されます。
- ユーザー・ノートのリカバリーが有効になっている場合、同じ TSO/ISPF ユーザーは、2 つの別々の SPF セッションで設定された同じ障害項目または同じ MVS ダンプ・データ・セットの対話式再分析を同時に実行できません。
- 「障害エントリ・リスト」画面の「ファイル」プルダウン・メニューから選択できる、最後に使用されたヒストリー・ファイルおよび障害項目リストは、各 ISPF セッションで別々に維持されます。この情報に変更が加えられるときに終了した最後の Fault Analyzer ISPF インターフェース・セッションによって、ユーザーの ISPF プロファイルの情報が更新されます。

## Fault Entry List 画面

Fault Analyzer ISPF インターフェースが始動すると、Fault Entry List 画面が表示されます。[図 10 : Fault Entry List 画面の例 ページ 64](#) Fault Entry List 画面の例を示しています。

図 10. Fault Entry List 画面の例

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'IBMUSER.DEMO.HIST'

{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry), P (Package), L (Lock), U (Unlock), A (Display source), J (Edit JCL).}

  Fault ID Job/Tran User ID Sys/Job Abend Date      Time
  ---      -
  F00323 IDIVPCOB IBMUSER MVS2   S0C7 2019/12/21 13:02:25
  F00445 ALLANT01 JACKIED MVS8   S0C7 2019/12/19 03:29:57
  F00444 ALLANT01 JACKIED MVS8   S0C7 2019/11/28 20:25:30
  F00442 ALLANT01 ALLANT  MVS8   S0C7 2019/09/10 22:20:10
  F00349 CS05     CICSUSER CSCB0050 ASRA 2019/08/23 07:47:23
  F00348 CS04     CICSUSER CSCB0040 ASRA 2019/08/23 07:46:36
  F00345 CS01     CICSUSER CSCB0010 AEIL 2019/08/23 07:43:35
  F00050 PSTRANDR PSTRAND  STPLEX4B S0C4 2019/08/02 17:03:18
  F00035 CICS53    n/a      MVS2    n/a    2019/04/05 14:49:11
F1=Help      F3=Exit      F4=MatchCSR  F5=RptFind   F6=Actions   F7=Up
F8=Down      F10=Left     F11=Right   F12=MatchALL

```



**注:** Fault Entry List 画面に PF キーが表示されない場合、これを表示するには、次の ISPF コマンドを入力してください。

```
FKA ON
```

黄色 (デフォルトの色) で示されたフィールドが、ポイント・アンド・シュート可能なフィールドです。この有効化は、このようなフィールドにカーソルを置いて Enter キーを押すと追加情報を表示できることを示しています。例えば、「Abend」列または「I\_Abend」列の異常終了コードを選択すれば、その異常終了コードに対する説明が表示されます。

Fault Analyzer ISPF インターフェースの使用時に最後に選択された履歴・ファイルまたはビュー ([ビューの使用 ページ 66](#)を参照) が、デフォルトで表示されます。初めてインターフェースを使用すると、有効な DataSets オプションのサブオプションを使用して最初の履歴・ファイル名が取得されます。別の履歴・ファイルまたはビューを選択する場合は、[表示される履歴・ファイルまたはビューの変更 ページ 66](#)を参照してください。

最後に ISPF インターフェースが使用されたときにビューが選択されており、ビューにエラーが含まれていた場合、Fault Entry List 画面が表示される前にエラー画面が表示される可能性があります。[図 11: エラー画面の例 ページ 65](#)に、エラー画面の例が示されています。



図 11. エラー画面の例

```

----- Error -----
Command ==> _____ Line 1 Col 1 76
The following problems were found while processing the view in Scroll ==> CSR
DA.VIEWS(SWBAD1):

* -HistCols syntax error: Missing starting parenthesis. The -HistCols
  specification has been ignored.

* Data set 'xyz' open error: EDC5049I The specified file name could not be
  located.

* -Match syntax error: The subcommand entered for the "FRED" command was
  invalid. The -Match specification has been ignored.

Press PF3 to continue.

*** Bottom of data.

F1=Help      F3=Exit      F7=Up        F8=Down      F12=Cancel

```

エラー画面を終了するには、PF3 キーを押してください。

誤ったビュー・メンバーが読み取られるたびにエラー画面が表示されるため、例えば、Fault Entry List Column Configuration 画面が表示されているときにも、エラー画面が表示される可能性があります。この画面が表示されないようにするには、ビューに示されたエラーを訂正する必要があります。

デフォルトでは、Fault Entry List 画面の項目は、逆の日時順にリストされ、(異常終了の日時に基づく) 最新の障害項目が一番上に表示されます。

リスト内の各障害項目は、1 行に表示され、画面の左側にある障害 ID により識別されます。障害項目ごとに表示されるその他の情報は、ユーザーが決定できます。この詳細については、[Fault Entry List の列構成 ページ 71](#)を参照してください。HistCols オプションを指定せず、ユーザーがカスタマイズを行わない場合に示されるデフォルト情報が、[図 10: Fault Entry List 画面の例 ページ 64](#)に示されています。

表示されるフィールドを使用して、必要な障害を識別できます。または表示の内容を削減し、障害の一部のみを表示できます。この詳細については、[障害項目のソートと突き合わせ ページ 78](#)を参照してください。

ヘルプ・テキストが使用可能になっている場合に画面の一番上に表示されているように ([図 10: Fault Entry List 画面の例 ページ 64](#)を参照)、個々のヒストリー・ファイル項目に対して複数の行コマンドが使用可能です。詳細については、[特定の障害へのアクションの適用 ページ 87](#)を参照してください。ヘルプ・テキストの表示または非表示の方法については、[ヘルプ・テキストの追加または除去 ページ 127](#)を参照してください。

この画面は、標準の UP、DOWN、LEFT、および RIGHT コマンドに応答します。これらのコマンドは、デフォルトでそれぞれ PF7、PF8、PF10、および PF11 ファンクション・キーに割り当てられます。これらのキーを使用すると、必要に応じて横方向または縦方向に画面をスクロールして、入手可能な情報をすべて表示できます。

使用可能な行コマンドをリストするオプションのヘルプ・テキストは、画面の最上位の行が表示される場合にのみ表示されます。表示が任意の行数分スクロールダウンされると、ヘルプ・テキストは消去されますが、表示が先頭にスクロールされ

ると再表示されます。ヘルプ・テキストの一般情報については、[ヘルプ・テキストの追加または除去 ページ 127](#) を参照してください。

履歴・ファイルまたはビュー入力フィールド、および列見出しは、ビューからスクロールアウトされることはありません。しかし、水平方向にスクロールすると、列見出しはその下のデータとともにスクロールされます。

画面の左側の行コマンド入力フィールドは、水平スクロールを行ってもその位置のままです。

画面の右上隅には、現在一番上に表示されている行番号、および現在表示されている左端と右端の列が示されています。

Fault Entry List の末尾には、次の行が表示されます。

```
*** Bottom of data.
```

この行は、スクロール可能な Fault Analyzer ISPF インターフェースのすべての画面の一番下であることを示すのに使用されます。

Fault Analyzer ISPF インターフェースを終了するには、Fault Entry List 画面から Exit コマンド (PF3) を実行するか、Fault Entry List 画面の File メニューから Exit Fault Analyzer オプションを選択します。

## ビューの使用

複数の履歴・ファイルから履歴・ファイル項目を同時に表示すると役立つ場合は、以下を行うことができます。

- [Fault Entry List] 画面の履歴・ファイル名の代わりにビュー名を指定できます。
- [File] メニューの [List Views] オプションを使用してビュー名を選択できます。このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください。

これらのビューの定義は、IDIVIEWS parmlib 構成メンバーで DataSets オプションの IDICNFxx サブオプションにより識別される PDS または PDSE データ・セットで設定する必要があります。

ビューを使用すると、以下を行うことができます。

- 複数の履歴・ファイルから同時に障害項目を表示できます。
- [Fault Entry List] 画面に対して、特定の列のレイアウトを指定できます ([デフォルトの列レイアウトの指定 ページ 351](#)を参照)。
- 障害項目の初期表示リスト用の選択基準を指定できます ([最初の障害項目選択基準の指定 ページ 351](#)を参照)。

ビューのセットアップ方法については、[ビューのセットアップ ページ 350](#)を参照してください。

## 表示される履歴・ファイルまたはビューの変更

Fault Analyzer ISPF インターフェースが最初に開始されるときに、前回表示された履歴・ファイルまたはビューが表示されます。別の履歴・ファイルまたはビューを選択するには、次の方法のいずれかを実行してください。

## 別の履歴・ファイルまたはビューの名前を入力する

表示しようとする履歴・ファイルまたはビューの名前を指定するには、[図 12:別の履歴・ファイルまたはビューの名前の入力 ページ 67](#) の ❶ のように、「*Fault History File or View*」行にその名前を入力します。ここでは、履歴・ファイル名 MY.HIST が選択されています。履歴・ファイルまたはビューの名前を入力した後、Enter キーを押して、障害項目を表示します。

図 12. 別の履歴・ファイルまたはビューの名前の入力

```

File  Options  View  Services  Help
-----
IBM Fault Analyzer - Fault Entry List                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'my.hist' ❶

{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).}

   Fault ID Job/Tran User ID Sys/Job Abend Date      Time
---  F00323 IDIVPCOB IBMUSER  MVS2   S0C7 2019/12/21 13:02:25
---  F00445 ALLANT01 JACKIED  MVS8   S0C7 2019/12/19 03:29:57
---  F00444 ALLANT01 JACKIED  MVS8   S0C7 2019/11/28 20:25:30
---  F00442 ALLANT01 ALLANT   MVS8   S0C7 2019/09/10 22:20:10
---  F00349 CS05     CICSUSER CSCB0050 ASRA 2019/08/23 07:47:23
---  F00348 CS04     CICSUSER CSCB0040 ASRA 2019/08/23 07:46:36
---  F00345 CS01     CICSUSER CSCB0010 AEIL 2019/08/23 07:43:35
---  F00050 PSTRANDR PSTRAND  STPLEX4B S0C4 2019/08/02 17:03:18
---  F00035 CICS53   n/a      MVS2   n/a    2019/04/05 14:49:11
F1=Help      F3=Exit      F4=MatchCSR F5=RptFind   F6=Actions   F7=Up
F8=Down      F10=Left     F11=Right   F12=MatchALL

```

履歴・ファイルまたはビューの命名規則は次のとおりです。

- 履歴・ファイル名の場合、標準の TSO  
命名規則が適用されます。つまり、入力された名前が単一引用符で囲まれていない場合、この名前には自動的に TSO 接頭部が付きます。例えば、TSO 接頭部が FRED  
に設定されているときに、履歴・ファイル名 FRED.HIST を指定する場合は、HIST または  
'FRED.HIST' のどちらかを「*Fault History File or View*」行に入力します。  
終わりの引用符が欠落している場合、自動的に追加されます。
- ビュー名は、IDIVIEWS DDname  
に関連したいずれかのデータ・セット内のメンバー名です。これらの名前を指定するには、小括弧で  
囲みます。例えば、ビュー・メンバー ABC が表示されるように指定するには、「*Fault History File or  
View*」行で (ABC) と入力します。  
右小括弧が欠落している場合、自動的に追加されます。

選択できる履歴・ファイルのリストを入手する場合、ワイルドカードを使用して履歴・ファイル・パターンを指定することもできます。ワイルドカードは、1 つ以上の % 記号またはアスタリスク (\*) ある  
いはその両方で構成されます。

\*

単一のアスタリスクを単独で指定した場合、その位置には少なくとも1つの修飾子が必要であることを示します。1つの修飾子内にある単一アスタリスクは、その位置には0個以上の文字が入ることを示します。

\*\*

二重アスタリスクは、その位置には0個以上の修飾子が入ることを示します。1つの修飾子内にある二重アスタリスクは無効です。

%

単一の%記号は、その位置には任意の英数字または国別文字が1文字入ることを示します。

%%...

各修飾子の中に1個から8個の%記号を指定できます。

以下の例は、有効な履歴・ファイル・パターンを示しています。

履歴・ファイル・パターン	結果のリスト
'FRED.*'	FRED が最初の修飾子であり、少なくとも修飾子がもう1つある、すべての履歴・ファイル名。
'FRED.**'	FRED が最初の修飾子である、すべての履歴・ファイル名。
'FRED.**.HIST'	FRED が最初の修飾子であり、HIST が最後の修飾子であり、その間に0個以上の修飾子が含まれる、すべての履歴・ファイル名。
'AAA%*.B*%%*B'	AAA で始まり、少なくとも高位修飾子内にもう1文字あり、2番目の修飾子は始めと終わりがBであり、それらのBの間に少なくとも3文字が含まれる、すべての履歴・ファイル名。

引用符を使用して履歴・ファイル名を囲むことに関する規則は、履歴・ファイル・パターンにも適用されます。

履歴・ファイル・パターンの最初の修飾子は、プレフィックス変換後に(引用符で囲まない指定により適用された場合)、ワイルドカードのみ(例えば、'\*'、'\*\*'、および '\*\*.HIST')で構成されてはなりません。ただし、PREFIX ON で実行している場合は、\*\*.HIST は有効です。

### 以前に使用した履歴・ファイルまたはビューを選択する

最後に表示された 10

個の履歴・ファイルまたはビューのレコードが保持されます。以前に表示された履歴・ファイルまたはビューを選択するには、まず File メニューの Last Accessed Fault History Files or Views オプションを選択します(このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。このオプションを選択すると、[図 13 : Last Accessed Fault History Files or Views 画面の例 ページ 69](#)の例のように、Last Accessed Fault History Files or Views 画面が開きます。

図 13. Last Accessed Fault History Files or Views 画面の例

```

File Options View Services Help
Last Accessed Fault History Files or Views
Enter the number corresponding to one of the following
previously accessed fault history files or views and press
Enter:
-- 1. 'XMPUSER.DEMO.HIST'
   2. 'XMPUSER.HIST'
   3. 'DA.DCAT'
   4. (APC) Sample view of APC history files
   5.
   6.
   7.
   8.
   9.
  10.
                                HLD ①
F1=Help      F3=Exit      F5=Hold/Rel  F12=Cancel
F05495      883          S62041A S62041  SB34      U3500
F03678      557          S62041A S62041  SB34      U3500
F1=Help      F3=Exit      F4=MatchCSR F5=RptFind  F6=Actions  F7=Up
F8=Down     F10=Left     F11=Right   F12=Retrieve

```

図 13 : Last Accessed Fault History Files or Views 画面の例 ページ 69

に表示されている Last Accessed Fault History Files or Views

画面で、先頭のカーソル位置に目的の履歴・ファイルまたはビューの名前に対応する番号を入力し、Enter キーを押して、選択された履歴・ファイルまたはビューの項目を表示します。

変更を加えずに Fault Entry List 画面に戻るには、PF3 または PF12 を押します。

履歴・ファイル名がリストから抜け落ちるのを防ぐには、入力フィールドにリスト項目の番号を入力して PF5 を押します。項目の右側に「HLD」インディケータが表示されます。上記の例の ① を参照してください。

履歴・ファイル名が既に保持されている (項目の右側に「HLD」が表示されている) 場合は、入力フィールドにリスト項目の番号を入力して PF5 を再び押すことで、その履歴・ファイル名を解放できます。

### 以前に使用した履歴・ファイル項目を選択する

最後に使用された 10

個の履歴・ファイル項目のレコードが保持されます。以前に表示された履歴・ファイル項目を選択するには、まず File メニューの Last Accessed Fault History File Entries オプションを選択します (このオプションの一般的な選択方法については、アクション・バーのプルダウン・メニュー ページ 100 を参照してください)。このオプションを選択すると、図 14 : Last Accessed Fault History File Entries 画面の例 ページ 70 の例のように、Last Accessed Fault History File Entries 画面が開きます。

図 14. Last Accessed Fault History File Entries 画面の例

```

File Options View Services Help
S
C Last Accessed Fault History File Entries
Enter the number corresponding to one of the following
previously accessed history file entries and press Enter:
F
-- 1. 'XMPUSER.DEMO.HIST(F00323)'
   2. 'XMPUSER.DEMO.HIST(F00345)'
   3. 'XMPUSER.DEMO.HIST(F00348)'
   4. 'XMPUSER.HIST(F00331)'
   5.
   6.
   7.
   8.
   9.
  10.
F1=Help    F3=Save    F4=Reset    F7=Up      F8=Down    F10=Left
F11=Right  F12=Cancel

F00345 CS01    CICSUSER CSCB0010 AEIL  2019/08/23 07:43:35
F00050 PSTRANDR PSTRAND  STPLEX4B S0C4  2019/08/02 17:03:18
F00035 CICS53    n/a      MVS2     n/a    2019/04/05 14:49:11
F00034 CICS53    n/a      MVS2     S08E  2019/03/22 13:12:23
F1=Help    F3=Exit    F4=MatchCSR F5=RptFind F6=Actions F7=Up
F8=Down    F10=Left   F11=Right   F12=MatchALL

```

図 14 : Last Accessed Fault History File Entries 画面の例 ページ 70 に表示されている

Last Accessed Fault History File Entries 画面から、先頭のカーソル位置に目的の

ヒストリー・ファイル項目に対応する番号を入力し、Enter キーを押して、その項目を表示します。

変更を加えずに Fault Entry List 画面に戻るには、PF3 または PF12 を押します。

### ビューのリストからビューを選択する

選択可能なビューのリストを表示するには、まず File メニューの List Views オプションを選択します

(このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。これで、[図 15 : View List 画面の例 ページ 70](#) に示されている画面が開きます。

図 15. View List 画面の例

```

File Options View Services Help
s
c View List
Command ==> _____ Scroll ==> CSR
F
Line commands: S (select) B (browse).
{
r
- Name      Description
- APC      Sample view of APC history files
- BATCH    Batch History Files
- DB2      SAMPLE VIEW OF DB2 HISTORY FILES
- JOHNS    View of CICS and IMS History files
- SUBS     SubSystem History Files
- SW       Test
- TOM      Tom's Views
- VIEW2    TEST VIEW
- ZZZZZZZ (No description available)

*** Bottom of data.

F1=Help    F3=Exit    F7=Up      F8=Down    F12=Cancel
F8=Down    F10=Left   F11=Right

```

この画面から、各ビューに対して次の行コマンドのいずれかを入力できます。

**B**

このコマンドを実行すると、ビュー・メンバーをブラウズできます。例えば、ビュー・メンバー JOHNS がブラウズ用に選択された場合、[図 16 : File Browse 画面の例](#) ページ 71 のように、このメンバーの内容が表示されます。

File Browse 画面から View List に戻るには、PF3 または PF12 のどちらかを押します。

図 16. File Browse 画面の例

```

S
I
C
F
{
r
File Options View Services Help
      View List
      File Browse
C 'DA.VIEWS(JOHNS)' Line 1 Col 1 76
L Command ==> ----- Scroll ==> CSR
* View of CICS and IMS History files
CTEST.DUMPA.DACICS.DCAT
CTEST.DUMPA.DAIMS.DCAT
B *** Bottom of data.
*
F1=Help F3=Exit F7=Up F8=Down F12=Cancel
F8=D

```

**S**

このコマンドを使用すると、表示するビューを選択し、自動的に直前の画面に戻ります。“[Fault History File or View]”行に、選択されたビュー名が指定されています。

選択されたビューを表示するには、PF3 を押します。

変更を加えずに、直前に表示された履歴・ファイルまたはビューに戻るには、PF12 を押します。

他の履歴・ファイルやビューを選択すると、Fault Entry List 画面の列構成が変わる可能性があります。

## Fault Entry List の列構成

Fault Entry List 画面に表示される障害情報は、有効な HistCols オプションにより決まります。HistCols オプションが使用されない場合は、デフォルトは [図 10 : Fault Entry List 画面の例](#) ページ 64 のとおりです。

COLS コマンドを入力するか、View メニューの Column Configuration オプションを選択すると、個々のユーザーが Fault Entry List 画面の情報を変更できます (このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー](#) ページ 100 を参照してください)。これで、[図 17 : Fault Entry List Column Configuration 画面の例](#) ページ 72 の例のように、Fault Entry List Column Configuration 画面が開きます。

図 17. Fault Entry List Column Configuration 画面の例

```

File View Services Help
-----
Fault Entry List Column Configuration                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Current Fault Entry List Column Configuration (Sample Data):

  Fault_ID Job/Tran User_ID Sys/Job Abend Date      Time
  F00249 IDIVPCOB FRED      MVSA      S0C7  2019/11/22 15:29:03

Column Configuration Settings:

{Below, you may change your Fault Entry List display column configuration. To
make a column visible, or to change its relative display position, enter a
non-zero positive value in the Order column; to hide a column, enter 0. The
resulting column configuration is shown above.}

Order Column
  1  Fault_ID
  2  Job/Tran
  3  User_ID
  4  Sys/Job
F1=Help      F3=Sa      F8=Down
F10=Left     F11=Right  F12=Cancel

Default column configuration used  ①

```

Fault Entry List Column Configuration 画面は、2つのセクションに分かれています。

- 最初のセクションは、Current Fault Entry List Column Configuration セクションです。このセクションは、現行の列構成を見出しとサンプル・データで表示します。このセクションでは、最初に画面を横方向にスクロールしなくても、選択された列のどれが Fault Entry List 画面に表示されるのかを確認できます。
- 2番目は、Column Configuration Settings セクションです。このセクションでは、Fault Entry List 画面で使用される列を変更できます。

列を可視にしたり、相対的な表示位置を変更するには、ゼロ以外の正の値を Order 列に入力してください。列を非表示にするには、0 を入力します。

Enter キーを押した後、変更した結果の列構成が、Current Fault Entry List Column Configuration セクションに表示されます。

Fault\_ID 列を非表示することはできません。表示位置が指定されない場合、デフォルトの先頭の列に指定されます。

Fault Entry List Column Configuration 画面が最初に表示されるとき、現行の列構成を読み取った場所を示すメッセージが発行されます (図 17: Fault Entry List Column Configuration 画面の例 ページ 72 の ① を参照してください)。次の4つの可能性があります。

#### ・デフォルトの列構成を使用

この値は、ヒストリー・ファイル、または、有効な -HistCols が指定されていないビューが Fault Entry List 画面で選択され、さらにユーザーの ISPF プロファイルでデフォルト構成への変更が保管されていないことを示します。デフォルトの列構成は、有効な FAISPFopts(HistCols(...)) オプションにより決まります。

この構成に変更が加えられると、変更は、ユーザーの ISPF プロファイルに汎用の列構成として保管されます。



RESET コマンド (PF4) を入力しても、影響はありません。

• **ユーザー・プロファイルから読み取られた汎用の列構成**

この値は、ヒストリー・ファイル、または、有効な -HistCols が指定されていないビューが Fault Entry List 画面で選択され、さらにユーザーの ISPF プロファイルに汎用の列構成が存在することを示します。

この構成を変更すると、ユーザーの ISPF プロファイルの汎用の列構成と置き換えられます。

RESET コマンド (PF4) を入力すると、列構成がデフォルト構成にリセットされます。

• **ビュー・メンバーの *member-name* から読み取られた列構成**

この値は、有効な -HistCols が指定されたビューが Fault Entry List 画面で選択され、さらにユーザーの ISPF プロファイルに変更が保管されていないことを示します。

この構成が変更されると、その変更内容はユーザーの ISPF プロファイルにビュー固有の列構成として保管されます。

RESET コマンド (PF4) を入力しても、影響はありません。

• **ユーザー・プロファイルから読み取られたビュー *member-name* の固有の列構成**

この値は、有効な -HistCols が指定されたビューが Fault Entry List 画面で選択され、さらにユーザーの ISPF プロファイルにビュー固有の列構成が存在することを示します。

この構成が変更されると、ユーザーの ISPF プロファイルにおいてビュー固有の列構成が変更内容で置き換えられます。

RESET コマンド (PF4) を入力すると、列構成はビューの -HistCols 指定にリセットされます。

SAVE コマンド (PF3) を使用すると、現行の列構成を ISPF プロファイルに保管し、新しい構成がアクティブになった Fault Entry List 画面に戻ります。それ以降の対話式 Fault Analyzer セッションはすべて、以後の修正によって変更されるか、RESET コマンドを使用してデフォルトにリセットされるか、ISPF プロファイルが削除されるまで、この構成を使用します。

CANCEL コマンド (PF12) を使用すると、変更を保管せずに Fault Entry List Column Configuration 画面から元の画面に戻ります。

## 選択可能な列

選択可能な列ごとに表示される情報は、次のとおりです。

表 4. 障害項目リスト列

列のラベル	データ型	説明
Fault_ID	文字	障害に割り当てられた ID。
異常終了	文字	最初の (複数ある場合) 異常終了コード。

表 4. 障害項目リスト列 (続く)

列のラベル	データ型	説明
		IDISNAP を使用して作成された障害項目の場合は、異常終了コード「SNAP」が示されます。
Appl_ID	文字	CICS アプリケーション ID。
CICS_Trn	文字	障害が発生した CICS トランザクション ID。この列は、CICS のみに適用されます。
Class	文字	ジョブが実行していたジョブ・クラス。
CPU_sec		障害項目の作成中にレポート処理の終了時に使用される CPU 秒数。
Date	日付	LOCALE オプション従属フォーマットで障害が発生した日付。
Date_Time	日付と時刻	この列は日付と時刻を組み合わせた列で、日付と時刻の値は空白で区切られています。
DTFJ_Status	文字	該当する場合、Java 分析 DTFJ 処理ステータス。  ダンプ・エラー 保留中 開始済み 終了 不合格
Dups	整数型	重複した障害が検出された数。重複の判別に関する詳細は、 <a href="#">NoDup ページ 611</a> を参照してください。
Dup_Count	整数型	推奨されていません。代わりに、Dups を使用してください。
Dup_Date	日付	LOCALE オプション従属フォーマットで重複障害が最近発生した日付。重複が発生しなかった場合、この日付は初期異常終了日付 (『Date』を参照) に設定されます。
Dup_Date_Time	日付と時刻	この列は Dup_Date と Dup_Time を組み合わせた列で、日付と時刻の値は空白で区切られています。
Dup_Time	文字	LOCALE オプション従属フォーマットで重複障害が最近発生した時刻。重複が発生しなかった場合、この時刻は初期異常終了時刻 (『Time』を参照) に設定されます。

表 4. 障害項目リスト列 (続く)

列のラベル	データ型	説明
EXEC_Pgm	文字	JCL EXEC PGM= パラメーターからの異常終了ジョブ・ステップ・プログラム名。
Group_ID	文字	セキュリティー・サーバー・デフォルト・グループ ID。
History_File_DSN	文字	表示された障害項目を含むヒストリー・ファイルのデータ・セット名。
I_Abend	文字	Fault Analyzer が起動される原因となった異常終了コード。  IDISNAP を使用して作成された障害項目の場合は、異常終了コード「SNAP」が示されます。
IMS_Pgm	文字	IMS™ を伴う障害の IMS™ アプリケーション・プログラム名。
Job/Tran	文字	CICS® トランザクションの異常終了の場合、この値は CICS® トランザクション ID です。それ以外の場合は、異常終了したジョブの名前です。この列は、Jobname 列と CICS_Trn 列の情報を組み合わせます。
Job_ID	文字	異常終了したジョブの JES ジョブ ID。
Job_Type	文字	異常終了したジョブのタイプ (以下のいずれか)。  <b>Batch</b> バッチ・ジョブ  <b>CICS®</b> CICS® トランザクション  <b>CICSsys®</b> CICS® システム  <b>DumpReg</b> ダンプの登録  <b>STC</b> 開始済みタスク  <b>TSO</b> TSO ユーザー
Jobname	文字	異常終了したジョブの名前。
Lock	文字	障害項目ロック・フラグ。このフラグの目的については、 <a href="#">障害項目情報の表示 ページ 134</a> を参照してください。

表 4. 障害項目リスト列 (続く)


列のラベル	データ型	説明
Locked	文字	現在の障害項目のロックの状態を示します。  <b>あり</b> 障害項目はロックされています。  <b>なし</b> 障害項目はロックされていません。
Lock_Usr	文字	Lock フラグを最後に変更したユーザー ID。
Minidump	文字	次のようなミニダンプの可用性の表示。  <b>あり</b> ミニダンプが使用可能  <b>なし</b> ミニダンプが使用不可
Module	文字	障害点モジュール名。  ダンプ登録障害項目の場合、この値は、エラーに関与したロード・モジュールとして SVC ダンプ・ヘッダー SDWA 内で識別されたモジュールの名前です。この値は、初期再分析と障害項目リフレッシュの後で、障害点モジュール名に更新されます (別の名前になる場合があります)。
MD_Pages	整数型	障害のために保管されたミニダンプ・ページの数。
MVS_Dump	文字	次のような MVS™ ダンプの可用性の表示。  <b>あり</b> MVS™ ダンプが使用可能である  <b>なし</b> MVS™ ダンプが使用不可である   <b>注:</b> この列は、MVS™ ダンプ・データ・セットが異常終了時に使用可能であったかどうかを示します。関連する MVS™ ダンプ・データが削除されたか、障害項目が表示されているシステム上に存在しなくなった可能性があります。
MVS_Dump_DSN	文字	障害発生と同時期に書き込みのあった関連 MVS™ ダンプ・データ・セットの名前。

表 4. 障害項目リスト列 (続く)


列のラベル	データ型	説明
		 <b>注:</b> この列は、MVS™ ダンプ・データ・セットが異常終了時に使用可能であったかどうかを示します。関連する MVS™ ダンプ・データが削除されたか、障害項目が表示されているシステム上に存在しなくなった可能性があります。
Netname	文字	CICS® トランザクション・ネット名。
Offset	16 進	障害点オフセット。
Program	文字	障害点プログラム名。この列には、最大 8 文字のプログラム名を表示できます。8 文字を超えるプログラム名は切り捨てられます。
Program_(long)	文字	障害点プログラム名。この列には、最大 32 文字のプログラム名を表示できます。
Spilled	文字	関連する XDUMP データ・セットにスピルされたミニダンプ・ページが含まれているかどうかを示します。  <b>あり</b>  XDUMP にスピルされたミニダンプ・ページが含まれています。  <b>なし</b>  XDUMP データ・セットにスピルされたミニダンプ・ページが含まれていません。
Stepname	文字	異常終了したジョブのジョブ・ステップ名。
Sys/Job	文字	CICS® トランザクションの異常終了の場合、この名前は CICS® ジョブ名です。それ以外の場合は、ジョブが異常終了したシステムの ID です。この列は、System 列と Jobname 列の情報を結合します。
System	文字	異常終了が発生したシステム ID。
Task	文字	障害が発生した CICS® トランザクション・タスク番号。この列は、CICS® 障害のみに適用されます。
Term_ID	文字	CICS® トランザクション端末 ID。
Time	時刻	LOCALE オプション従属フォーマットで障害が発生した時刻。
Tran_ID	文字	推奨されていません。代わりに CICS_Trn を使用してください。

表 4. 障害項目リスト列 (続く)

列のラベル	データ型	説明
User_ID	文字	CICS® および IMS™ メッセージ処理領域 (MPP) の場合、異常終了するトランザクションに関連するユーザー ID。その他の場合は、異常終了ジョブに関連するユーザー ID。
User_Title	文字	ユーザー保守のタイトル情報。詳しくは、 <a href="#">障害項目情報の表示 ページ 134</a> を参照してください。
Username	文字	ユーザー保守の名前情報。詳しくは、 <a href="#">障害項目情報の表示 ページ 134</a> を参照してください。

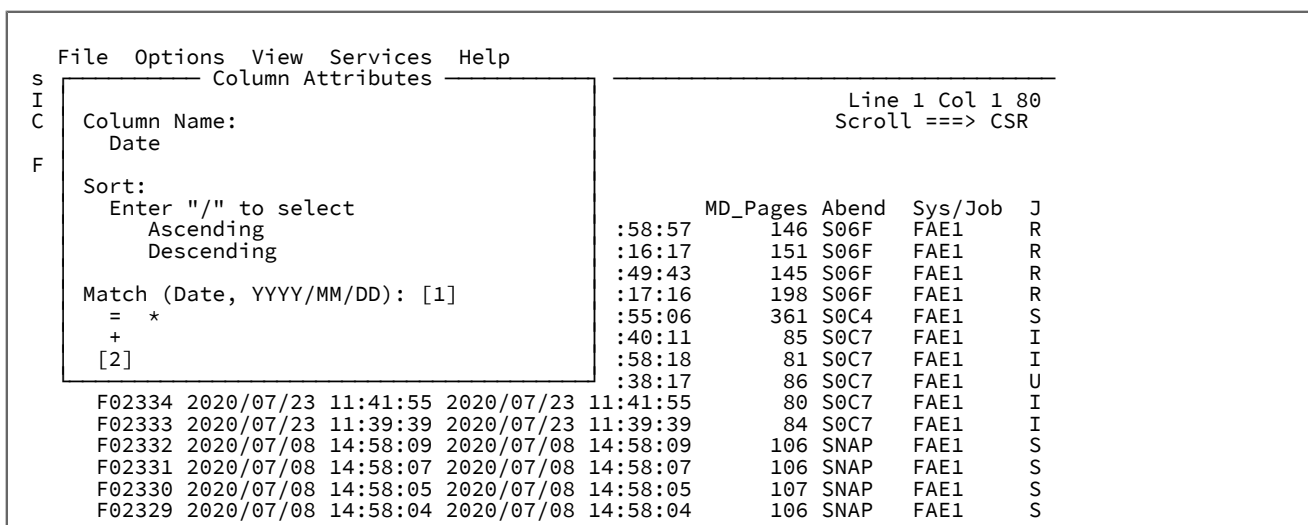
### 障害項目のソートと突き合わせ

列属性を使用して、類似したジョブ名または同じ異常終了コードなど、所定の一致基準を満たす障害項目のサブセットを表示することができます。障害項目の完全なリスト、または基準に一致する項目のサブセットを昇順または降順にソートすることができます。列属性の突き合わせとソートを行うと、例えば、類似したパターンを使用して障害を検索したり、項目を連続したグループに集めて一定の範囲の項目を削除したりするのに役立ちます。

列属性を表示するには、「Fault Entry List」画面の列見出しにタブを移動し、カーソルを見出しに置いて、Enter キーを押します。Column Attributes 画面を使用して、列のソート順、列の一致基準、またはその両方を指定します。最初に、Column Attributes 画面には、デフォルトの一致演算子 (=) とデフォルトの一致値 (\*) を指定する単一の一致基準が表示されます。デフォルトの一致基準は、画面内のすべての項目と一致します。

図 18: 「Date」列が選択された Column Attributes 画面の例 ページ 78 では、「Date」列が選択されています。

図 18. 「Date」列が選択された Column Attributes 画面の例



- 列のすべての障害項目をソートするには、昇順または降順の属性入力フィールドにスラッシュ (/) を入力して Enter キーを押します。
- 基準に一致しない障害項目を画面から削除するには、一致基準を指定して Enter キーを押します。
- 基準に一致する障害項目のサブセットにソート順を適用するには、一致基準と一致する項目のソート順序の両方を指定して、Enter キーを押します。

## Match 演算子

デフォルトの Match 演算子 (=) を上書きして変更します。有効な Match 演算子は次のとおりです。

```
=
    等しい

!=
    等しくない

>
    次より大きい

>=
    次以上

<
    次より小さい

<=
    次以下
```

次のいずれかが当てはまる場合、Match 演算子は = または != に制限されます。

- 列のデータ・タイプが文字である。
- 一致値全体が単一の \* 文字である場合を除き、一致値に 1 つ以上のワイルドカード文字が含まれる。ワイルドカード文字の詳細については、[一致値 ページ 79](#)を参照してください。

## 一致値

一致値には大/小文字の区別がなく、ワイルドカードを含めることができます。

```
*
    アスタリスクは、ゼロ、1 文字、または複数の文字を表します。

%
    % 記号は、単一文字を表します。
```

デフォルトの一致値 \* は、任意のデータ値に一致します。

一致値にワイルドカード文字が含まれている場合、一致値全体が単一の \* 文字である場合を除き、すべてのデータ型は文字データとして評価されます。

Column Attributes 画面では、選択した列のデータ・タイプが Match ヘッダーの括弧内に表示されます (図 18: 「Date」列が選択された Column Attributes 画面の例 ページ 78 の [1] を参照)。指定する一致値は、列のデータ型に有効でなければなりません。

表 5. 列属性のデータ型と値

データ型	有効な値
文字	任意の文字。
日付 <i>date_format</i>	<p>一致値にワイルドカード文字が含まれていない場合は、以下のいずれかとして日付を指定します。</p> <ul style="list-style-type: none"> <li>• ロケール固有の形式の日、月、および年。</li> <li>• <code>TODAY</code> キーワードを使用して指定する、現在の日付または現在の日付に対して相対的な日付。</li> </ul> <p><code>TODAY</code> キーワードを使用しない限り、Date 一致値は、Match ヘッダーに表示される日付形式 (/文字など) に従っている必要があります。</p> <p><b>YY</b></p> <p>2 桁の年。</p> <p>69 ~ 99 の範囲の値は、20 世紀の年 (1969 ~ 1999) を表します。00 ~ 68 の範囲の値は、21 世紀の年 (2000 ~ 2068) を表します。</p> <p><b>YYYY</b></p> <p>4 桁の年。</p> <p><b>MM</b></p> <p>月 (01 ~ 12)。</p> <p><b>DD</b></p> <p>日 (01 ~ 31)。</p> <p><b>TODAY   TODAY-days</b></p> <p><code>TODAY</code> キーワードは、現在の日付 (<code>TODAY</code>) または現在の日付を基準にしたそれ以前の日付 (<code>TODAY-days</code>) と照合させます。</p> <p>例えば、Match ヘッダーが日付形式を <code>MM/DD/YYYY</code> として表示するとします。</p>



表 5. 列属性のデータ型と値 (続く)

データ型	有効な値
	<ul style="list-style-type: none"> <li>• 09/27/2020 は、有効な一致値の例です。</li> <li>• TODAY-2 と TODAY は、有効な一致値の例です。</li> <li>• 9/27/20 と 09.27.2020 は、無効な一致値の例です。</li> </ul>
時刻、 <i>time_format</i>	<p>一致値にワイルドカード文字が含まれていない場合は、時刻一致値を、区切り文字 (: 文字など) を含め Match ヘッダーに表示される形式で指定します。</p> <p><b>HH</b></p> <p>時間、12 時間クロック (1~12)。</p> <p><b>hh</b></p> <p>時間、24 時間クロック (0~24)。</p> <p><b>mm</b></p> <p>分 (0~59)。</p> <p><b>ss</b></p> <p>秒 (0~59)。</p> <p><b>ampm</b></p> <p>AM または PM (12 時間クロック) で表したロケール固有の時間。</p> <p>例えば、Match ヘッダーが時刻形式を HH: mm: ss ampm として表示するとします。</p> <ul style="list-style-type: none"> <li>• 01:25:59 PM は、有効な一致値の例です。</li> <li>• 13:25:59 は、無効な一致値の例です。</li> </ul>
日時、 <i>date_format time_format</i>	<p>日付と時刻の組み合わせの値。一致値にワイルドカード文字が含まれていない場合は、Match ヘッダーの日付形式で示される形式で日付値を指定します。時刻はオプションです。含まれている場合は、Match ヘッダーの時刻形式で示される形式で時刻値を指定します。</p>
10 進数	10 進数の 0~9、小数点 (.) または 10 進数のコンマ (,)、あるいは負符号 (-)。
整数型	10 進数の 0~9 または負符号 (-)。
16 進	16 進数の 0~9 および A~F。
アドレス	16 進数の 0~9 および A~F。

表 5. 列属性のデータ型と値 (続く)

データ型	有効な値
	<p>オプションで、1つのアンダースコア ( _ ) 文字をアドレスに含めることができます。含まれている場合:</p> <ul style="list-style-type: none"> <li>アンダースコアの左側の桁は、64 ビット・アドレスのビット 0~31 として右揃えされます。</li> <li>アンダースコアの右側の桁は、64 ビット・アドレスのビット 32~63 として右揃えされます。</li> </ul> <p>例:</p> <p><code>1_1</code></p> <p>アドレス <code>100000001</code> と同等です。</p>

### 複数の一致基準の使用

列ごとに最大 10 個の一致基準を指定できます。データ値は、一致するすべての基準を満たしている必要があります (論理 AND)。

- 一致基準を追加するには、+ ポイント・アンド・シュート・フィールドにカーソルを置き、Enter キーを押します。(図 18: 「Date」列が選択された Column Attributes 画面の例 ページ 78 の [2] を参照。)
- 一致基準を削除するには、演算子フィールドを消去して Enter キーを押します。

図 19: 複数の一致基準を指定している Column Attributes 画面の例 ページ 82 Date 列の複数の一致基準を表示します。

図 19. 複数の一致基準を指定している Column Attributes 画面の例

```

File Options View Services Help
-----
Column Attribute
-----
I
C Column Name:
  Date
F
Sort:
  Enter "/" to select
  Ascending      :58:57
  Descending     :16:17
Match (Date, YYYY/MM/DD):
  >= 2020/07/08 :49:43
  < TODAY-7     :17:16
  != *31        :55:06
  +             :17:16
  +             :53:33
  +             :34:40
  +             :40:11
  +             :58:18
  +             :38:17
F02334 2020/07/23 11:41:55 2020/07/23 11:41:55
F02333 2020/07/23 11:39:39 2020/07/23 11:39:39
F02332 2020/07/08 14:58:09 2020/07/08 14:58:09
F02331 2020/07/08 14:58:07 2020/07/08 14:58:07
F02330 2020/07/08 14:58:05 2020/07/08 14:58:05
F02329 2020/07/08 14:58:04 2020/07/08 14:58:04
MD_Pages Abend Sys/Job J
146 S06F FAE1 R
151 S06F FAE1 R
145 S06F FAE1 R
198 S06F FAE1 R
361 S0C4 FAE1 S
83 S0C7 FAE1 I
193 S06F FAE1 R
85 S0C7 FAE1 I
81 S0C7 FAE1 I
86 S0C7 FAE1 U
80 S0C7 FAE1 I
84 S0C7 FAE1 I
106 SNAP FAE1 S
106 SNAP FAE1 S
107 SNAP FAE1 S
106 SNAP FAE1 S
  
```

## 列属性の適用

ソートと突き合わせは、現在表示されているデータ行に適用されます。同じ列のソート値または一致値を 2 回変更した場合、つまり最初にある列を変更した後で別の列を変更した場合、2 回目のソートまたは一致は最初のソートまたは一致の後に表示されるデータ行に適用されます。

基準に一致しない障害項目が表示から削除された後、リセットを実行してそれらを復元する必要があります。表示をリセットするには、次のいずれかを実行します。

- リセット・ポイント・アンド・シュート・フィールドにカーソルを置き、Enter キーを押す。
- コマンド行に RESET コマンドを入力する。

図 20: 前の図の基準に一致する障害項目 ページ 83 図 19: 複数の一致基準を指定している Column Attributes 画面の例 ページ 82 で一致基準を適用した結果の例を示します。

図 20. 前の図の基準に一致する障害項目

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List          13 of 322 rows matched
Command ==>> _____ Scroll ==>> CSR

Fault History File or View : 'NWILKES.HIST'

  Fault_ID Date      Time      Date_Time      MD_Pages Abend  Sys/Job  J
  F60415 2020/08/07 09:55:06 2020/08/07 09:55:06 361 S0C4  FAE1    S
  F02343 2020/08/04 09:53:33 2020/08/04 09:53:33 83  S0C7  FAE1    I
  F02342 2020/08/03 15:34:40 2020/08/03 15:34:40 193 S06F  FAE1    R
  F02340 2020/08/03 09:40:11 2020/08/03 09:40:11 85  S0C7  FAE1    I
  F02335 2020/07/29 10:38:17 2020/07/29 10:38:17 86  S0C7  FAE1    U
  F02334 2020/07/23 11:41:55 2020/07/23 11:41:55 80  S0C7  FAE1    I
  F02333 2020/07/23 11:39:39 2020/07/23 11:39:39 84  S0C7  FAE1    I
  F02332 2020/07/08 14:58:09 2020/07/08 14:58:09 106 SNAP  FAE1    S
  F02331 2020/07/08 14:58:07 2020/07/08 14:58:07 106 SNAP  FAE1    S
  F02330 2020/07/08 14:58:05 2020/07/08 14:58:05 107 SNAP  FAE1    S
  F02329 2020/07/08 14:58:04 2020/07/08 14:58:04 106 SNAP  FAE1    S
  F02328 2020/07/08 14:58:02 2020/07/08 14:58:02 106 SNAP  FAE1    S
  F02327 2020/07/08 12:15:44 2020/07/08 12:15:44 211 S06F  FAE1    R

** Bottom of data.

```

## 障害を突き合わせて選択する追加の方法

障害項目のソートと突き合わせ ページ 78 で説明されている Column Attributes 画面の一致機能に加えて、次の方法で障害項目を突き合わせることができます。

- 一致する値をカーソルで選択
- 既存値の上書き入力
- MATCH コマンドの使用

これらの各方法は、Column Attributes 画面と統合されており、一致基準を明示的に入力した場合と同じように列の MATCH 基準を更新します。

## 一致する値をカーソルで選択

突き合わせる値にカーソルを移動して、PF4 (MatchCSR) を押すと、Fault Analyzer は、このフィールドの値を共有する障害を障害ヒストリー・ウィンドウに表示します。

例えば、[図 10 : Fault Entry List 画面の例 ページ 64](#)のサンプル画面で、最後に表示される項目の Abend 値にカーソルを移動して PF4 を押すと、異常終了が S0CB だった障害のみが新規画面に表示されます。その結果の項目リストは、次のようになります。

図 21. 1 回の突き合わせを行った後の Fault Entry List

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List                               ❶ 3 of 319 rows matched
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'IBMUSER.DEMO.HIST'
-----
{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).}

  Fault_ID Job/Tran User_ID Sys/Job ❷ Abend Date      Time
  ---      -
  F00294 DB2NL2   IBMUSER MVS2   S0CB  2019/02/20 14:42:29
  F00292 DB2LE2   IBMUSER MVS2   S0CB  2019/02/20 14:38:25
  F00049 DACBB001 JCULLEN MVS2   S0CB  2019/02/01 08:56:27

*** Bottom of data.

F1=Help      F3=Exit      F4=MatchCSR  F5=RptFind   F6=Actions   F7=Up
F8=Down      F10=Left     F11=Right   F12=MatchALL

```

注:

❶

(前に表示された行のうち)

選択された値に一致した行数を示すメッセージが出されます。このメッセージは、ファンクション・キーまたは Enter キーを押すと消えます。

❷

MATCH が現在アクティブになっている列の見出しが、強調表示されます。これらの見出しは、MATCH がリセットされるまで強調表示されたままです。

これで 2 番目の項目の User ID (ユーザー ID) 値にカーソルを移動して PF4 を押すと、新しい画面は次のようになります。

図 22. 2 回の突き合わせを行った後の Fault Entry List

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List                2 of 3 rows matched
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'IBMUSER.DEMO.HIST'

{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).}

  Fault ID Job/Tran User ID Sys/Job  Abend Date      Time
  ---
  -- F00294 DB2NL2  IBMUSER MVS2    SOCB  2019/02/20 14:42:29
  -- F00292 DB2LE2  IBMUSER MVS2    SOCB  2019/02/20 14:38:25

*** Bottom of data.

F1=Help      F3=Exit      F4=MatchCSR  F5=RptFind   F6=Actions   F7=Up
F8=Down      F10=Left     F11=Right   F12=MatchALL

```

突き合わせは限定的です。2 番目の突き合わせを適用する際に選択される障害は、最初の突き合わせを満たす障害のみに限定されます。例えば、ユーザー ID による突き合わせを行い、次に、ダンプ状況で突き合わせると、結果として特定の状況を持つ 1 人の所有者の項目のみが表示されます。この代わりに、状況による突き合わせを行った場合は、すべてのユーザー ID についてこの状況を持つ項目がすべて表示されます。

## 既存値の上書き入力

突き合わせのもう 1 つの方法は、既存の値を上書き入力する方法です。例えば、「Abend」列に障害項目に対して SOCB という値が含まれる場合、4 を 1 に上書き入力して値を SOC1 にし、Enter キーを押します。これにより、実行された MATCH は異常終了値 SOC1 を持つ障害項目のみを表示するようになります。

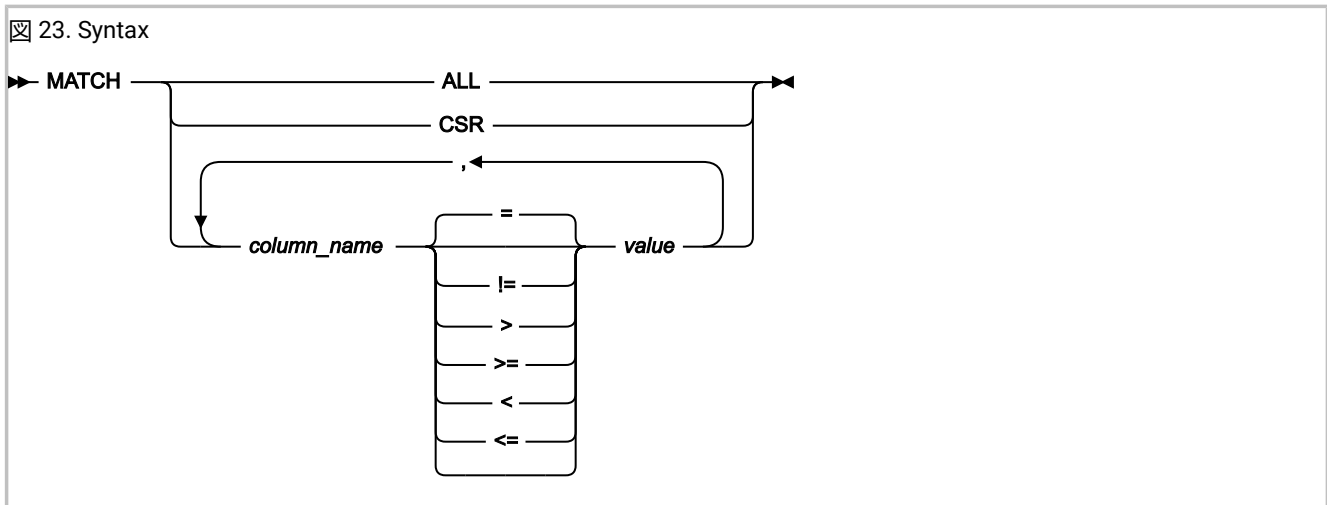
ワイルドカード文字を使用して、汎用 MATCH 値を指定できます。サポートされているワイルドカード文字にはアスタリスク (\*) と % 記号があり、それぞれ 0 個以上の文字と 1 文字のみを意味しています。

Enter キーを押す前に、任意の数の値を上書き入力できます。ただし、同じ列の値が複数の行で上書き入力された場合、その列に対して最後に上書き入力された値が使用されます。

既存の値を上書きする方法は、既に表示されている値によく似た値で突き合わせを行う場合に、特に便利です。単に表示されている値を目的の値に上書き入力し、Enter キーを押すだけで MATCH が実行されるため、最低限の入力操作だけで済みます。

## MATCH コマンドの使用

MATCH コマンドの構文は次のとおりです。



**注:** コンマまたは空白文字のいずれかを区切り文字に使用できます。

上記構文図では、*column\_name* は [選択可能な列 ページ 73](#) に示す列名のいずれでもかまいません (例えば、Abend または IMS\_Pgm)。列名および指定値の大/小文字は区別されません。

MATCH ALL (PF12 に同じ) は、制限を含むすべての一致条件を除去し、最初に履歴・ファイルを表示した時と同じ項目を表示します。このオプションは、REFRESH と同じ動作はしません。REFRESH は履歴・ファイルを参照するため、初回の Fault Analyzer の始動時以降に履歴・ファイルに書き込まれた新しい項目を表示できます。また、リフレッシュした場合はすべての一致条件が取り除かれます。

MATCH CSR はカーソル一致です。これを機能させるには、カーソルを値に移動し、Enter キーを押します。これは、カーソルを置いて PF4 を押すのと本質的に同じです ([一致する値をカーソルで選択 ページ 84](#) を参照してください)。

その他のキーワードはフィールドに対応し、値を使用してフィールド名を判別できます。オプションで、フィールド名と値の間に演算子を指定できます。デフォルトの演算子は = です。突き合わせの時には、値の大/小文字は区別されていません。

値に空白、コンマ、または二重引用符 (") が含まれている場合は、その値を二重引用符で囲みます。引用符付きストリング内の二重引用符はすべて、二重引用符を重ねる必要があります。例えば、次のユーザー・タイトルが表示されたとします。

```
A "B",C
```

この場合、次のように MATCH コマンドを指定します。

```
MATCH USER_TITLE "A ""B"",C"
```

\* はワイルドカードとして使用できます。これを値に追加すると、Fault Analyzer は、\* の前に入力された値で始まるすべての値を突き合わせます。すべての値はストリングであるため、例えば次のように入力できます。

```
MATCH DATE 2019/07*
```

これにより 2019 年 7 月のすべての項目が表示されます。

サポートされているもう 1 つのワイルドカード文字はパーセント (%) 記号で、1 つの必須文字を表す場合に使用します。

列の一致する値は、列に表示されるデータと同じ形式で指定します。例えば、日付列に年/月/日という形式の日付が含まれている場合、一致する値は同じように年/月/日という形式でなければなりません。

データ・タイプおよび有効な値について詳しくは、[障害項目のソートと突き合わせ ページ 78](#)を参照してください。Column Attributes 画面で一致基準に適用される規則は、MATCH コマンドにも適用されます。

例:

「日時」列に `month.day.year` の形式の日付と、`hours:minutes:seconds` の形式の時刻が含まれている場合、以下の例が有効な一致する値です。

```
1.13.2020
06.5.2020 0:0:0
TODAY
TODAY-365
```

**Date** 列または **Date & Time** 列で TODAY キーワードを使用して、現在の日付 (`TODAY`) または現在の日付に相対的な日付の範囲 (`TODAY-days`) のいずれかを指定します。例:

```
TODAY
TODAY-10
```

`TODAY-days` を指定する場合、空白や他の区切り文字はマイナス記号の前後には使用できません。

MATCH コマンドの一部として入力する値は、表示されている必要はありません。つまり、MATCH に対しては、画面の可視領域にある列の値を使用することも、現行のスクロール・ウィンドウ外の列の値を使用することもできます。ただし、現在、画面に選択されている列に対してのみ MATCH を実行できます。表示するように選択されていない列での一致する値は無視されます。

一致する値を適用しても、これに該当する項目がなければ、Fault Analyzer は、メッセージ “No matches” を表示し、項目のない画面を表示します。

## 特定の障害へのアクションの適用

項目に対して行コマンドを入力することにより、特定の障害にアクションを適用できます。使用可能なアクションを以下に示します。

### A - ソースの表示

LangxCapture

オプションが有効になっている場合、リアルタイム分析中にキャプチャーされたソース・レベルのデバッグ情報を表示します。[LangxCapture ページ 608](#) を参照してください。

### B - バッチ再分析

バッチ・ジョブを実行依頼して、選択された障害項目を再分析します。分析レポートが、SYSPRINT に書き込まれます。

詳しくは、[バッチ再分析の実行 ページ 154](#)を参照してください。

#### C - コピー

障害項目を別の履歴・ファイルにコピーします。

詳しくは、[履歴・ファイル項目のコピー ページ 143](#)を参照してください。

#### D - 削除

履歴・ファイルから障害項目を削除します。項目を削除すると、即時に Fault Entry List 画面からその項目が削除され、それ以降は最新表示をしても表示されることはありません。

詳しくは、[履歴・ファイル項目の削除 ページ 131](#)を参照してください。

#### H - 重複履歴

選択された障害項目の重複とみなされた既存の障害に関する詳細が表示されます (その詳細が入手可能な場合)。

障害項目に対して重複の詳細が使用可能な場合は、当該項目の Dups 列の値がポイント・アンド・シュート・フィールドになります。この場合、障害項目に対して H 行コマンドを入力すると、この項目に対する Dups 列の値にカーソルを置いて Enter キーを押した場合と同じに効果があります。

詳しくは、[障害項目の重複履歴の表示 ページ 139](#)を参照してください。

#### I - 対話式再分析

選択された障害に対して対話式再分析を実行します。しばらくすると、対話式レポートが表示されます。対話式レポートによってリアルタイム分析レポートが置き換えられることはありません。

詳しくは、[対話式再分析の実行 ページ 162](#)を参照してください。

#### J - JCL の編集

JclCapture オプションが有効になっている場合、リアルタイム分析中にキャプチャーされた JCL を表示および編集します。このコマンドは、対話式再分析で使用される JCL コマンドと同じ機能を実行します。[JCL ページ 111](#) および [JclCapture ページ 607](#)を参照してください。

#### L - ロック

デフォルトのロック・フラグ設定を使用して、障害項目をロックします。

デフォルトのロック・フラグ設定を変更するには、[Fault Analyzer プリファレンス ページ 117](#)を参照してください。

#### M - 移動

障害項目を別の履歴・ファイルに移動します。

詳しくは、[履歴・ファイル項目の移動 ページ 145](#)を参照してください。



**P - 障害項目のパッケージ化**

JCL を生成し、関連する障害項目データを Fault Analyzer 製品サポート・チームへの送信に最適な terse 形式データ・セットにパッケージ化します。JCL は編集セッションで表示されるため、修正および送信できます。

詳しくは、[障害項目のパッケージ化 ページ 146](#) を参照してください。

**U - アンロック**

障害エントリーのロック解除。

**V (または S) - レポートの表示**

保管された障害分析レポートの表示

詳しくは、[保管レポートの表示 ページ 125](#) を参照してください。

**X - XMIT**

障害項目を、指定されたユーザー ID およびノードに XMIT します。

詳しくは、[ヒストリー・ファイル項目の送信 ページ 145](#) を参照してください。

**? - 障害項目情報の表示**

障害項目の情報を表示します。特に、この情報には関連 MVS™ ダンプ・データ・セット名が示されます (そのダンプ・データ・セット名がある場合)。

詳しくは、[障害項目情報の表示 ページ 134](#) を参照してください。

項目に対して行コマンドを入力し、Fault Analyzer がこのコマンドを完了できない場合、行コマンドは行からクリアされません。この状態の例をいくつか以下に示します。

- 関連ダンプ・データ・セットがない障害に対してバッチ・ダンプ再分析を実行しようとした。
- 当該ダンプ・データ・セットが使用不可である。

Enter を押す前に、複数の項目に対して行コマンドを入力できます。この場合、Fault Analyzer は、最上部の項目を始めとして、各コマンドを受け入れようとしています。Fault Analyzer がコマンドを受け入れることができない場合は、以下の状況が発生します。

- Fault Analyzer は処理を停止します。
- Fault Analyzer は、処理できた各項目から行コマンドをクリアします。
- Fault Analyzer は、処理できなかった各項目の行コマンドを残すか、またはコマンドを受け入れることができなかった対象項目の行コマンドを残します。

**ヒストリー・ファイルのプロパティ**

現在選択されている障害ヒストリー・ファイルの属性および統計を表示するには、まずアクション・バーから **File > Fault History File Properties** を選択します。

これにより、Fault History File Properties 画面が表示されます。

図 24. Fault History File Properties 画面の例

```

Fault History File Properties
Command ==> _____ Line 1 Col 1 76
                               Scroll ==> CSR

Enter the Exit command (PF3) to return to the fault history file display.

History File Name . . . . . : DA.DCAT
History File Type . . . . . : PDSE (Library type 1)
Primary Space . . . . . : 1,000 Cylinders (180,000 Pages)
Secondary Space . . . . . : 200 Cylinders ( 36,000 Pages)
Allocated Space . . . . . : 4,347 Cylinders (782,460 Pages)
Used Space. . . . . : 599,907 Pages
Allocated Extents . . . . . : 27
Data Set Utilization. . . . : 76.67%
History File Access . . . . : Update
Fault ID Prefix . . . . . : F
Logical History File Size . : 600,000 Pages (76.68% of Allocated Space)
History File Utilization. . : 99.98%
Minimum Fault Entries . . . : 25,ThenAUTO
Current Fault Entries . . . : 4,192
Number of Entries With
  Minidump. . . . . : 4,092 (97.61% of Total)
Maximum Minidump Size . . . : 2,840 Pages
Enter the Exit command (PF3) to return to the fault history file display.

```

この画面から、次の情報が入手できます。

#### 割り振られたエクステント

ヒストリー・ファイルに割り振られたエクステントの総数。ユーザーがヒストリー・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

#### 割り当て済みスペース

元の割り振りユニットで割り振られたデータ・セット・スペース合計数。PDSE ヒストリー・ファイルの場合、相当するページ数も表示されます。ユーザーがヒストリー・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

#### Average Minidump Size

このヒストリー・ファイルまたはビューに保管されているすべてのミニダンプの平均サイズが、ミニダンプ・ページ数として表示されます。

#### Current Fault Entries

ヒストリー・ファイル内の現在の障害項目数。

#### データ・セット使用率

使用されたページを、割り振られたページに対する割合 (パーセント) として表したものの。PDSE ヒストリー・ファイルのみで使用できます。ユーザーがヒストリー・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

#### Fault ID Prefix

ヒストリー・ファイル内のすべての障害項目に割り当てられた障害 ID 接頭部。この接頭部は、SETFAULTPREFIX 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーで変更できます ([ヒストリー・ファイルの管理 \(IDIUTIL](#)

ユーティリティー) ページ 441を参照してください)。また、この設定は、Fault Analyzer ISPF インターフェースを使用して変更できます (障害履歴・ファイル設定の変更 ページ 94を参照してください)。

### History File Access

Read または Update。

### 履歴・ファイル名

明示的に表示される履歴・ファイルの名前、またはビューに含まれる履歴・ファイルの名前。

### History File Type

この名前は次のいずれかとして示されます。

- PDSE (Library)
- PDSE (Library タイプ 1)
- PDSE (Library タイプ 2)
- PDS (Partitioned Data Set)

### 履歴・ファイルの使用率

履歴・ファイル・サイズのパーセントとしての使用スペース。(PDSE 履歴・ファイルに対してのみ表示されます。)ユーザーが履歴・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

### 論理履歴・ファイル・サイズ

使用可能な場合、現在の論理履歴・ファイル・サイズ。ユーザーが履歴・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

### 障害項目の最小数

PDSE

履歴・ファイルが自動的に保守される前に履歴・ファイルに存在していなければならない障害項目の数

(自動保守を実現するために割り振られたデータ・セット・エクステンツの数は無関係)。履歴・ファイルが自動的に維持されるようになると、障害項目の数は、現在使用可能なデータ・セット・スペースによってのみ制限されます。これ以上のデータ・セット・エクステンツは通常は割り振られず、スペース不足状態は発生しないものと予期されています。

障害項目の最小数は、IDIUTIL バッチ・ユーティリティーの SetMinFaultEntries

制御ステートメントを使用して変更できます (履歴・ファイルの管理 (IDIUTIL

ユーティリティー) ページ 441を参照してください)。また、この設定は、Fault Analyzer ISPF

インターフェースを使用して変更できます (障害履歴・ファイル設定の変更 ページ 94を参照してください)。

### 障害項目の最大数

最も古い項目の自動削除が行われるまでに、この履歴・ファイル内に保持される障害項目の最大数。

この障害項目数を達成するために、必要に応じて追加のデータ・セット・エクステン트가割り振られます。ヒストリー・ファイルに *nnn* の障害項目が入る前にデータ・セット内に使用可能なスペースがなくなった場合（つまり、データ・セット・エクステン트의最大数に達するか、ボリュームが満杯になった場合）、スペース不足状態が発生します。

n/a の場合は、ヒストリー・ファイルに障害項目の最大数が割り振られていません。

PDS ヒストリー・ファイルの場合、障害項目の最大数は IDIUTIL バッチ・ユーティリティの SetMaxFaultEntries 制御ステートメントで変更でき、PDSE ヒストリー・ファイルの場合、IDIUTIL バッチ・ユーティリティの SetMinFaultEntries 制御ステートメントで変更できます（[ヒストリー・ファイルの管理 \(IDIUTIL ユーティリティ\)](#) ページ 441 を参照）。また、この設定は、Fault Analyzer ISPF インターフェイスを使用して変更できます（[障害ヒストリー・ファイル設定の変更 ページ 94](#) を参照してください）。

### Maximum Minidump Size

このヒストリー・ファイルまたはビューに保管されている最大ミニダンプが、ミニダンプ・ページ数として表示されます。

### Minimum Minidump Size

このヒストリー・ファイルまたはビューに保管されている最小ミニダンプが、ミニダンプ・ページ数として表示されます。

### Number of Entries With Associated MVS™ Dump

関連した MVS™ ダンプ・データ・セットを伴う、ヒストリー・ファイルまたはビュー内の項目の数。合計項目数に対する割合（パーセント）も表示されます。

### Number of Entries With Minidump

保管されたミニダンプを含む、ヒストリー・ファイルまたはビュー内の項目の数。合計項目数に対する割合（パーセント）も表示されます。

### Number of History Files

ビューが表示される場合、そのビューに含まれるヒストリー・ファイルの数。個々のヒストリー・ファイルに関する情報がその下に表示されます。

### 1 次スペース

元の割り振りユニットで割り振られた 1 次データ・セット・スペース。PDSE ヒストリー・ファイルの場合、相当するページ数も表示されます。ユーザーがヒストリー・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

### 2 次スペース

元の割り振りユニットで割り振られた 2 次データ・セット・スペース。PDSE ヒストリー・ファイルの場合、相当するページ数も表示されます。ユーザーがヒストリー・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

### Total Number of Entries

ビュー内の項目の合計数。

## 使用スペース

PDSE ヒストリー・ファイルの場合、これは使用ページの合計数です。PDS

ヒストリー・ファイルの場合、これは使用した元の割り振りユニットの合計数です。ユーザーがヒストリー・ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

## ビュー名

ビューが表示される場合、そのビューの名前。

## 新しいヒストリー・ファイルの割り振り

新しいヒストリー・ファイルを割り振るには、まず File (ファイル) メニューの New Fault History File Allocation (新しい障害ヒストリー・ファイルの割り振り) オプションを選択します (このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。これで、[図 25: New History File Allocation \(新しいヒストリー・ファイルの割り振り画面の例 ページ 93\)](#)に示す例のように、「新しいヒストリー・ファイルの割り振り」画面が開きます。

図 25. New History File Allocation (新しいヒストリー・ファイルの割り振り画面の例)

```

New History File Allocation                               Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR
Press Enter to allocate new history file, or press PF3/PF12 to cancel.

History File Name . . . . . : _____
Primary Space . . . . . : 10          Cylinders
Secondary Space . . . . . : 25          Cylinders
Fault Entry Prefix. . . . . : F          (1-3 alphabetic characters)
Data Set Type . . . . . : LIBRARY      (LIBRARY or PDS)
Minimum Fault Entries . . . : 100       (25-9999999)

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F7=Up        F8=Down     F12=Cancel

```

この画面には、以下のフィールドが表示されます。

### History File Name

割り振られるヒストリー・ファイルの名前。この名前が単一引用符で囲まれていない場合、現行の TSO 接頭部が高位修飾子として使用されます。



**注:** Fault Entry List (障害項目リスト) の Move (移動) または Copy (コピー)

行コマンドに対して存在しないヒストリー・ファイルを指定した結果としてこの画面が表示された場合には、History File Name (ヒストリー・ファイル名)



フィールドは入力に対して使用できませんが、Move/Copy (移動/コピー) のターゲットの履歴・ファイル名が表示されます。

### Primary Space

1 次スペースとして割り振るシリンダーの数。デフォルトは 10 です。

### Secondary Space

2 次スペースとして割り振るシリンダーの数。デフォルトは 25 です。

### Fault Entry Prefix

この履歴・ファイル内で、すべての障害 ID に割り当てられる 1 文字から 3 文字の接頭部。デフォルトは F です。

### Data Set Type

割り振るべき履歴・ファイル・データ・セットのタイプ (以下のいずれか)。

- LIBRARY (PDSE)
- PDS (Partitioned Data Set)

デフォルトは LIBRARY です。

### Minimum Fault Entries

自動スペース管理が発生する前に、履歴・ファイルに存在する必要がある障害項目の最小数。この数は、25 から 9999999 の範囲内であればなりません。デフォルトは 100 です。



**注:** このフィールドは、「Data Set Type」フィールドに「LIBRARY」が示される場合にのみ表示されます。

### Maximum Fault Entries

履歴・ファイルで維持する障害項目の最大数 (25 から 9999999 までの数)。デフォルトは 100 です。



**注:** このフィールドは、「Data Set Type」フィールドに「PDS」が示される場合にのみ表示されます。

## 障害履歴・ファイル設定の変更

履歴・ファイルの設定を ISPF インターフェースから変更するには、履歴・ファイルが Fault Entry List 画面で選択されているか、または現在選択されているビューに含まれていることを確認してください。

障害 ID 接頭部あるいは障害項目の最小数または最大数を変更できます。最初に、アクション・バーから **File > Change Fault History File Settings** を選択します。

ビューが現在選択されている場合、そのビューに含まれている履歴・ファイルのリストから履歴・ファイルを選択するオプションが提供されます。

次に、「Change Fault History File Settings (障害ヒストリー・ファイル設定の変更)」画面(図 26 : Change Fault History File Settings 画面の例 ページ 95 に例が示されています)が提供されます。

図 26. Change Fault History File Settings 画面の例

```

Change Fault History File Settings                               Line 1 Col 1 76
Command ==> ----- Scroll ==> CSR

History File Name . . . . . : DA.DCAT
Data Set Type . . . . . : PDSE (Library type 1)
Primary Space . . . . . : 1,000 Cylinders (180,000 Pages)
Secondary Space . . . . . : 200 Cylinders ( 36,000 Pages)
Allocated Space . . . . . : 4,347 Cylinders (782,460 Pages)
Used Space. . . . . : 599,907 Pages
Allocated Extents . . . . . : 27
Current Number of Fault
  Entries . . . . . : 4192

Press Enter to change history file settings, or press PF3/PF12 to cancel.

Fault Entry Prefix. . . . . : F_____ (1-3 alphabetic characters)
Logical History File Size
(Pages) . . . . . : 600000 (0-4238460)
Minimum Fault Entries (*) . : 25 (25-9999999)

(*) When the total number of fault entries exceeds this value, then the
    history file is auto-managed.

```

この画面には、以下のフィールドが表示されます。

#### ヒストリー・ファイル名

設定変更の対象となるヒストリー・ファイル名。

#### データ・セット・タイプ

選択されたヒストリー・ファイル・タイプ (以下のいずれか)。

- PDSE (Library)
- PDSE (Library タイプ 1)
- PDSE (Library タイプ 2)
- PDS

#### 1 次スペース

元の割り振りユニットで割り振られた 1 次データ・セット・スペース。PDSE  
ヒストリー・ファイルの場合、相当するページ数も表示されます。ユーザーがヒストリー・ファイルに対する  
データ・セットの READ 権限を有している場合にのみ表示されます。

#### 2 次スペース

元の割り振りユニットで割り振られた 2 次データ・セット・スペース。PDSE  
ヒストリー・ファイルの場合、相当するページ数も表示されます。ユーザーがヒストリー・ファイルに対する  
データ・セットの READ 権限を有している場合にのみ表示されます。

**割り当て済みスペース**

元の割り振りユニットで割り振られたデータ・セット・スペース合計数。PDSE  
 ヒストリー・ファイルの場合、相当するページ数も表示されます。ユーザーがヒストリー・ファイルに対する  
 データ・セットの READ 権限を有している場合にのみ表示されます。

**使用スペース**

PDSE ヒストリー・ファイルの場合、これは使用ページの合計数です。PDS  
 ヒストリー・ファイルの場合、これは使用した元の割り振りユニットの合計数です。ユーザーがヒストリー・  
 ファイルに対するデータ・セットの READ 権限を有している場合にのみ表示されます。

**割り振られたエクステント**

ヒストリー・ファイルに割り振られたエクステントの総数。ユーザーがヒストリー・ファイルに対するデー  
 タ・セットの READ 権限を有している場合にのみ表示されます。

**Current Number of Fault Entries**

選択されたヒストリー・ファイルに現在存在する障害項目の数。

**障害項目接頭部**

1 から 3 文字までの英字で構成される接頭部。付加される障害項目番号との組み合わせで、このヒストリー・  
 ファイルの障害 ID を構成します。ヒストリー・ファイル内に新しく作成されたすべての障害エントリーに、  
 この接頭部が付加されます。ヒストリー・ファイルごとに異なる接頭部を使用することで、障害 ID を簡単に  
 識別できるようになります。

表示された障害項目接頭部の初期値は、当該ヒストリー・ファイルの現在の設定を常に反映しています。

**論理ヒストリー・ファイル・サイズ (ページ数)**

このフィールドは、PDSE  
 ヒストリー・ファイル用に表示されます。また、ユーザーがヒストリー・ファイルに対するデータ・セットの  
 READ 権限を有している場合にも表示されますが、これら以外の場合では表示されません。

この値は論理ヒストリー・ファイル・サイズを 4K  
 ページの数で指定し、ヒストリー・ファイルのスペース管理を支援するために Fault Analyzer  
 により使用されます。障害エントリーの最小数がヒストリー・ファイル内に作成されるまで、設定には 0 が表  
 示されます。

**PDSE ヒストリー・ファイルのサイズの変更 [ページ 349](#)**

で説明した手順を使用してゼロ以外の数を変更する場合、値を 0 に設定します。これにより、論理ヒスト  
 リー・ファイル・サイズは、ヒストリー・ファイルで指定された最小障害エントリー数の次の障害エン  
 トリーを作成するときに、現在割り振られているページ数に自動的に設定されます。


最大許容値は割り振り可能な論理的な最大ページ数を表し、2 次スペース・サイズ内の残りのすべてのエク  
 ステントを含む、最大データ・セット・エクステント数に到達していると推定します。ヒストリー・ファイルの  
 スペースが不足する可能性があるため、このサイズに近い値は指定しないでください。



指定した値が使用スペースの値より小さいか、割り当て済みスペースの値より大きい場合、警告が表示されません。


### 障害項目の最小数

自動スペース管理が発生する前に、PDSE  
ヒストリー・ファイルに存在する必要がある障害項目の最小数。この数は、25 から 9999999  
の範囲内でなければなりません。

 **注:** このフィールドは、「Data Set Type」フィールドに「PDSE (Library)」が示される場合にのみ表示されます。

### 障害項目の最大数

PDS ヒストリー・ファイルに維持する障害項目の最大数で、25 から 9999999 までの数。

 **注:** このフィールドは、Data Set Type フィールドに「PDS」を指定した場合にのみ表示されます。

表示された初期の最小障害項目数または最大障害項目数は、通常、当該ヒストリー・ファイルの現在の設定を反映します。ただし、以下の場合には、この値は推奨値です。

- ヒストリー・ファイルが PDS ファイルで、障害項目の最大数が設定されていない。この場合は、障害項目の最大数の初期値が 100 に設定される。
- ヒストリー・ファイルが PDSE ファイルで、障害項目の最小数が設定されていない。この場合は、障害項目の最小数の初期値が 100 に設定される。
- ヒストリー・ファイルが PDSE ファイルで、障害項目の最小数が設定されていないか、あるいは 25 未満である。この場合は、障害項目の最小数の初期値が 25 に設定される。

障害項目値の最小数を変更した場合、PDSE ヒストリー・ファイルは、以前に自動管理されていたか否かに関係なく、常に自動管理が使用可能になります。

必要なすべての設定を変更後、Enter キーを押して保管します。

保管せずに終了する場合は、PF3 または PF12 を押します。

この画面で提供される機能は、IDIUTIL バッチ・ユーティリティー SetFaultPrefix、SetMaxFaultEntries、および SetMinFaultEntries 制御ステートメントを使用した場合と同等です。

 **注:** アクション・バー・オプション **File > Change Fault History File Settings** は、現在表示されているヒストリー・ファイルまたは現行 View 内のすべてのヒストリー・ファイルに対するユーザーの管理者権限が制限されている場合



は選択できません。権限の制限については、[ヒストリー・ファイル設定の変更の制限 ページ 308](#)を参照してください。

## ヒストリー・ファイル・アクセス情報のリセット

以前にアクセスされたヒストリー・ファイルまたはビュー、および以前にアクセスされたヒストリー・ファイル項目についてのすべての情報をリセットするには、File メニューの Clear Last Accessed Information オプションを選択します (このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。

このオプションを選択した直後に、File メニューの Last Accessed Fault History File Entries オプションを選択しても、選択可能な項目がありません。しかし、File メニューの Last Accessed Fault History Files or Views オプションには、現在アクティブなヒストリー・ファイルまたはビューの項目が1つ表示されます。

## 障害項目情報の最新表示

ヒストリー・ファイルまたはビューを表示している間、例えば、異常終了したジョブのリアルタイム分析のために、新しい項目が追加される可能性があります。ヒストリー・ファイルまたはビューを再読み取りして、それらの項目を組み込むには、REFRESH コマンドを実行するか、「View」メニューの「Refresh」オプションを選択します (このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。

この方法で最新表示を実行すると、画面が再フォーマットされ、一番上の行の左端の列に移動します。最新表示の時点でアクティブであった MATCH コマンドのフィルター操作は、リセットされます。

## 暗黙最新表示

暗黙最新表示は、Fault Entry List 画面から Enter キーを押すと実行され、このとき、基本コマンドの入力、行コマンドの入力、または列データの上書きを行わないと、障害項目のリストはヒストリー・ファイルまたはビューから再読み込みされます。

## 保留状態の更新情報

Fault Entry List 画面を初回表示または最新表示するときの応答時間が長くなり過ぎるのを避けるため、表示される情報はすぐに使用可能なヒストリー・ファイルに限定されます。

すぐには使用可能でないヒストリー・ファイルの例は、現在表示中の MVS™ イメージとは別のイメージで使用中のファイルなどです。この場合、表示される情報は、ヒストリー・ファイル・データ・セットの \$\$INDEX メンバー内に現在ある障害項目に限定されます。しかし、他の MVS™ イメージ上の IDIS サブシステムによって管理される \$\$INDEX メンバーがキャッシュに入れられたままの状態、そのメンバーに新規または変更された障害が追加されたとしても、それらの情報は組み込まれません。別の IDIS サブシステムがヒストリー・ファイルの制御を解放し次第、更新された情報を表示できるようになります。

すぐには使用可能でないヒストリー・ファイルが1つ以上ある場合、次の例 (❶) に示すように、画面の先頭付近にメッセージが表示されます。

図 27. 保留状態の更新情報がある Fault Entry List 画面の例

```

File Options View Services Help
IBM Fault Analyzer - Fault Entry List                                Refresh complete
Command ==> ----- Scroll ==> CSR

Fault History File or View : (ALLTEST) All testing
6 history files might have updates pending. ①
  Fault_ID MD Pages Dups Dup Date Dup Time Abend I_Abend Module System
  ---
  IMS14434   96      2019/08/03 13:51:52 S0CB  U4039  IDCB0060 FAE1
  IMS14433   99      2019/08/03 13:51:40 SNAP  SNAP   IDCB0040 FAE1
  IMS14432   98      2019/08/03 13:51:26 S0CB  U4039  IDCB0020 FAE1
  IMS14431   51      2019/08/03 13:51:16 U0428  U0428  DFSPCC20 FAE1
  IMS14430   98      2019/08/03 13:51:01 S0CB  U4039  IDCB0020 FAE1
  IMS14429   99      2019/08/03 13:50:46 S0CB  U4039  IDCB0010 FAE1
  IMS14428  106      2019/08/03 13:50:33 S0CB  U4039  IDCB0090 FAE1
  IMS14427  735      2019/08/03 13:50:11 S0CB  U4039  IDCB0080 FAE1
  IMS14426  105      2019/08/03 13:49:54 S0CB  U4039  IDCB0070 FAE1
  IMS14424   99      2019/08/03 13:49:35 S0CB  U4039  IDCB0060 FAE1
  IMS14423   51      2019/08/03 13:49:26 U0456  U0456  DFSPCC20 FAE1
  IMS14422  100      2019/08/03 13:49:15 SNAP  SNAP   IDCB0040 FAE1
  IMS14421   98      2019/08/03 13:49:02 S0CB  U4039  IDCB0020 FAE1
  IMS14420   96      2019/08/03 13:48:48 S0CB  U4039  IDCB0020 FAE1
  IMS14419   51      2019/08/03 13:48:39 U0428  U0428  DFSPCC20 FAE1
  IMS14418   97      2019/08/03 13:48:26 S0CB  U4039  IDCB0010 FAE1

```

保留状態の更新情報がある履歴・ファイル数のメッセージ (①) 上にカーソルを置き、Enter キーを押すと、次の例のように History File Updates Pending 画面が表示されます。

図 28. History File Updates Pending 画面の例

```

File Options View Services Help
History File Updates Pending                                       Line 1 Col 1 76
Command ==> ----- Scroll ==> CSR

The most current information from the following history files might not be
included--press PF3, PF12, or Enter to continue, and refresh later by
pressing Enter again:

CTEST.DANLEPLI.DCAT
CTEST.DAPLRMVS.DCAT
CTEST.DAQJSMVS.DCAT
CTEST.DAQSCMVS.DCAT
CTEST.DAQSOMVS.DCAT

  IMS14427   735      2019/08/03 13:50:11 S0CB  U4039  IDCB0080 FAE1
  IMS14426  105      2019/08/03 13:49:54 S0CB  U4039  IDCB0070 FAE1
  IMS14424   99      2019/08/03 13:49:35 S0CB  U4039  IDCB0060 FAE1
  IMS14423   51      2019/08/03 13:49:26 U0456  U0456  DFSPCC20 FAE1
  IMS14422  100      2019/08/03 13:49:15 SNAP  SNAP   IDCB0040 FAE1
  IMS14421   98      2019/08/03 13:49:02 S0CB  U4039  IDCB0020 FAE1
  IMS14420   96      2019/08/03 13:48:48 S0CB  U4039  IDCB0020 FAE1
  IMS14419   51      2019/08/03 13:48:39 U0428  U0428  DFSPCC20 FAE1
  IMS14418   97      2019/08/03 13:48:26 S0CB  U4039  IDCB0010 FAE1

```

この画面には、最新情報が入手できていない可能性があるすべての履歴・ファイルの名前がリストされます。この画面から戻るには、PF3、PF12、または Enter キーを押してください。

最新情報が必要なときに保留状態の更新情報メッセージを受け取った場合でも、再度 Enter キーを押して暗黙最新表示を実行するときには、たいいてい最新情報が表示されます。

## 障害項目の有効期限の制御

障害項目の早すぎる削除を防止するために、障害項目を無期限にロックするか、障害項目の初期作成以後の指定した日数だけロックすることができます。どちらの場合も、この削除はロック・フラグによって制御されます。ロック・フラグは、Fault Entry Information 画面を使用して表示または変更できます (詳しくは、[障害項目情報の表示 ページ 134](#)を参照してください)。

ロック・フラグは、ENV.LOCK\_FLAG フィールドを介してユーザー出口によって設定することもできます (詳しくは、[ENV-共通出口環境情報 ページ 650](#)を参照してください)。

## アクション・バーのプルダウン・メニュー

Fault Analyzer ISPF インターフェースが使用する画面の大部分には、パネルの一番上にメニュー・バーがあります。ACTIONS ISPF コマンド (一部のディスプレイではデフォルトでは PF6 にマップ) を使用すると、選択可能な左端のアクションにカーソルが置かれます。ISPF 設定値によっては、タブ・キーを押すと、カーソルを他のアクションに移動できる場合があります。または、単に上/下/左/右の矢印キーを使用するだけで、目的のアクションにカーソルを置くこともできます。カーソルは、アクションが選択される以前に配置されていた画面上の位置に自動的に再配置されるため、PF キーを使用して ACTIONS コマンドを実行する方が便利です。

関連したプルダウン・メニューを表示するには、カーソルをメニュー・バーの項目に置いた後、Enter キーを押してください。

プルダウン・メニューのオプションを選択するには、関連したオプション番号を初期のカーソル位置に入力するか、(上/下/左/右の矢印キーを使用して) そのオプションの行の任意の場所にカーソルを置いて、Enter キーを押します。選択できないオプションはすべて、数字のオプション番号ではなく、アスタリスク (\*) によって示されます。

選択可能なプルダウン・メニュー・オプションの以下のリストでは、次のような形式を使用しています。

```
menu_name->menu_option->menu_option...
```

それぞれの意味は以下のとおりです。

### menu\_name

画面の一番上に表示されるアクション・バーのプルダウン・メニューの名前。

### menu\_option

最初のメニューおよび後続のメニューから選択できるオプションの名前。

以下に、選択可能なプルダウン・メニュー・オプションを、アルファベット順にリストします。

### File->Analyze MVS™ Dump Data Set

SYSMDUMP または SVC ダンプ・データ・セットの対話式分析を開始するために使用されます。主に、CICS® システム・ダンプ分析 ([CICS システム異常終了ダンプ分析の実行 ページ 251](#)を参照) または Java™ ダンプ分析 ([Java 分析の実行 ページ 262](#)を参照) を対象としています。

### File->Change Fault History File Settings

障害項目ヒストリー・ファイル設定における障害 ID 接頭部または障害項目の最大数を変更する場合に使用します。 [障害ヒストリー・ファイル設定の変更 ページ 94](#)を参照してください。

**File->Clear Last Accessed Information**

以前にアクセスされた履歴・ファイルまたはビュー、および以前にアクセスされた履歴・ファイル項目についての情報をリセットする場合に使用します。[履歴・ファイル・アクセス情報のリセット ページ 98](#)を参照してください。

**File->Exit**

このオプションを選択するのは、ISPF EXIT コマンドを発行するのと同じです。通常、このコマンドは原稿画面を終了して、呼び出し元の画面に戻ります。

**File->Exit Fault Analyzer**

Fault Entry List 画面を終了する場合に使用します。このオプションを選択することは、EXIT コマンドを実行する (または PF3 キーを押す) ことと同じです。

**File->Exit Interactive Reanalysis**

対話式再分析レポート内の任意の場所から、Fault Entry List 画面に戻る場合に使用します。このオプションを選択することは、EXIT コマンドを実行する (または PF3 キーを押す) ことと同じではありません。EXIT コマンドは、直前の画面にのみ戻ります。

**File->Fault Entry Information**

[図 72: Fault Entry Information 画面の例 ページ 135](#) に示された、Fault Entry Information 画面を開くために使用されます。このオプションを選択することは、INFO コマンド ([INFO ページ 111](#) を参照) を発行すること、または Fault Entry List 画面 ([特定の障害へのアクションの適用 ページ 87](#) を参照) から 1 つの障害項目に対して「？」行コマンドを入力することと同じです。

**File->Fault History File Properties**

現在選択されている履歴・ファイルまたはビューに関する、属性および統計情報を表示する場合に使用します。[履歴・ファイルのプロパティ ページ 89](#) を参照してください。

**File->Format CICS® Auxiliary Trace Data Set**

CICS® 補助トレース・データ・セットをフォーマットするために使用します。[CICS 補助トレース・データ・セットのフォーマット設定 ページ 260](#) を参照してください。

**File->Last Accessed Fault History File Entries**

以前にアクセスされた最大 10 個の履歴・ファイル項目のリストを表示する場合に使用します。詳しくは、[表示される履歴・ファイルまたはビューの変更 ページ 66](#) を参照してください。

**File->Last Accessed Fault History Files or Views**

以前にアクセスされた最大 10 個の履歴・ファイルまたはビューのリストを表示する場合に使用します。[表示される履歴・ファイルまたはビューの変更 ページ 66](#) を参照してください。

**File->List Views**

選択可能なすべてのビューのリストを表示する場合に使用します。[表示される履歴・ファイルまたはビューの変更 ページ 66](#) を参照してください。

### File->New Fault History File Allocation

新しいヒストリー・ファイルの割り振りを許可します。詳しくは、[新しいヒストリー・ファイルの割り振り ページ 93](#)を参照してください。

### 「ヘルプ」->「バージョン情報」 Fault Analyzer

Fault Analyzer の著作権情報および一般的な使用情報を表示する場合に使用します。[製品の著作権、ライセンスおよびバージョン情報の表示 ページ 131](#)を参照してください。

### Options->Batch Reanalysis Options

バッチ再分析用のオプションを設定する場合に使用します。[バッチ再分析オプション ページ 154](#)を参照してください。

### Options->Fault Analyzer Preferences

Fault Entry List 画面の動作に影響を与えるオプションを設定する場合に使用します。[Fault Analyzer プリファレンス ページ 117](#)を参照してください。

### Options->Interactive Reanalysis Options

対話式再分析用のオプションを設定する場合に使用します。詳しくは、[対話式再分析オプション ページ 162](#)を参照してください。

### Options->Options in Effect

現在有効なオプションと、それが指定されている場所を表示する場合に使用します。詳細については、[現在有効なオプションの表示 ページ 152](#)を参照してください。

### Services->Copy Current Display to Data Set

現在の画面の内容全体をデータ・セットにコピーする場合に使用します。[ファイルへの対話式画面のコピー ページ 130](#)を参照してください。

### Services->COBOL Explorer

これは、COBOL Explorer  
ダイアログを開始してプログラム選択のプロンプトを出すために使用されます。詳しくは、[COBOL Explorer ページ 246](#)を参照してください。

### Services->IDIS Subsystem Information

これは、対話式 IDIS サブシステム・インターフェースを起動するために使用されます。詳しくは、[対話式 IDIS サブシステム・インターフェースの使用 ページ 147](#)を参照してください。

### Services->LANGP Side File Formatting Utility

これは、LANGP  
サイド・ファイル・フォーマット・ユーティリティーを呼び出して疑似コンパイラ・リストを表示するために使用されます。詳しくは、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* を参照してください。

**Services->List User Notes**

現行の障害項目に存在するすべてのユーザー・ノートを表示するために使用します。このオプションを選択するのは、NOTELIST コマンドを発行するのと同じです。[ユーザー・ノートの作成と管理 ページ 218](#)を参照してください。

**Services->Message ID Lookup**

ユーザーが選択したメッセージや異常終了コードの説明、またはその他の情報を表示する場合に使用します。詳しくは、[ユーザー選択メッセージまたは異常終了コードの説明の表示 ページ 128](#)を参照してください。

**Services->Service Information**

インストールと保守の状況に関する情報を表示するために使用します。詳しくは、「[Fault Analyzer の保守](#)」の[ステップ 3: サービス・レベルを検証する \(オプション\) ページ 460](#)を参照してください。

**Services->Storage Map**

現在のアドレス・スペースのレイアウトを表示するために使用されます。詳しくは、[アドレス・スペース・ストレージ・マップの表示 \(STGMAP\) ページ 224](#)を参照してください。

**View->Add Blank Lines**

表示される情報を分離するために、必要に応じて空白行を使用して画面をフォーマットする場合に使用します。この値はデフォルトです。[空白行の追加または除去 ページ 127](#)を参照してください。

**View->Add Help Text**

経験の少ないユーザーを支援する説明文を画面に追加する場合に使用します。この値はデフォルトです。詳しくは、[ヘルプ・テキストの追加または除去 ページ 127](#)を参照してください。

**View->Add Pseudo Assembler Instructions**

これは、疑似アセンブラー命令を「[Compiler Listing](#)」画面に追加するために使用されます。詳しくは、[ソース・コードの表示 ページ 212](#)を参照してください。

**View->Collapse Duplicate Fault Entries**

別個の障害項目として表示される重複障害インスタンスを Fault Entry List 画面から削除する場合に使用します。[障害項目の重複ヒストリーの表示 ページ 139](#)を参照してください。

**View->Column Configuration**

Fault Entry List

画面にリストされる障害について、表示される情報の列を変更する場合に使用します。詳しくは、[Fault Entry List の列構成 ページ 71](#)を参照してください。

**View->Expand Duplicate Fault Entries**

重複障害インスタンスを別個の障害項目として Fault Entry List 画面に表示する場合に使用します。[障害項目の重複ヒストリーの表示 ページ 139](#)を参照してください。

**View->Preferred formatting Width**

優先する画面フォーマット幅を設定する場合に使用します。[優先フォーマット幅の設定 ページ 128](#)を参照してください。

**View->Refresh**

これは、選択された履歴・ファイル/ビューにある項目をすべて再読み取りするために使用されます。[障害項目情報の最新表示 ページ 98](#)を参照してください。

**View->Remove Blank Lines**

少数の行のみを表示できる画面上に最大の情報量を表示できるようにするために、画面内でできるだけ空白行を除去する際に使用します。詳しくは、[空白行の追加または除去 ページ 127](#)を参照してください。

**View->Remove Help Text**

画面から説明文を除去する際に使用します。[ヘルプ・テキストの追加または除去 ページ 127](#)を参照してください。

**View->Remove Pseudo Assembler Instructions**

これは、疑似アセンブラー命令を「Compiler Listing」画面から削除するために使用されます。この値はデフォルトです。[ソース・コードの表示 ページ 212](#)を参照してください。

## コマンド

Fault Analyzer ISPF インターフェースから、以下の基本コマンドを使用できます。



**注:** すべてのコマンドがどの画面からも実行できるわけではありません。詳細については、個々のコマンドの説明を参照してください。

## CE

これは COBOL Explorer を呼び出します。COBOL Explorer はイベント・ソース行において変数 (分岐分析に使用される) を要求します。分岐分析は、選択された変数の使用を示す手続きトレースバックです。

このコマンドは、対話式再分析時に入力します。

図 29. Syntax

```

  CE ----->
     |
     | program_name
     |
  
```

***program\_name***

関連 COBOL プログラム・イベントの名前。この名前が省略された場合は、COBOL プログラム・イベントのポップアップ・リストが表示されます。

詳しくは、[「COBOL Explorer ページ 246」](#)を参照してください。



## CICSD

CICSD コマンドを使用して、障害項目の作成時に取り込んだ CICS 3270 Screen Buffer を表示します。対話式レポートの画面で「[Last CICS 3270 Screen Buffer ページ 184](#)」リンクによって表示されるプレーン・テキストの表記とは異なり、CICSD コマンドでは最新の CICS 3270 画面のすべての色と属性 (2 バイト文字を含む) を表示します。

CICSD コマンドは、Fault Analyzer バージョン 14.1.14 以降を使用して取り込んだ障害項目にのみ有効です。

図 30. Syntax

▶▶ CICSD ◀◀

## CICSLINK

これは、「CICS® トレース・リンク分析」画面を表示します。この画面には、ネストおよびリンクされた CICS® トランザクション・プログラムの実行パスおよび階層が示されます。[内部 CICS トレース・リンク分析 \(CICSLINK\) ページ 189](#) の分析画面の例を参照してください。

CICS® システム・ダンプまたは補助トレース・データ・セットに含まれるフォーマット設定済み CICS® トレースを表示しているときにこのコマンドを実行した場合に、分析対象のタスク番号にカーソルがまだ置かれていなければ、分析対象のタスク番号を求めるプロンプトが出されます。

このコマンドは、フォーマット設定済み CICS® トレースを表示しているときにのみ使用できます。

図 31. Syntax

▶▶ CICSLINK ◀◀

詳しくは、[CICS Trace Formatting ページ 186](#)を参照してください。

## CICSSTG

CICSSTG コマンドを使用して CICS タスク・ストレージ域をフィルターおよび表示します。このコマンドは、対話式レポートからのみ使用できます。

図 32. Syntax

▶▶ CICSSTG ◀◀

詳しくは、[CICS トランジション・ストレージの表示 \(CICSSTG\) ページ 223](#)を参照してください。

## CMDS

CMDS コマンドは、ISPF 対話式再分析セッションで使用可能なコマンドのリストを表示します。「S」を使用してコマンドを選択すると、コマンドを実行できます。

図 33. Syntax

▶ CMDS ◀

## COLS

Fault Entry List Column Configuration 画面を立ち上げます。この画面では、Fault Entry List 画面に表示される情報を調整できます。

このコマンドは、Fault Entry List 画面からのみ使用できます。

図 34. Syntax

▶ COLS ◀

詳しくは、[Fault Entry List の列構成 ページ 71](#)を参照してください。

## COPY

現行の画面を、指定されたデータ・セット名にコピーします。

図 35. Syntax

▶ COPY ————— ◀  
                   └── data\_set\_name ─┘

各部の意味は以下のとおりです。

### **data\_set\_name**

画面が書き込まれる先のデータ・セット。

データ・セット用の標準の TSO 命名規則が適用されます。つまり、データ・セット名が単一引用符で囲まれていない場合、現行の TSO 接頭部が高位修飾子として使用されます。データ・セットが分割される場合、メンバー名もデータ・セット名の後に小括弧に入れて指定できます。

データ・セット名が指定されない場合、データ・セット名を入力できるポップアップ・パネルが表示されます。

データ・セットの論理レコード長が不足している場合、コピーされたデータは切り捨てられます。

COPY コマンドに関する詳細および使用例については、[ファイルへの対話式画面のコピー ページ 130](#)を参照してください。

## CUROPTS

このコマンドを使用して、現在有効なオプションとオプションが指定されている場所を表示します。このコマンドは、Fault Entry List 画面からのみ使用できます。また、メニュー・バーのプルダウンから「Options->Options In Effect」オプションを選択すると同じことを行えます。

▶▶ CUROPTS ◀◀

詳しくは、[現在有効なオプションの表示 ページ 152](#) を参照してください。

## DISASM

このコマンドは、対話式レポート内からのみ使用可能であり、ストレージの指定されたアドレスからオブジェクト・コードを逆アセンブルする場合に使用できます。

図 36. Syntax

▶▶ DISASM ◀◀

このコマンドの使用法については、[オブジェクト・コードの逆アセンブル ページ 234](#)を参照してください。

## DSECT

このコマンドは、対話式レポート内からのみ使用可能であり、ユーザー提供のアセンブラー・マクロまたは DSECT サンプル集に基づいて、ストレージ域のフォーマットを指示する場合に使用できます。

図 37. Syntax

▶▶ DSECT ◀◀

このコマンドの使用法については、[DSECT 情報を使用したストレージ域のマッピング ページ 227](#)を参照してください。

## DUPS

このコマンドは、対話式レポート内からのみ使用可能であり、現行の障害項目に対して発生した重複障害に関する詳細を表示するために使用できます。

図 38. Syntax

▶▶ DUPS ◀◀

重複障害の詳細の表示については、[障害項目の重複ヒストリーの表示 ページ 139](#)を参照してください。

## EDIT

このコマンドは、現在の画面を一時的なシーケンシャル・データ・セットにコピーし、このデータ・セットを ISPF EDIT セッションで開きます。

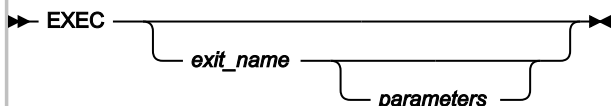
図 39. Syntax

▶▶ EDIT ◀◀

## EXEC

このコマンドは、対話式レポート内からのみ使用可能であり、フォーマット・ユーザー出口を呼び出す場合に使用できません。

図 40. Syntax



各部の意味は以下のとおりです。

### exec\_name

実行されるフォーマット・ユーザー出口の名前。

出口名を指定しない場合、画面が表示され、出口を選択できます。

### parameters

フォーマット・ユーザー出口に受け渡されるオプション・パラメーター。

複数の空白区切りパラメーターを指定した場合、これらは個別にユーザー出口に渡されます。

例えば、ユーザー出口 MYEXEC が 2 つのパラメーター「A」と「B」を受け入れる場合、出口は次のように ISPF コマンド行から呼び出されます。

```
EXEC MYEXEC A B
```

そしてパラメーターは以下を使用して出口で受け取ることができます。

```
PARM1 = ARG(1)
PARM2 = ARG(2)
```

注:

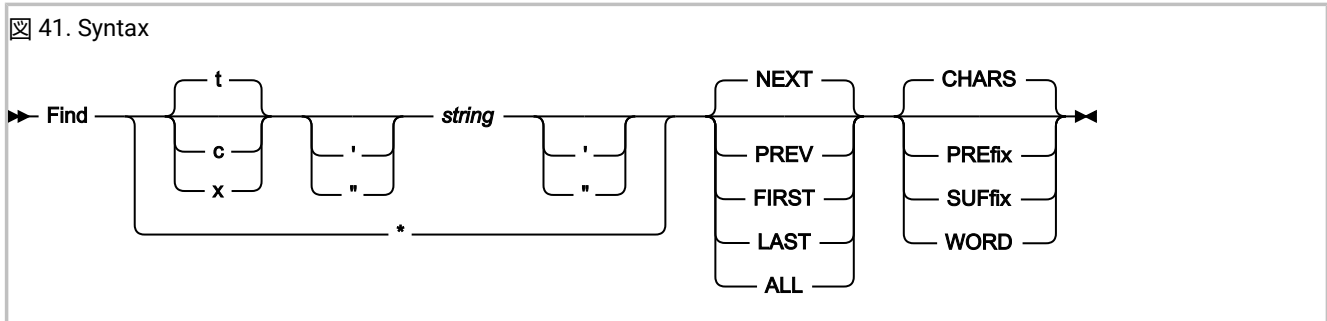
```
PARSE ARG PARM1 PARM2
```

これは期待どおりに動作しません。

独自のフォーマット・ユーザー出口の作成方法と、それらの出口をご使用の環境で使用可能にする方法については、[ユーザー固有レポートのフォーマット ページ 237](#) を参照してください。

## FIND

現行画面内のテキスト・ストリングを検索する場合に使用します。



各部の意味は以下のとおりです。

**t**

テキスト・ストリングが検索対象であることを示します。Dump Storage 画面を除くすべての画面では、テキスト・ストリングは大文字小文字が区別されません。Dump Storage 画面では、すべてのテキスト・ストリングについて大文字小文字が区別されます。

**c**

文字ストリングが検索対象であることを示します。文字ストリングには、大文字小文字の区別があります。

**x**

16 進数が検索対象であることを示します。偶数桁で指定する必要があります。

**string**

検索対象の文字ストリング。このストリングに、FIND コマンド・キーワードと誤解される可能性がある空白、引用符、またはトークンが含まれている場合 (例えば、ストリング NEXT を検索する場合)、このストリングは、単一引用符か、二重引用符のどちらかで囲む必要があります。終わりの引用符は、開始の引用符と同じ種類 (単一または二重) である必要があります。それ以外の引用符はすべて、検索ストリングの一部と見なされます。

**\***

以前の検索対象と同じストリングを使用することを指定します。

**NEXT**

現在のカーソル位置の後の最初の位置から検索が開始し、前方に検索を行って、ストリングの次のオカレンスを検出することを指定します。NEXT がデフォルトです。

**PREV**

現在のカーソル位置の前の最初の位置から検索が開始し、後方に検索を行って、ストリングの直前のオカレンスを検出することを指定します。

**FIRST**

画面の一番上から検索が開始し、前方に検索を行って、ストリングの最初のオカレンスを検出することを指定します。

**LAST**

画面の一番下から検索が開始し、後方に検索を行って、ストリングの最後のオカレンスを検出することを指定します。

**ALL**

画面の一番上から検索が開始し、前方に検索を行って、ストリングのすべてのオカレンスを検出することを指定します。画面の右上隅にあるメッセージが、検出されたオカレンス数を示します。

**CHARS**

検索対象の文字のシーケンスのオカレンスはすべて一致と見なされることを示します。この値はデフォルトです。

**PREfix**

一致と見なされるために、検索対象の文字のシーケンスの前に、1 つ以上の空白または属性文字が先行する必要があることを指定します。PRE または PREFIX を指定できます。

**SUFfix**

一致と見なされるために、検索対象の文字のシーケンスの後に、1 つ以上の空白または属性文字が続く必要があることを指定します。SUF または SUFFIX を指定できます。

**WORD**

一致と見なされるために、検索対象の文字のシーケンスの前後に、1 つ以上の空白または属性文字を指定する必要があることを指定します。

**画面のタイプによる FIND コマンドの差異**

「Dump Storage」画面から発行された FIND コマンドは、他のすべての画面から発行された FIND コマンドとは異なる動作をします。（「Dump Storage」画面は、SHOW コマンドを使用するか、アドレスのポイント・アンド・シュート・フィールドにカーソルを置いて Enter キーを押すと表示されます。）

**表 6. 画面のタイプによる FIND コマンドの差異**

Dump Storage 画面での FIND コマンド	他のすべての画面での FIND コマンド
T'テキスト' または C'テキスト' のどちらを使用するかにかかわらず、またはいずれも使用しない場合も、文字ストリングはすべて大/小文字が区別されます。	文字ストリングの大文字小文字を区別するかどうかは、T'テキスト' (デフォルト) または C'テキスト' のいずれを使用するかにより決定されます。
ターゲット・ストリングは、ミニダンプと関連 MVS™ ダンプのみを対象に検索され、ストレージの説明、見出し、および同様にフォーマット設定された文字ベースのテキストは検索対象に含まれません。一方ミニダンプと MVS™ ダンプは、現在表示されているアドレス範囲のみでなく、全体が検索対象となります。	ターゲット・ストリングは、16 進ダンプ・フォーマットなどのフォーマット設定された文字ベースのテキストのみで検索されます。ミニダンプまたは MVS™ ダンプは検索されません。

表 6. 画面のタイプによる FIND コマンドの差異 (続く)

Dump Storage 画面での FIND コマンド	他のすべての画面での FIND コマンド
ミニダンプで 16 進値を検索するには、X'テキスト'フォーマットを使用します。	16 進フォーマットのストレージを検索するには、文字ストリング・フォーマットを使用します。
FIND コマンドのターゲットは、定様式表示で複数行に分割されている場合でも検出できます。	FIND コマンドのターゲットは、定様式表示で複数の行に分割されている場合は検出できません。

## IDISINFO

IDISINFO コマンドは、対話式 IDIS サブシステム・インターフェースを起動するために使用されます。

図 42. Syntax

▶▶ IDISINFO ◀◀

詳しくは、[対話式 IDIS サブシステム・インターフェースの使用 ページ 147](#)を参照してください。

## INFO

このコマンドは、対話式レポート内からのみ使用可能であり、現行の障害項目に関する情報を表示するために使用できません。

INFO コマンドを発行することは、対話式再分析レポートのアクション・バーの「File」プルダウン・メニュー ([アクション・バーのプルダウン・メニュー ページ 100](#)を参照) から「Fault Entry Information」を選択すること、または Fault Entry List 画面 ([特定の障害へのアクションの適用 ページ 87](#)を参照) から 1 つの障害項目に対して「？」行コマンドを入力することと同じです。いずれの場合にも、[図 72 : Fault Entry Information 画面の例 ページ 135](#)に示される、Fault Entry Information 画面が表示されます。

図 43. Syntax

▶▶ INFO ◀◀

## JCL

JclCapture オプションが、リアルタイム処理中に異常終了したジョブの JCL を正常にキャプチャーした後、JCL コマンドを使用して、ISPF EDIT パネルに JCL を表示することができます。[JclCapture ページ 607](#)を参照してください。

図 44. Syntax

▶▶ JCL ◀◀

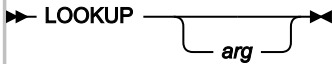
この JCL コマンドは、対話式レポートからのみ使用できます。

JCL コマンドを使用することは、対話式再分析レポートの「Abend Job Information」セクションから「Abend Job JCL」ポイント・アンド・シュート・フィールドを選択することと同等です。

## LOOKUP

指定されたメッセージ ID や異常終了コードの説明、またはその他の情報を表示します。

図 45. Syntax



各部の意味は以下のとおりです。

### arg

検索するメッセージ ID、異常終了コード、またはその他の情報項目。

arg が指定されていない場合はポップアップ・パネルが表示され、必要な情報を選択または検索できます。

アスタリスクをワイルドカード文字として使用して、任意の数の文字を表すことができます。アスタリスクの指定が特にならない場合、arg の先頭と終わりの両方に暗黙的アスタリスク指定があると見なされます。

%記号を使用すると、単一文字を表すことができます。

以下に、ワイルドカード文字の使用法を示した検索引数の例を示します。

### XYZ

\*XYZ\* に一致するものを暗黙的に検索します。これは、XYZ の前後に 0 文字以上の文字があることを意味します (例えば、XYZ、AXYZ、XYZB、AXYZBBB)。

### A\*

A で始まり、その後に 0 文字以上が続く、すべてのものが一致します (例えば、A、AB、ABA000I)。

### A%

A で始まり、その後に 1 文字のみが続く、すべてのものが一致します (例えば、AA、AB、AC)。

### ABA%%%I

ABA で始まり、その後に 4 文字続き、I で終わる、すべてのものが一致します (例えば、ABA0000I)。

検索結果に複数の情報項目が含まれる場合は、リストが表示され、そこから情報を選択できます。

Fault Analyzer から提供される情報、または IDIHUSRM データ・セットのメンバー IDI.SIDISAM1 で指定された情報のみが入力可能です。

情報(メッセージや異常終了コードの説明など)は以下の場所で以下の順序で検索されます。

1. IDIHUSRM データ・セットのメンバー IDI.SIDISAM1 内のユーザー定義情報。詳しくは、[アプリケーション固有の説明の指定 ページ 455](#)を参照してください。
2. IBM® データ・セット内の IDIDOC 提供の情報。



3. IBM データ・セット内の IDIVSxxx 提供の情報 (xxx は Language オプションに有効な 3 文字の言語 ID)。
4. メッセージまたは異常終了コードの説明の場合は、Message and Abend Code Explanationのユーザー出口で、代わりの説明を指定できます。詳しくは、[メッセージおよび異常終了コード説明ユーザー出口 ページ 485](#)を参照してください。

LOOKUP コマンドと、その LOOKC バリエントを Fault Analyzer ISPF インターフェースの外部から発行することもできますが、その場合、ISPF の下で発行する必要があり、サンプルの Fault Analyzer ISPF コマンド・テーブルが使用可能にされている必要があります。詳しくは、[ISPF 環境の変更 ページ 328](#)を参照してください。

LOOKUP コマンドに関する詳細および使用例については、[ユーザー選択メッセージまたは異常終了コードの説明の表示 ページ 128](#)を参照してください。

## MATCH

MATCH コマンドについては、[障害を突き合わせて選択する追加の方法 ページ 83](#)を参照してください。

## NEXT

このコマンドは、対話式レポート内からのみ使用可能であり、このコマンドの動作は、コマンドが起動される元の画面のタイプによって異なります。

### イベント詳細

次のイベントを選択するには、このコマンドを使用します。例えば、現在イベント番号 2 が表示されている場合に、このコマンドを入力すると、使用可能であれば、イベント番号 3 が表示されます。

### ダンプ・ストレージ (Dump Storage)

PREV コマンドを入力する前に表示されるストレージ・アドレスが、再度表示されます。

このコマンドは通常、PF11 ファンクション・キーに割り当てられます。

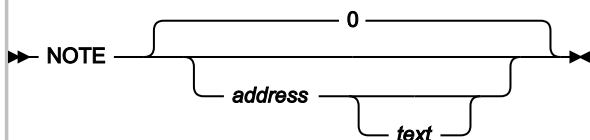
図 46. Syntax

▶ NEXT ◀

## NOTE

このコマンドを使用して、編集セッションでユーザー・ノートを作成または既存のユーザー・ノートを編集できます。これは、対話式レポートからのみ使用できます。

図 47. Syntax



アドレスのユーザー・ノートが既に存在する場合、そのユーザー・ノートが開かれます。それ以外の場合は、新しいユーザー・ノートが作成されます。

新しいユーザー・ノートを作成する場合は、ノート・アドレスが「上記と「同様」」で制限されているために表示されないストレージ・セクション内にある場合など、ダンプ・ストレージ画面のストレージ領域を上書きする代わりにこのコマンドとしてこのコマンドを使用します。

パラメーターを使用せずに(デフォルトのアドレスは 0) NOTE コマンドを使用すると、問題の説明やチェック項目の伝達などの障害エントリーの一般的なノートを管理できます。

詳しくは、[ユーザー・ノートの作成と管理 ページ 218](#)を参照してください。

## NOTELIST

このコマンドは、対話式レポート内からのみ使用可能であり、現行の障害項目が存在するすべてのユーザー・ノートを表示する場合に使用できます。

図 48. Syntax



詳しくは、[「ユーザー・ノートの作成と管理 ページ 218」](#)を参照してください。

## PREV

このコマンドは、対話式レポート内からのみ使用可能であり、このコマンドの動作は、次のタイプの情報のどちらが表示されるかによって異なります。

### Event details

直前のイベントを選択するには、このコマンドを使用します。例えば、現在イベント番号 2 が表示されている場合に、このコマンドを入力すると、イベント番号 1 が表示されます。

### Dump Storage Display

以前に表示されたストレージ・アドレスがある場合は、そのアドレスが表示されます。

このコマンドは通常、PF10 ファンクション・キーに割り当てられます。

図 49. Syntax



## QUIT

このコマンドの動作は、このコマンドが発行された場所によって異なります。

- 対話式再分析レポートの中で発行された場合、Fault Entry List 画面に戻ります。

この動作は、File プルダウン・メニューから「Exit Interactive Reanalysis」を選択することと同等です。

- Fault Entry List 画面で発行された場合、ISPF インターフェースを終了します。

この動作は、File プルダウン・メニューから「Exit Fault Analyzer」を選択することと同等です。

図 50. Syntax

▶▶ QUIT ◀◀

## REFRESH

このコマンドは、Fault Entry List 画面からのみ使用できます。

再読み取りが行われる現行の履歴・ファイルまたはファイルが、そのファイルまたはビューの最初の選択以降、または REFRESH コマンドが最後に実行されて以降に作成されたすべての更新を、画面に組み込むようにします。

アクティブである可能性のある MATCH コマンドがリセットされます。

REFRESH コマンドを実行すると、View メニュー・バーのプルダウン・メニューから Refresh オプションを選択したのと同じ機能を実行します。

図 51. Syntax

▶▶ REFRESH ◀◀

## RESET

このコマンドは、Fault Entry List Column Configuration 画面からのみ使用できます。

Fault Entry List の列構成が、有効な FAISPFopts(HistCols(...)) オプションによって定義される構成に変更されるようにします。

図 52. Syntax

▶▶ RESET ◀◀

## RPTFIND

前回の FIND コマンドに対して入力された検索指数の次のオカレンスを検出します。このコマンドは、デフォルトで PF5 ファンクション・キーにマップされます。

図 53. Syntax

▶▶ RPTFIND ◀◀  
RF

## RUNCHAIN

このコマンドは、対話式レポート内からのみ使用可能であり、チェーン・データ域を表示する場合に使用できます。

A rectangular box with a thin border containing the text "54. Syntax" and "RUNCHAIN" with arrows on either side.
 

#### 54. Syntax

▶▶ **RUNCHAIN** ◀◀

このコマンドの使用法については、[チェーン・データ域の表示 ページ 231](#)を参照してください。

## SHOW

このコマンドは、対話式レポート内からのみ使用可能であり、保管場所を表示する場合に使用できます。

 A diagram showing the syntax of the SHOW command. It starts with "SHOW" followed by a bracketed "address" field. Above the address field is a "0" with a bracket. To the right is a bracketed "offset" field, with a "+" sign above it and a "-" sign below it. A large arrow above the offset field points to the left, indicating that the offset is subtracted from the address.
 

#### 55. Syntax

例えば、アドレス 007F2300 のストレージを表示するには、次のように入力します。

```
SHOW 7F2300
```

address パラメーターは 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。

アドレスを指定せずに SHOW コマンドを発行した場合、0 または SHOW コマンドが発行された CICS® システム・ダンプ分析のデータ域の表示に関連したストレージのアドレス、または SHOW に最後に選択されたアドレスのいずれかが、デフォルトで使用されます。

オフセットは SHOW コマンドを使用して指定されます。例:

```
SHOW 142A0 + 1C0 + 1B - 8
```

オフセットはすべて 16 進数で指定されます。SHOW と基本アドレスは 1 つ以上の空白で区切る必要があります。これに対し、各オフセットを区切る空白は任意で挿入します。

## SHOWFREE

このコマンドを使用すれば、TSO 領域内の使用可能なストレージ量を調べることができます。

 A rectangular box with a thin border containing the text "56. Syntax" and "SHOWFREE" with arrows on either side.
 

#### 56. Syntax

▶▶ **SHOWFREE** ◀◀

SHOWFREE コマンドの出力例を以下に示します。

```
Largest Contiguous Virtual Storage Block Available:
  Above the Line. . : 12.30 MB
  Below the Line. . :  3.66 MB
Total Virtual Storage Available:
  Above the Line. . : 24.86 MB
  Below the Line. . :  3.70 MB
***
```

## SIT

CICS 障害項目の対話式再分析中に CICS® システム初期化パラメーターを表示します。

図 57. Syntax

▶▶ SIT ◀◀

## STCK

このコマンドは、対話式レポート内からのみ使用可能であり、バイナリーの STORE\_CLOCK 値を目視で確認できる 日付と時刻のフォーマットに変換する場合に使用できます。

図 58. Syntax

▶▶ STCK ◀◀

このコマンドの使用法については、[STORE CLOCK 値の変換 ページ 236](#)を参照してください。

## STGMAP

このコマンドは、対話式レポート内からのみ使用可能であり、Storage Map 画面を表示する場合に使用できます。

図 59. Syntax

▶▶ STGMAP ◀◀

このコマンドの使用法については、[アドレス・スペース・ストレージ・マップの表示 \(STGMAP\) ページ 224](#)を参照してください。

## VIEWS

このコマンドは、Fault Entry List 画面からのみ使用可能であり、現在使用可能なすべてのビューのリストを表示する場合に使用できます。VIEWS コマンドを発行すると、File アクション・バーのプルダウン・メニューから List Views オプションが選択された場合と同じ機能が実行されます ([ビューのリストからビューを選択する ページ 70](#)を参照してください)。

図 60. Syntax

▶▶ VIEWS ◀◀

## Fault Analyzer プリファレンス

Fault Analyzer ISPF インターフェース全般に適用される環境設定を指定するには、「**Fault Entry List**」画面の「**Options**」メニューから **Fault Analyzer Preferences** を選択します。これで、[図 61 : Fault Analyzer プリファレンス画面の例 ページ 118](#)に示されている画面が開きます。(メニュー・オプションの一般情報については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください。)

図 61. Fault Analyzer プリファレンス画面の例

```

File View Services Help
-----
Fault Analyzer Preferences                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Press PF3 to save options or PF12 to cancel.

Confirm Fault Entry Deletion: Y (Y/N)
Default Lock Flag Value . . . : /
Locked Fault Entry
Highlighting. . . . . : Y (Y/N)
  Color . . . . . : R (B=Blue, G=Green, P=Pink, R=Red, T=Turquoise)
  Column. . . . . : L (F=Fault_ID, L=Lock, B=Both)
Expanded Duplicate Fault
Entry Highlighting. . . . . : Y (Y/N)
  Color . . . . . : P (B=Blue, G=Green, P=Pink, R=Red, T=Turquoise)
  Column. . . . . : B (F=Fault_ID, D=Dups, B=Both)

*** Bottom of data.

```

設定できるオプションは、以下のとおりです。

#### Confirm Fault Entry Deletion

**N**

障害項目の削除を試行する際、「**Confirm Fault Entry Deletion**」画面が毎回表示されない。

**Y**

障害項目の削除を試行する際、「**Confirm Fault Entry Deletion**」画面が毎回表示される。

このオプションは、確認画面から変更することもできます。「**Confirm Fault Entry Deletion**」画面の詳細については、[ヒストリー・ファイル項目の削除 ページ 131](#)を参照してください。

#### デフォルト・ロック・フラグ値

[L] ライン・コマンドの使用時にロック・フラグ値として設定する1つまたは2つの文字。空白の場合は、[/] が使用されます。

ロック・フラグの最終値は、IDIXLOCK ロック・フラグ制御出口によって行われた変更に応じて変化します。[障害項目ロック・フラグ値の制御 ページ 133](#)を参照してください。

#### ロックされた障害項目の強調表示

Fault Entry List 画面のロックされた障害項目の強調表示を制御する単一の文字。空白の場合は、「N」が使用されます。

**N**

ロックされた障害項目は強調表示されません。

**Y**

ロックされた障害項目が強調表示されます。y を指定すると、追加のオプションが表示されます:

**色**

ロックされた障害項目の強調表示の色を示す単一の文字。色を指定しない場合は、**R** を使用します。

**B**

ブルー

**G**

グリーン

**P**

ピンク

**R**

赤 (デフォルト)

**T**

ターコイズ

**列**

Fault Entry List 画面で強調表示する列を示す単一の文字。空白の場合は、**L** を使用します。

**F**

Fault\_ID 列

**L**

Lock 列 (デフォルト)

**B**

Fault\_ID 列と Lock 列の両方

**展開した重複障害項目の強調表示**

Fault Entry List 画面で展開された重複障害項目の強調表示を制御する単一の文字。空白の場合は、**Y** を使用します。

**N**

展開された重複障害項目が強調表示されない。

**Y**

展開した重複障害項目が強調表示される。**Y** を指定すると、追加のオプションが表示されます:

**色**

Fault Entry List

画面で展開された重複障害項目の色の強調表示を制御する単一の文字。空白の場合は、**P** を使用します。

- B**  
ブルー
- G**  
グリーン
- P**  
ピンク (デフォルト)
- R**  
レッド
- T**  
ターコイズ

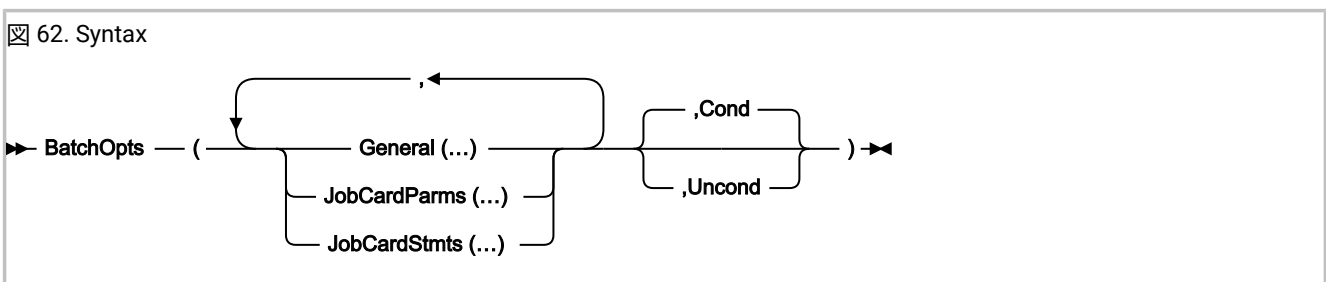
**列**

Fault Entry List 画面で強調表示する列を示す単一の文字。空白の場合は、**B** を使用します。

- F**  
Fault\_ID 列
- D**  
Dups 列
- B**  
Fault\_ID 列と Dups 列の両方 (デフォルト)

**BatchOpts**

BATCHOPTS サブオプションを使用して、Batch Reanalysis Options 画面でデフォルト設定の代替設定を指定します。[バッチ再分析オプション ページ 154](#) を参照してください。



**BatchOpts の COND および UNCOND サブオプション**

BATCHOPTS サブオプションの各レベルで、COND および UNCOND サブオプションを指定できます:



- 最高レベル (BATCHOPTS サブオプション)
- 中間レベル (GENERAL、JOB CARDPARMS および JOB CARDSTMTS サブオプション)
- 最低レベル (JCLEDIT や MSGCLASS などの個々の設定)

COND (条件付き) では、ユーザーが別の設定をまだ指定していない場合、選択済みの値が画面上でデフォルト設定として使用されます。そのため、はじめてのユーザーにはこちらが適用されます。また、ユーザーが画面上で設定をクリアした場合にもこちらが毎回適用されます。

UNCOND (無条件) では、以前の設定に関係なく、任意の Fault Analyzer ISPF インターフェース・セッションの開始時に、選択済みの値が画面の現在の設定として使用されます。セッション中に、設定をその他の有効な値に変更することができます。この値は、次のセッションまで有効になります。

より高いレベルで UNCOND が指定されていない限り、すべてのレベルで COND がデフォルトになります。つまり、上位レベルで指定した内容は、デフォルトでそれぞれの下位レベルに伝搬されます。

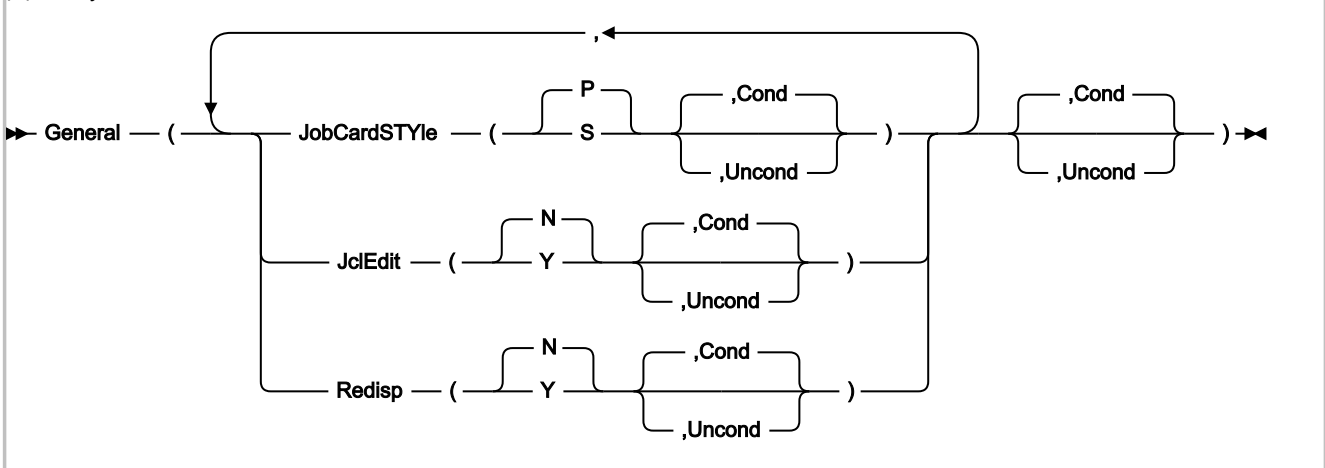
次の例を考えてみます。

```
FAISPFOPPTS (BATCHOPTS (GENERAL (JOB CARDSTYLE (P), JCLEDIT (Y), JOB CARDSTMTS (<stmt1>, <stmt2>),
JOB CARDPARMS (MSGCLASS (A), COND), UNCOND))
```

- JOB CARDPARMS (MSGCLASS (A)) は条件付きです。
- GENERAL (JOB CARDSTYLE (P)、JCLEDIT (Y)、JOB CARDSTMTS (<stmt1>, <stmt2>) は無条件です。

### BatchOpts の一般サブオプション

図 63. Syntax



GENERAL サブオプションは、Batch Reanalysis Options 画面の「General Options」セクションに適用されます。

**JOB CARDSTYLE(P |S [COND |UNCOND]))**

**JCSTY(P |S [C |U])**

Batch Reanalysis Options 画面の「Job card style」フィールドに次の値を指定します。

**P**

パラメーター

**S**

ステートメント

指定していない場合、デフォルトはPです。

**JCLEEDIT(Y IN [, COND |UNCOND]))**

**JE(Y | N [, C | U])**

Batch Reanalysis Options 画面の「**Display panel to edit generated JCL**」フィールドに次の値を指定します。

**Y**

あり

**N**

なし

指定していない場合、デフォルトはNです。

**REDISP(Y IN [, COND |UNCOND]))**

**R(Y | N [, C | U])**

Batch Reanalysis Options 画面の「**Redisplay this panel before each reanalysis**」フィールドに次の値を指定します。

**Y**

あり

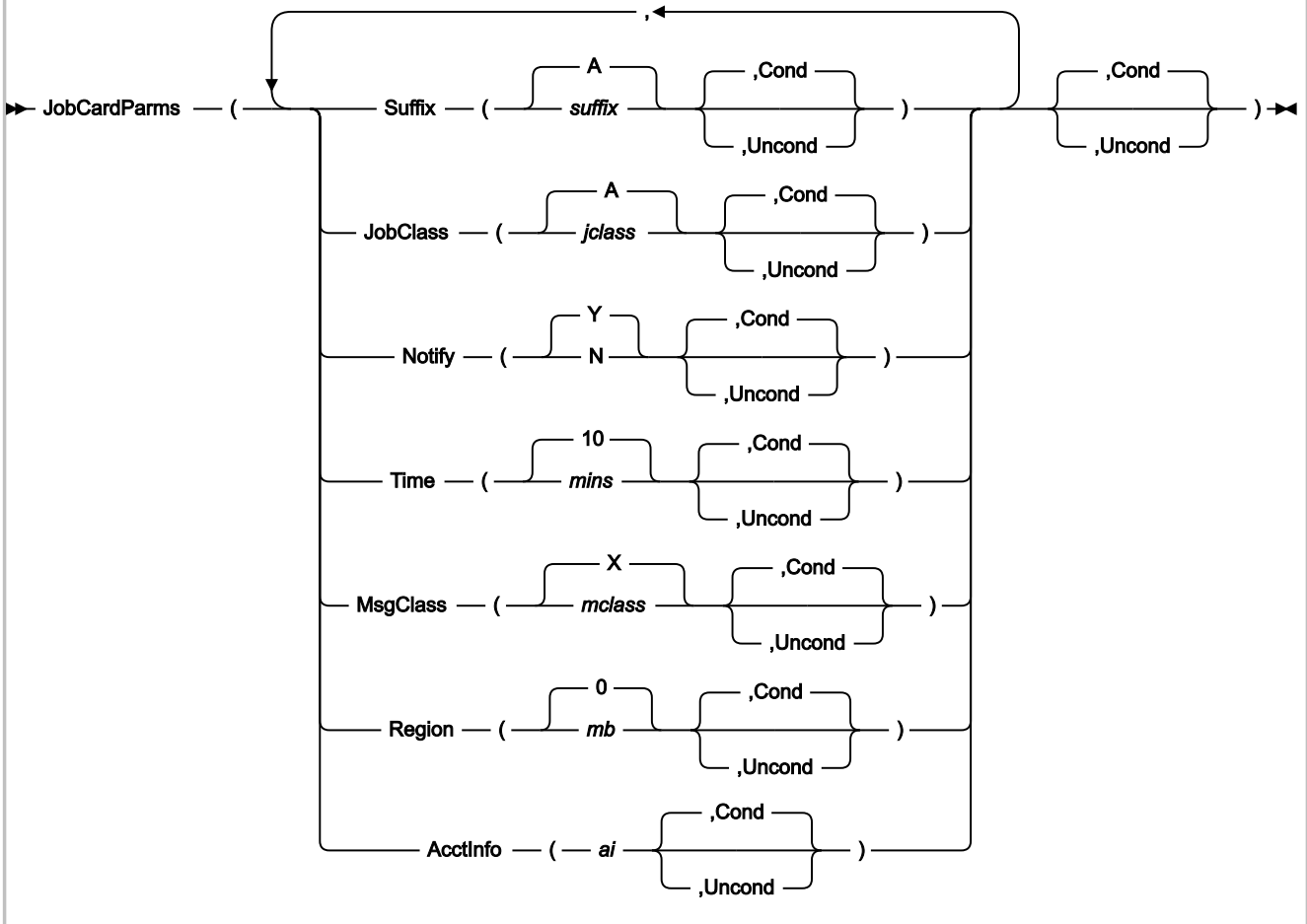
**N**

なし

指定していない場合、デフォルトはNです。

## BatchOpts JobCardParms サブオプション

図 64. Syntax



JOBCLASDPARMS サブオプションは、Batch Reanalysis Options 画面の「**Job Card Parameters**」セクションに適用されません。

**SUFFIX( *suffix* [, COND | UNCOND] )**

**S( *suffix* [, C | U] )**

Batch Reanalysis Options 画面の「**Job name suffix**」フィールドの値を指定します:

***suffix***

A ~ Z、0 ~ 9、@、# または \$

指定していない場合、デフォルトは A です。

**JOBCLASS( *jclass* [, COND | UNCOND] )**

**JC( *jclass* [, C | U] )**

Batch Reanalysis Options 画面の「**Job class**」フィールドに次の値を指定します。

***jclass***

A ~ Z または 0 ~ 9

指定していない場合、デフォルトは A です。

**NOTIFY(Y | N [, COND | UNCOND]))**

**N(Y | N [, C | U])**

Batch Reanalysis Options 画面の「**Job notify**」フィールドに次の値を指定します。

**Y**

あり

**N**

なし

指定していない場合、デフォルトは Y です。

**TIME( mins [, COND | UNCOND]))**

**T( mins [, C | U])**

Batch Reanalysis Options 画面の「**Job time minutes**」フィールドに次の値を指定します。

**mins**

0 から 99。

指定していない場合、デフォルトは 10 です。

**MSGCLASS( mclass [, COND | UNCOND]))**

**MC( mclass [, C | U])**

Batch Reanalysis Options 画面の「**Message class**」フィールドに次の値を指定します。

**mclass**

A ~ Z または 0 ~ 9

指定していない場合、デフォルトは X です。

**REGION( mb [, COND | UNCOND]))**

**R( mb [, C | U])**

Batch Reanalysis Options 画面の「**Region megabytes**」フィールドに次の値を指定します。

**mb**

0 から 2047。

指定していない場合、デフォルトは 0 です。

**ACCTINFO( ai [, COND |UNCOND]))**

**AI( ai [, C |U])**

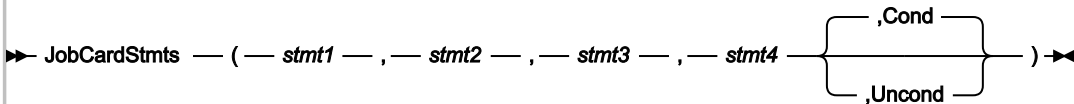
Batch Reanalysis Options 画面の「Accounting info」フィールドに次の値を指定します。

*ai*

最大 50 文字。

### BatchOpts JobCardStmts サブオプション

図 65. Syntax



最大 4 つのコンマ区切り JCL ステートメントを指定できます。

- ステートメントが指定されていない場合 (例えば、3 つのステートメントのみが必要な場合) でも、省いたステートメントの区切りコンマを指定する必要があります。
- ステートメントに空白またはコンマが含まれている場合は、一重引用符または二重引用符で囲む必要があります。ステートメントに同じタイプの引用符が既に含まれている場合は、その引用符を 2 つ重ねます。

例えば、次のような JCL ステートメントを作成するには:

```
//MYJOB JOB (123),'X Y Z',
```

以下の項目を指定できます:

```
'//MYJOB JOB (123),"X Y Z",'
```

```
"//MYJOB JOB (123),'X Y Z',"
```

```
'//MYJOB JOB (123),'X Y Z'','
```

最後の例では、2 つの一重引用符が XYZ を囲んでいます。

### 保管レポートの表示

障害に関連する保管済みレポートを表示するには、ヒストリー・ファイル画面で項目に対して **V** (または **S**) を入力します。

図 66 : Saved Report 画面の例 ページ 126 Saved Report 画面の例を示します。

図 66. Saved Report 画面の例

```

File View Services Help
-----
Saved Report                                     Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
- Expand all / + Collapse all
*****
* IBM Fault Analyzer for z/OS V14R1M8 (PH13453 2019/09/03)
*
* Copyright IBM Corp. 2000, 2017. All rights reserved.
* Copyright HCL Technologies Ltd 2017, 2019. All rights reserved.
*****

JOBNAME: IDIVPPL2  SYSTEM ABEND: 0C9              FAE1      2019/06/22  09:28:

+ <H1> S Y N O P S I S
+ <H1> E V E N T   S U M M A R Y
+ <H1> E V E N T   D E T A I L S
+ <H2> EVENT 1 OF 5: CALL (DSA ADDRESS 00034018)
+ <H2> EVENT 2 OF 5: CALL (DSA ADDRESS 000340E0)
+ <H2> EVENT 3 OF 5: CALL (DSA ADDRESS 000341D0)
+ <H2> EVENT 4 OF 5: CALL (DSA ADDRESS 00034390)
+ <H3> Associated Storage Areas
  F1=Help      F3=Exit      F5=RptFind    F7=Up          F8=Down      F10=Left
  F11=Right

```

レポート全体を表示するには、スクロール・キー (PF7/PF8/PF10/PF11) を使用してください。

ヒストリー・ファイルから表示されたレポートは、通常 JES スプールのジョブ出力に挿入されたリアルタイム・レポートと同じです。

しかし、レポートが最初に作成されたときに障害項目に書き込まれなかった場合、「V」または「S」行コマンドでレポートを初めて表示しようとしたときに可能であれば常に、疑似バッチ再分析レポートが追加されます。ただし、ヒストリー・ファイルへのアクセス権限がユーザーに必要です。リアルタイム・レポートのない障害項目は、DeferredReport オプションを有効にして作成された障害項目か、リカバリー障害記録障害項目です。

表示できるレポートはリアルタイム・レポートの場合か、または後の段階で作成して保管されたバッチ再分析レポートの場合があるため、障害項目に含まれているレポートを指すのに共通の用語「保管レポート」を使用します。

保管レポートを作成する場合、対話式再分析オプションではなく、バッチ再分析オプションを使用することに注意してください。

保管レポートを含まない障害項目の保管レポートを表示しようとして、保管レポートを作成できない場合、障害項目にミニダンプが含まれているか関連 MVS™ ダンプが存在すると、代わりに「I」行コマンドが使用されたかのように、対話式再分析が自動的に実行されます。このプロセスは、CICS® システム・ダンプ障害項目、またはユーザーに更新アクセス権のないヒストリー・ファイル内に保管レポートがない障害項目に当てはまります。

ナビゲーションを容易にするために、レポートの個々のセクションは、各レポートの見出しの前の + または - 記号が表示されたポイント・アンド・シュート・フィールドにカーソルを置き、Enter キーを押すことで、縮小または拡張できます。

- -記号が表示されている場合、セクションは現在展開されています。-記号にカーソルを置いて Enter キーを押すと、セクションが省略されます。
- +記号が表示されている場合、セクションは現在省略されています。+記号にカーソルを置いて Enter キーを押すと、セクションが展開されます。

縮小されたレポートのセクションは、見出し行のみ表示可能です。

画面の先頭は、レポートのセクションすべてを一括して拡張または縮小可能にする、2つの +/- ポイント・アンド・シュート・フィールドです。これらのいずれかが選択されると必ず、現行設定がユーザーの ISPF プロファイルに保管され、後続の Saved Report 画面での初期設定として使用されます。

## ブランク行の追加または除去

一度にごく少数の行しか表示できない画面では、Fault Analyzer 画面にできるだけブランク行を挿入しないようにすると便利です。情報は「圧縮」され、垂直方向にスクロールする必要性が少なくなります。当然、情報の読みやすさは、やや損なわれます。

View メニューの Add Blank Lines オプションと Remove Blank Lines オプションは、それぞれブランク行の挿入または除去を制御します。これらのメニュー・バーのプルダウン・メニュー・オプションは、大部分の Fault Analyzer 画面で使用可能です。(Fault Analyzer アクション・バーのプルダウン・メニュー全般については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。

デフォルトの設定では、ブランク行を追加します。

View メニューでは、現在アクティブなオプションを逆にすることしかできません。つまり、Add Blank Lines が有効である場合は、Remove Blank Lines のみが選択可能であり、逆の場合も同じです。

## ヘルプ・テキストの追加または除去

一部の画面では、経験の少ないユーザーへの支援が提供されます。例えば、選択可能なオプションの説明や、情報が検索された方法の詳述です。

View メニューの Add Help Text オプションと Remove Help Text オプションは、このヘルプ情報の組み込みまたは除外をそれぞれ制御します。これらのアクション・バーのプルダウン・メニュー・オプションは、大部分の Fault Analyzer 画面で使用可能です (Fault Analyzer アクション・バーのプルダウン・メニュー全般については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。

画面のヘルプ・テキストは、中括弧 ({} ) で囲まれます。

デフォルトの設定では、ヘルプ・テキストを追加します。

View メニューでは、現在アクティブなオプションを逆にすることしかできません。つまり、Add Help Text が有効である場合は Remove Help Text のみが選択可能であり、その逆も同様です。

## 優先フォーマット幅の設定

Fault Analyzer ISPF インターフェースにより、ユーザーは優先フォーマット幅を選択できます。この幅は、Fault Analyzer が情報をフォーマットして表示する場合に可能な限り使用する幅です。

優先フォーマット幅は、Fault Analyzer 画面の実際の幅と同じではありません。表示される一部の情報 (例えば、テキストの段落) の幅のみに影響を与えます。これ以外の情報は、設計により固定されているため、このフォーマット幅の影響を受けません。

View メニューの Preferred Formatting Width オプションを選択すると、Preferred Formatting Width 画面が表示されます (このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#) を参照してください)。

図 67. Preferred Formatting Width 画面の例

```

File Options View Services Help
----- Preferred Formatting Width Specification -----
Specify the display formatting width to be used by Fault Analyzer whenever
possible and press Enter.

Preferred Formatting Width 80_ (Minimum 80)

F1=Help      F3=Exit      F12=Cancel

Fault ID Job/Tran User ID Sys/Job Abend Date Time
F00323 IDIVPCOB IBMUSER MVS2 S0C7 2019/12/21 13:02:25
F00445 ALLANT01 JACKIED MVS8 S0C7 2019/12/19 03:29:57
F00444 ALLANT01 JACKIED MVS8 S0C7 2019/11/28 20:25:30
F00442 ALLANT01 ALLANT MVS8 S0C7 2019/09/10 22:20:10
F00349 CS05 CICSUSER CSCB0050 ASRA 2019/08/23 07:47:23
F00348 CS04 CICSUSER CSCB0040 ASRA 2019/08/23 07:46:36
F00345 CS01 CICSUSER CSCB0010 AEIL 2019/08/23 07:43:35
F00050 PSTRANDR PSTRAND STPLEX4B S0C4 2019/08/02 17:03:18
F00035 CICS53 n/a MVS2 n/a 2019/04/05 14:49:11
F00034 CICS53 n/a MVS2 S08E 2019/03/22 13:12:23
F1=Help      F3=Exit      F4=MatchCSR F5=RptFind F6=Actions F7=Up
F8=Down      F10=Left     F11=Right   F12=MatchALL

```

受け入れられる最小幅は 80 文字であり、デフォルト値と同じです。999 文字を指定できますが、最大幅は、常に実際の画面幅により制限されます。

適当なフォーマット幅を入力した後、Enter キーを押すと、指定したフォーマット幅を使用する直前の画面に戻ります。

フォーマット幅が画面の物理的な幅を超える場合、スクロール・コマンド LEFT (PF10) および RIGHT (PF11) を使用すると、画面全体を表示できます。

フォーマット幅を変更せずに直前の画面に戻るには、EXIT (PF3) または CANCEL (PF12) コマンドを入力してください。

## ユーザー選択メッセージまたは異常終了コードの説明の表示

LOOKUP コマンドは、対話式障害ヒストリー・ファイル画面または対話式障害再分析レポートから発行できます。このコマンドを使用して、ユーザーが選択したメッセージ ID または異常終了コードの説明を表示できます。また、DB2® SQLCODE や VSAM フィードバック・コードなどの、その他の情報を表示する場合も使用できます。(コマンド構文については、[LOOKUP ページ 112](#)を参照してください。)



例えば、次のコマンドを入力するとします。

```
LOOKUP IEC141I
```

図 68 : Message ID Look-Up 画面の例 ページ 129 のように、メッセージの説明を記載するパネルが表示されます。

図 68. Message ID Look-Up 画面の例

```

Message IEC141I Explanation                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR_

IEC141I
IEC141I 013-rc,mod,jjj,sss, ddname[-#] [,dev,volser, dsname]
Explanation: An error occurred during the processing of an OPEN macro.
System completion code 013, with the return code, accompanies this
message.

In the message text:

rc          The return code.

mod         The name of the module in which the error was detected.

jjj        The job name.

F1=Help    F3=Exit    F7=Up      F8=Down    F12=Cancel

```

パラメーターを指定せずに LOOKUP コマンドを入力すると、メッセージ ID、異常終了コード、またはその他の使用可能な情報項目を指定できる画面が表示されます。

図 69. Lookup Search and Browse 画面の例

```

Lookup Search and Browse                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR_

Either search for abend codes, messages, and miscellaneous information by
typing a pattern, or browse such information using the expand/collapse browser
below.

Search. . . . . : _____

+ Abend Codes
+ Messages
+ Miscellaneous Information

F1=Help    F3=Exit    F7=Up      F8=Down    F12=Cancel

```

Lookup Search and Browse 画面では、「検索」フィールドに引数を指定するか (構文は [LOOKUP ページ 112](#) と同じです)、またはポイント・アンド・シュート・フィールドの展開 (+) と縮小 (-) を使用して必要な情報にナビゲートすることができます。

カーソルをポイント・アンド・シュート・フィールドの + (展開) に置くと、選択可能なサブカテゴリが表示されます。例えば、[図 69 : Lookup Search and Browse 画面の例 ページ 129](#)で Abend Codes の横の + 記号にカーソルを置いて Enter キーを押すと、展開された以下の情報が選択可能になります。

```
- Abend Codes
+ IMS User Abend Codes
+ Language Environment User Abend Codes
+ CICS User Abend Codes
+ MVS Abend Codes
+ Messages
+ Miscellaneous Information
```

次の IMS™ User Abend Codes を展開すると、選択可能な IMS™ ユーザー異常終了コード説明のリストが提供されます。

```
- Abend Codes
- IMS User Abend Codes
  U0001
  U0002
  U0005
  U0008
  U0009
  .
  . (not all abend codes shown)
  .
  U3469
  U3498
  U3499
  U3999
  U4095
+ Language Environment User Abend Codes
+ CICS User Abend Codes
+ MVS Abend Codes
+ Messages
+ Miscellaneous Information
```

Fault Analyzer から提供される情報、または IDIHUSRM データ・セットのメンバー IDI.SIDISAM1 で指定された情報のみが入力可能です。

## ファイルへの対話式画面のコピー

COPY コマンドを使用すると、現行画面 (スクロール可能域全体を含む) のコピーを、順次データ・セット、または区分データ・セット内のメンバーに書き込むことができます。(コマンド構文については、[COPY ページ 106](#)を参照してください。)

例えば、次のコマンドを入力するとします。

```
COPY PRINT(A)
```

区分データ・セット *prefix.PRINT* 内のメンバー A に、現行の画面が書き込まれます。ここで、*prefix* は現行の TSO 接頭部です。

パラメーターを指定せずに COPY コマンドを入力すると、データ・セット名を指定できるポップアップ・パネルが表示されます。

## 製品の著作権、ライセンスおよびバージョン情報の表示

「ヘルプ」 > 「Fault Analyzer について」を選択することにより、Fault Analyzer の著作権および一般的な使用法情報を表示できます。(アクション・バーのプルダウン・メニュー ページ 100 を参照してください。)この画面にはインストール済みの Fault Analyzer 製品の、インストール済みバージョン、リリース、メンテナンス・レベルおよび適用済みフィックスも表示されます。

## ヒストリー・ファイル項目の削除

障害項目をヒストリー・ファイルから自動的に削除することも、明示的に削除することもできます。

### 障害項目の自動削除

Fault Analyzer ヒストリー・ファイルに障害項目の最大数が含まれている場合、新しい障害が発生すると、ヒストリー・ファイルから最も古いアンロックされた障害項目が自動的に削除されます。ISPF を使用してヒストリー・ファイル内の障害項目の数を構成するには、「File > Fault History File Properties」をクリックします。

### 障害項目の明示的な削除

ISPF Fault Entry List 画面を使用して、項目に対して D 行コマンドを入力することにより、ヒストリー・ファイルから障害項目を明示的に削除することができます。一連の障害ヒストリー項目を削除することもできます。最初と最後の障害ヒストリー項目の横に DD と入力すると、2つの項目およびその間に表示されるすべての項目が削除されます。突き合わせを使用して、削除する項目を先にまとめることもできます。(障害項目のソートと突き合わせ ページ 78 を参照。)一連の突き合わせた項目セットを削除する場合、画面上に表示された項目のみが削除されます。オリジナルのヒストリー・ファイルでは表示されていたが、現在は表示されていない項目の間に配置されている項目があるとしても、その項目は削除されません。

障害項目のロック・フラグが非数値または非ブランクの値に設定されている場合、または障害項目の作成後、指定された日数がまだ経過していない場合は、その項目を削除できません。アクセス権限が不十分であるため、または障害項目がロックされているために障害項目を削除できない場合は、その理由が表示されます。障害項目のロック ページ 132 および 障害項目情報の表示 ページ 134 を参照してください。

また、IDIUTIL ユーティリティを使用して障害項目を削除することもできます。DELETE 制御ステートメント ページ 445 を参照してください。

### 障害項目削除の確認の構成

デフォルトでは、Fault Analyzer は選択した障害項目を削除する前に確認を求めるプロンプトを表示します。

図 70. Confirm Fault Entry Deletion 画面の例

```

Command ==> _____ Scroll ==> CSR
Press Enter to delete the selected fault entries, or press PF3/PF12 to
abort the delete request.
Keep delete confirmation on : Y (Y/N)
Selected fault entries:
  SW00882 SW00886 SW00926 SW00927 SW16070 SW16072
*** Bottom of data.

F1=Help    F3=Exit    F7=Up      F8=Down    F12=Cancel

```

確認画面をバイパスするには、「**Keep delete confirmation on**」を「N」に設定します。

確認画面を復元するには、「**Confirm Fault Entry Deletion**」オプションを「Y」に設定します。詳細については、[Fault Analyzer プリファレンス ページ 117](#) を参照してください。

## 障害項目のロック

障害項目をロックして、ヒストリー・ファイルから削除されないようにすることができます。障害項目をロックしておく日数を指定したり、ロックが無期限になる(障害項目が削除されない)ように指定することができます。ロックされた障害項目を Fault Entry List 画面で強調表示することができます。オプションの IDIXLOCK 出口では、インストールの全体にわたって障害項目フラグ値を制御できます。

デフォルトでは、障害項目はロックされていません。障害項目のロック・フラグを変更するには、その障害項目に対する UPDATE 権限が必要です。

### Fault Entry List 画面からの障害項目のロック

Fault Entry List 画面から、障害項目に対して L 行コマンドを適用し、デフォルトのロック構成を使用して項目をロックすることができます。デフォルトのロック構成の設定については、[Fault Analyzer プリファレンス ページ 117](#) を参照してください。

### Fault Entry Information 画面からの障害項目のロック

障害項目のロックは、Fault Entry Information 画面から構成できます。

1. 障害項目情報を表示するには、以下のいずれかを実行します。
  - Fault Entry List 画面で、障害項目に対して ? 行コマンドを入力します。
  - Interactive Reanalysis Report で、以下のいずれかを実行します。
    - INFO コマンドを出します。詳細については、[INFO ページ 111](#) を参照してください。
    - アクション・バーから、**File > Fault Entry Information** を選択します。[アクション・バーのプルダウン・メニュー ページ 100](#) を参照してください。
2. ロック・フラグを構成します。[障害項目情報の表示 ページ 134](#) を参照してください。

## ユーザー出口を使用して障害項目をロックする

ENV.LOCK\_FLAG フィールドを使用して、ユーザー出口を介して障害項目をロックすることができます。[ENV - 共通出口環境情報 ページ 650](#)を参照してください。

## 障害項目ロック・フラグ値の制御

オプションで、システム・プログラマーを使用して障害項目のロックをインストールの全体にわたって制御できます。例えば、障害項目をロックできる日数を制限したり、障害項目を無期限にロックできないようにすることが可能です。障害項目ロック・フラグが(分析制御ユーザー出口または Fault Analyzer ISPF インターフェースを介して)変更される際、制御権はオプションの IDIXLOCK 出口ルーチンに渡されます。IDIXLOCK 出口は変更を受け入れるか、別の値に置き換えることができます。

IDIXLOCK 出口では、ユーザー ID またはセキュリティー・サーバー・グループ ID に基づいてさまざまな制限を課すことが可能です。例えば、IDIXLOCK 出口で管理者に任意のロック・フラグ値の設定を許可する一方で、他のユーザーをインストール全体のルールに対して制限することができます。

データ・セット IDI.SIDISAM1 内のメンバー IDISLOCK には、サンプル出口を作成するジョブが含まれています。この出口は、IDIXLOCK ライブラリーまたは別の APF に許可されたライブラリー内の IDI.SIDIAUTH という名前のロード・モジュールである必要があります。当該のライブラリーを LNKLIST に含めて、IDIXLOCK ロード・モジュールをすべてのジョブで検出できるようにする必要があります。

## ロックされた障害項目の強調表示

テキストの色を使用して、Fault Entry List 画面のロックされた障害項目の Fault ID 列および Lock 列をロックすることができます。強調表示することにより、ロックされた障害項目を画面で見つけやすくなります。画面スペースを節約する場合は、ロックされた項目を強調表示によって示し、Locked 列を画面から除外することができます。(Fault Entry List の列構成 [ページ 71](#)を参照してください。)逆に、Locked 列は強調表示に代わるテキストで、スクリーン・リーダーや色覚障害のある人でも読むことができます。Locked 列に基づいて障害項目をソートすることはできますが、強調表示に基づいてソートすることはできません。

ロックされた障害項目の強調表示を構成する方法については、[Fault Analyzer プリファレンス ページ 117](#)を参照してください。

次の例を考えてみます。

- 太字テキストは、強調表示された障害項目の列を示しています(ライブ画面では、構成されたテキストの色が表示されます)。
- 強調表示は、ロックされた障害項目の Fault\_ID 列と Locked 列の両方に対して構成されます。
- Fault Entry List 画面には、Fault ID が F82335 と F02203 のロックがアクティブになっていることが示されています。
- Fault ID が F82335 のロックはアクティブであり、障害が発生した日付から20日間延長されます。
- Fault ID が F02203 のロックがアクティブであり、有効期限がありません。
- Fault ID が C1C52396 の20日間のロックが期限切れになりました。

図 71. ロックされた項目が強調表示された Fault Entry List の例

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'EXPUSER.HIST'

  Fault ID  Lock  Locked  Date       Time       Abend  Sys/Job  Job/Tran  User ID
---  F02222  No      No      2020/04/16 10:08:42 S0C7   FAE1     IDIVPCOB EXPUSER
---  F82335 20     Yes     2020/04/01 10:37:59 ASRA    AS720F1  SW99     EXPUSER
---  F02214  No      No      2020/03/27 09:33:09 U0100  RRG1     EM062190 EXPUSER
---  CIC52396 20     No      2020/03/26 10:09:54 SNAP   E114CICS CS65     TESTFA1
---  F02208  No      No      2020/03/24 10:18:07 S0CB   FAE1     COBMST3  EXPUSER
---  F02207  No      No      2020/03/24 10:17:50 S0CB   FAE1     COBMST3  EXPUSER
---  F02203 /     Yes     2020/03/19 11:09:47 SNAP   FAE1     FIX       EXPUSER
---  F02200  No      No      2020/03/19 10:15:24 SNAP   FAE1     CUR       EXPUSER
---  F02195  No      No      2020/03/19 10:04:27 SNAP   FAE1     UI63650  EXPUSER
---  F02194  No      No      2020/03/19 10:03:15 SNAP   FAE1     UI63214  EXPUSER
---  F02193  No      No      2020/03/19 10:02:28 SNAP   FAE1     UI61521  EXPUSER
---  F02192  No      No      2020/03/19 10:01:18 SNAP   FAE1     UI59890  EXPUSER
---  F02191  No      No      2020/03/19 10:00:27 SNAP   FAE1     UI59820  EXPUSER
---  F02190  No      No      2020/03/19 09:59:21 SNAP   FAE1     UI58414  EXPUSER
---  F02189  No      No      2020/03/19 09:58:15 SNAP   FAE1     UI56500  EXPUSER
---  F02188  No      No      2020/03/19 09:56:43 SNAP   FAE1     UI56500  EXPUSER

F1=Help      F3=Exit      F4=MatchCSR  F5=RptFind   F6=Actions   F7=Up
F8=Down      F10=Left     F11=Right    F12=MatchALL

```

## 障害項目情報の表示

ヒストリー・ファイルは、1項目につき1行の情報として表示されます。そのため、障害に関する情報をすべて同時に表示できるわけではありません。

追加情報を表示するには、Fault Entry List 画面の項目に対して ? 行コマンドを入力します。対話式再分析レポートから、INFO コマンド ([INFO ページ 111](#)を参照) を発行するか、またはアクション・バーのFileプルダウン・メニュー ([アクション・バーのプルダウン・メニュー ページ 100](#)を参照) から **Fault Entry Information** を選択することによって、同じことを行うことができます。

Fault Entry Information 画面の例は、次のとおりです。

図 72. Fault Entry Information 画面の例

```

File View Services Help
-----
Fault Entry Information                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Fault ID. . . . . : BAT00009
User Name . . . . . : _____
User Title. . . . . : _____
Lock Flag . . . . . : ___ (Not locked)
Lock/Unlock User ID . . . . . : _____
Abend Code. . . . . : S0C4
POF Module Name . . . . . : IBMBLIIA
POF Program Name. . . . . : IBMBLIII
POF Offset. . . . . : 96C
Abend Date. . . . . : 2019/09/22
Abend Time. . . . . : 15:03:02
Job Name. . . . . : PLI23
Job ID. . . . . : JOB30836
Job Execution Class . . . . . : A
Job Type. . . . . : Batch
Job Step Name . . . . . : G0
EXEC Program Name . . . . . : @960IDI
User ID . . . . . : RTURNER
System Name . . . . . : FAE1
Application ID. . . . . : n/a
CICS Transaction ID . . . . . : n/a
CICS Task Number. . . . . : n/a
CICS Terminal ID. . . . . : n/a
CICS Terminal Netname . . . . : n/a
IMS Program Name. . . . . : n/a
Accounting Information. . . . . : n/a
Duplicate Count . . . . . : 2
Minidump Pages. . . . . : 60
History File Data Set Name. : DA.DCAT
Java DTFJ processing status : Started 2019/10/22 17:55:03
MVS Dump Data Set Name. . . : n/a
MVS Dump Status . . . . . : Not found

*** Bottom of data.
  F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
  F10=Left     F11=Right     F12=retrieve

```

画面サイズによっては、情報をすべて表示するのにスクロールダウンが必要な場合があります。

画面には、次のフィールドが表示されます。

#### Fault ID

選択された障害項目の ID。

#### ユーザー名

ユーザー保守の入出力フィールド。例えば、障害が割り振られたユーザーの名前や ID などです。このフィールド内容を変更するには、PF3 で戻る前に Enter キーを押してください。このフィールドの初期内容は、IDISNAP によって指定されます。

#### ユーザー・タイトル

障害の簡略説明を目的とした、ユーザー保守の入出力フィールド。このフィールドが使用される場合、バッチ分析レポートの最上部、および対話式再分析レポートの最初の画面の最上部に表示されます。このフィールド内容を変更するには、PF3 で戻る前に Enter キーを押してください。

## ロック・フラグ

ユーザーが維持する入出力フィールドで、障害項目が誤って削除されないようにするためのメカニズムを提供します。

このフィールドに数値 0 から 99 が含まれている場合、障害項目の有効期限の制御はアクティブです。指定された値は、障害項目が作成以後にロックされたままとなる日数です。時刻は考慮されません。例えば、障害項目が作成されたときに値 1 が指定されていると、同日の午前零時にロックが解除されます。したがって、値 0 は障害項目がロックされないことを意味します。

障害項目の有効期限の制御がアクティブで、まだ有効期限が過ぎていない場合、またはこのフラグを他の非ブランク文字に設定した場合、障害項目に対して以下は実行されません。

- IDIUTIL DELETE 処理 (詳しくは、[DELETE 制御ステートメント ページ 445](#)を参照)。



**注:** デフォルトでは障害項目は削除されませんが、このデフォルトは IDIUTIL ユーザー出口を使用してオーバーライドできます (詳しくは、[IDIUTIL 削除ユーザー出口 ページ 511](#)を参照)。

- 「D」または「DD」行コマンドを使用した ISPF インターフェースの Fault Entry List 画面からの削除 (詳しくは、[ヒストリー・ファイル項目の削除 ページ 131](#)を参照)。
- ヒストリー・ファイルの障害項目の最大数に達したため、およびロックされている障害項目がヒストリー・ファイル内で最古の障害項目になったために行われる自動削除。この場合、削除される障害項目の検索は、ロックされていない最古の障害項目が見つかるまで、日時順で継続されます。

デフォルトでは、ロック・フラグはブランクに設定され、障害項目の削除は妨げられません。

このフィールドには、次のように印刷可能文字を入力できます。

- 印刷不能文字が入力された場合は「/」に変更されます。
- 小文字が入力された場合は大文字に変換されます。

現在のロック状況はロック・フラグの直後に表示され、その値を変更した後に Enter キーを押した場合、ロック状況が更新されます。

ロック・フラグは以下の方法でも変更できます:

- 任意のユーザー出口で、ENV.LOCK\_FLAG フィールドの値を変更する。[ENV - 共通出口環境情報 ページ 650](#)を参照してください。
- オプションの IDIXLOCK 出口。[障害項目ロック・フラグ値の制御 ページ 133](#)を参照してください。

## Lock/Unlock User ID

「Lock Flag」設定を最後に変更したユーザー ID を示します。

## Abend Code

最初の (複数ある場合) 異常終了コード。



IDISNAP を使用して作成された障害項目の場合は、異常終了コード「SNAP」が示されます。

**POF Module Name**

障害点モジュール名。

**POF Program Name**

障害点プログラム名。

**POF Offset**

障害点オフセット。

**Abend Date**

異常終了が発生した日付。

**Abend Time**

異常終了が発生した時刻。

**Job Name**

項目を書き込んだバッチ・ジョブ、開始済みタスク、または TSO ユーザー ID の名前。

**Job ID**

異常終了したジョブの JES ID。

**Job Execution Class**

異常終了したジョブの JES 実行クラス。

**Job Type**

異常終了ジョブ・タイプ。

**Job Step Name**

項目を書き込んだジョブ・ステップの名前。

**EXEC Program Name**

異常終了ジョブ・ステップの JCL EXEC ステートメントで指定されたプログラム名。

**User ID**

異常終了ジョブに関連するユーザー ID。

**System Name**

異常終了が発生したシステム名。

**CICS® Transaction ID**

CICS® トランザクションで障害が発生した場合の、書き込まれる項目の原因である CICS® トランザクションの ID。

**CICS® タスク番号**

CICS® トランザクションで障害が発生した場合の、書き込まれる項目の原因であるタスク番号。

#### CICS® Terminal ID

CICS® トランザクション障害の場合は、CICS® 端末 ID。

#### CICS® Terminal Netname

CICS® トランザクション障害の場合は、CICS® 端末ネット名。

#### IMS™ Program Name

IMS™ プログラム名。

#### Accounting Information

ジョブ・アカウント情報。

#### Duplicate Count

重複する障害が記録されてから、同じ履歴・ファイルを使用して、この障害が発生した数。



**注:** この値は、パフォーマンスの理由のみのために、揮発性履歴・ファイル・キャッシュに保持されているため、発生した重複障害を必ずしもすべて含まない可能性があります。

ゼロ以外の値が表示され、障害についての重複する詳細が入力できる場合は、値がポイント・アンド・シュート・フィールドになります。ここにカーソルを置いて Enter を押すと、選択できます。この選択により、Fault Entry Duplicate History 画面が表示されます (詳しくは、[障害項目の重複履歴の表示 ページ 139](#) を参照)。

#### Minidump Pages

この障害用に履歴・ファイルに書き込まれたミニダンプ・ページの数。

#### History File Data Set Name

表示された障害項目を含む履歴・ファイルのデータ・セット名。

#### Java™ DTFJ Processing Status

Java™ アクティビティが検出された障害項目の場合、このフィールドには、非同期 Java™ DTFJ 処理の状況が以下のいずれかとして示されます。

- ダンプ・エラー
- Pending
- Started *date time*
- Finished, elapsed seconds *seconds*

#### MVS™ Dump Data Set Name

関連するダンプ・データ・セット名 (ある場合)。

#### MVS™ Dump Status

MVS™ ダンプ・データ・セット状況。

MVS™ ダンプ・データ・セット状況フィールドには、現行の障害に関連する可能性があるダンプについての情報が表示されます。このフィールドで使用可能な値は、以下のとおりです。

**Available**

MVS ダンプ・データ・セットは障害項目と関連付けられ、データ・セットは存在します。

**Not found**

MVS ダンプ・データ・セットは障害項目と関連付けられていましたが、データ・セットが存在しなくなりました。

いずれかのユーザー管理フィールドを変更後、Fault Entry Information 画面を終了すると、障害項目を新しい情報で更新するかどうかを確認するプロンプトが表示されます。障害項目を更新するには、Enter を押します。障害項目を更新しない場合は、PF3 または PF12 を押します。

## 障害項目の重複ヒストリーの表示

Fault Entry Duplicate History 画面には、その障害タイプについて有効である NoDup オプション制限時間内に発生し、定義されている重複基準に適合している、その他の障害に関する情報が表示されます。

この画面は、以下で呼び出すことができます:

- Fault Entry List 画面から障害項目に対して H 行コマンドを入力する。
- Fault Entry List 画面で、障害項目がポイント・アンド・シュート・フィールドとして示されたときに、この項目の「Dups」列の値にカーソルを置いて Enter を押す。
- 対話式再分析レポート内から DUPS 基本コマンドを発行する。
- 「Fault Entry Information」画面で、「Duplicate Count」値がポイント・アンド・シュート・フィールドとして示されたときに、この値にカーソルを置いて Enter を押す。

以下は、Fault Entry Duplicate History 画面の例です。

図 73. Fault Entry Duplicate History 画面の例

```

File View Services Help
-----
Fault Entry Duplicate History                               End of data
Command ==> ----- Scroll ==> CSR

Most recent duplicate
  occurred. . . . . : 2019/06/28 15:22:13
Initial abend occurred. . . : 2019/06/28 15:20:55
Total duplicate count . . . : 4

Duplicate details in reverse chronological order:

Date      Time      Jobname  Job_ID  System  Dup_Type  User_ID  Stepname
2019/06/28 15:22:13 CICS5FA1 JOB48697 FAE1     Normal    SIMCOC2  CICS5FA1
2019/06/28 15:22:02 CICS5FA1 JOB48697 FAE1     Normal    SIMCOC2  CICS5FA1

Date      Time      Jobname  Job_ID  System  Dup_Type
2019/06/28 15:21:08 CICS5FA1 JOB48697 FAE1     Fast
  User_ID  Term_ID  Count
  SIMCOC2  1265     2

*** Bottom of data.
F1=Help      F3=Exit      F5=RptFind   F7=Up        F8=Down      F10=Left
F11=Right
    
```

次の情報が提供されます。

**発生した最新の重複 (Most recent duplicate occurred)**

この障害の最新の重複が発生したローカル日時。日時形式は、有効なロケール・オプションに応じて異なります。

**発生した初期異常終了 (Initial abend occurred)**

項目が書き込まれる原因となった初期障害が発生した時のローカル日時。日時形式は、有効なロケール・オプションに応じて異なります。

**合計重複カウント (Total duplicate count)**

この障害に対して登録された重複の総数。

**逆年代順の重複の詳細 (Duplicate details in reverse chronological order)**

重複発生の詳細。

重複が NoDup(Normal) 処理に起因しているか、NoDup(CICSFast) 処理に起因しているかによって、提供される情報が異なります。

• NoDup(Normal) 重複

NoDup(Normal) 重複の場合、各インスタンスで提供される情報は、次の内容です。

**Date**

重複が発生した日付。日付形式は、有効なロケール・オプションに応じて異なります。

**Time**

重複が発生した時刻。時刻形式は、有効なロケール・オプションに応じて異なります。

**Jobname**

重複障害の原因となったジョブの名前。

**Job ID**

重複障害の原因となったジョブの JES ジョブ ID。

**System**

重複障害の原因となったジョブが実行されていたシステムの名前。

**Dup Type**

重複のタイプ (NoDup(Normal) 重複の場合は、必ず「通常 (Normal)」)。

**User ID**

重複障害の原因となったジョブに関連付けられたユーザー ID。

**Stepname**

重複障害の原因となったジョブ・ステップの名前。

複数の NoDup(Normal) が連続して発生した場合、共通の見出しの下でグループ化されます。

- NoDup(CICSFast) 重複

NoDup(CICSFast) 重複の場合、提供される情報は、以下の内容です。

- 記録間隔内に発生したすべての重複障害 (1 つ以上) に該当する詳細を持つ、共通のセクション。

**Date**

最初の重複が記録間隔内に発生した日付。日付形式は、有効なロケール・オプションに応じて異なります。

**Time**

最初の重複が記録間隔内に発生した時刻。時刻形式は、有効なロケール・オプションに応じて異なります。

**Jobname**

重複障害の原因となったジョブの名前。

**Job ID**

重複障害の原因となった CICS® 領域の JES ジョブ ID。

**System**

重複障害の原因となった CICS® 領域が実行されていたシステムの名前。

**Dup Type**

重複のタイプ (NoDup(CICSFast) 重複の場合は、必ず「高速 (Fast)」)。

- 記録間隔内に発生した最大 10 個の固有障害の詳細情報。

#### User ID

重複障害の原因となった CICS® ユーザー ID。

#### Terminal ID

重複障害の原因となった CICS® 端末 ID。

#### Count

記録間隔内に発生した CICS® ユーザー ID と端末 ID のこの固有の組み合わせのオカレンス総数。



**注:** すべての「カウント (Count)」値の合計が、記録間隔内に発生した重複の合計と等しくない場合、詳細を入手できない重複の数に関する情報を含む注記が提供されません。

この状態は、記録間隔中に、10 を超える CICS® ユーザー ID と端末 ID の固有の組み合わせで重複障害が発生した場合に、生じる可能性があります。



**注:** 表示されたすべての重複障害の合計が、障害項目の合計重複カウントと等しくない場合は、詳細を入手できない重複の数に関する情報を含む注記が提供されます。

この状態は、障害項目の詳細に記録できるよりも多くの重複障害が発生した場合に、生じる可能性があります。

## 重複障害項目の展開と縮小

[障害項目の重複ヒストリーの表示 ページ 139](#) で説明されているように、重複障害項目情報を表示せず、個別の障害項目に展開することができます。展開表示において、重複カウントが 0 より大きい障害項目では、元の障害項目とその重複した項目が個々の障害項目として表示されます。展開表示では、それぞれの重複した障害の発生を経時的に他の障害のコンテキストで確認できます。

**図 74:** 縮小された重複を示す [Fault Entry List 画面の例 ページ 143](#) にデフォルトの縮小表示を示します。障害項目 F02417、F61327、F82311 のインスタンスが重複しています。

図 74. 縮小された重複を示す Fault Entry List 画面の例

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List                      Refresh complete
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'XMPUSER.HIST.ABC'
-----
  Fault ID Dups  Lock MD Pages Date Time System  Abend  Jobname  J
  ---
  F02418           82 2020/09/14 11:26:31 FAE1   S0C7  IDIVPCOB B
  F02422          100 2020/09/14 11:25:52 FAE1   S06F  IDISSE1 S
  F02417           1 84 2020/09/08 10:12:13 FAE1   S0C7  IDIVPCOB B
  F61327           4 89 2020/09/08 10:06:39 FAE1   S0C7  FADUP   B
  F02416           85 2020/08/31 18:48:52 FAE1   S0C7  IDIVPCOB B
  F02143           / 52 2020/02/10 13:14:46 FAE1   S0C7  S0C7A   B
  F02140           45 2020/02/05 12:08:53 FAE1   S0C7  S0C7A   B
  F82311          12 / 276 2020/02/05 08:54:45 FAE1   ASRA  AS720F1 C
  F02139           25 52 2020/02/04 12:39:34 FAE1   S0C4  ASM64B  B
  F02138           25 80 2020/02/04 12:36:19 FAE1   S0C1  ASMLE64 B
  F02137           25 52 2020/02/04 09:44:54 FAE1   S0C4  ASM64B  B

** Bottom of data.

```

図 75: 展開された重複を示す Fault Entry List 画面の例 ページ 143 に示すように重複障害項目を展開するには、**View > Expand Duplicate Fault Entries** を選択します。

Fault Analyzer Preferences 画面から必要な強調表示オプションを選択すると、展開された重複障害項目をより容易に識別できるようになります。Fault Analyzer プリファレンス ページ 117 を参照してください。

図 75. 展開された重複を示す Fault Entry List 画面の例

```

File Options View Services Help
-----
IBM Fault Analyzer - Fault Entry List (Expanded)          Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'XMPUSER.HIST.ABC'
-----
  Fault ID Dups  Lock MD Pages Date Time System  Abend  Jobname  J
  ---
  F02418           82 2020/09/14 11:26:31 FAE1   S0C7  IDIVPCOB B
  F02422          100 2020/09/14 11:25:52 FAE1   S06F  IDISSE1 S
  F02417           1 84 2020/09/08 10:12:37 FAE1   S0C7  FADUP   B
  F02417           1 84 2020/09/08 10:12:13 FAE1   S0C7  IDIVPCOB B
  F61327           1 89 2020/09/08 10:11:30 FAE1   S0C7  FADUP   B
  F61327           1 89 2020/09/08 10:06:47 FAE1   S0C7  FADUP   B
  F61327           1 89 2020/09/08 10:06:46 FAE1   S0C7  FADUP   B
  F61327           1 89 2020/09/08 10:06:46 FAE1   S0C7  FADUP   B
  F61327           4 89 2020/09/08 10:06:39 FAE1   S0C7  FADUP   B
  F02416           85 2020/08/31 18:48:52 FAE1   S0C7  IDIVPCOB B
  F02143           / 52 2020/02/10 13:14:46 FAE1   S0C7  S0C7A   B
  F02140           45 2020/02/05 12:08:53 FAE1   S0C7  S0C7A   B
  F82311          11 / 276 2020/02/05 08:57:13 FAE1   ASRA  AS720F1 C
  F82311           1 / 276 2020/02/05 08:57:13 FAE1   ASRA  AS720F1 C
  F82311          12 / 276 2020/02/05 08:54:45 FAE1   ASRA  AS720F1 C
  F02139           25 52 2020/02/04 12:39:34 FAE1   S0C4  ASM64B  B

```

## ヒストリー・ファイル項目のコピー

コピーするには、Fault Entry List 画面で障害項目に対して C 行コマンドを入力します。

2 つの CC 範囲マーカを使用して、一連の障害ヒストリー項目をコピーすることもできます。一連の突き合わせた項目セットをコピーする場合、画面上に表示された項目のみがコピーされます。オリジナルのヒストリー・ファイル上で、表示されていないが、表示された項目の間に配置されている項目があるとしても、その項目はコピーされません。

Enter キーを押した後、C または CC コマンドでコピー用に選択された障害項目のリストを示す画面が提示されます。M または MM コマンドで移動用に選択された障害項目があれば、それらも含まれます。画面には、障害項目のコピー先にする履歴・ファイルを指定するフィールドがあります。このフィールドは、最後に指定した履歴・ファイルの名前で初期化されます。以下に、Enter キーを押した後に表示される Specify Move/Copy Options 画面の例を示します。

別のオプション **Include Tightly Coupled Dump Data Sets** では、次の設定が可能です。

#### すべて

選択した障害項目に一意に属する、すべての関連するダンプ・データ・セットが複製されるように指定します。

#### 条件付き

選択した障害項目に一意に属する、すべての関連するダンプ・データ・セットが、障害項目がミニダンプに含まれていない場合にのみ、複製されるように指定します。例えば、リカバリー障害記録 (RFR) 障害項目がまだ分析されていない場合などが考えられます。

#### None

関連するダンプ・データ・セットがコピーされないことを指定します。

図 76. Specify Move/Copy Options 画面の例

```
Specify Move/Copy Options                               Line 1 Col 1 76
Command ==> ----- Scroll ==> CSR

Verify or change move/copy options for the selected fault entries and
press Enter, or press PF3/PF12 to abort the move/copy request.

Move/Copy Options:
  Destination History File: 'IDI.COPY.HIST'
  Include Tightly Coupled
  Dump Data Sets. . . . . : N (All/Conditionally(*)/None)
  (*) Only include tightly coupled dump data sets for fault entries with no
  minidump.

Selected Fault Entries:
  BAT14648 BAT14649 BAT14650 BAT14651 BAT14652 BAT14653 BAT14654 BAT14655
  BAT14656 BAT14657 BAT14658 BAT14659 BAT14660 BAT14661 BAT14662 BAT14663
  BAT14664 BAT14665 BAT14666 BAT14667 BAT14668 BAT14669 BAT14670 BAT14671
  BAT14672

*** Bottom of data.
  F1=Help      F3=Exit      F7=Up        F8=Down      F12=Cancel
```

移動/コピー処理を開始する前に、宛先として指定された履歴・ファイルに対する UPDATE アクセス許可の有無が検査されます。指定された履歴・ファイルが存在しない場合、New History File Allocation 画面が表示されます (詳細については、[新しい履歴・ファイルの割り振り ページ 93](#) を参照してください)。

可能な場合は、オリジナルの障害 ID が保存されます。ただし、同じ障害 ID が宛先履歴・ファイルに既に存在する場合、障害番号は次の有効な ID に増分されます。

1 つ以上の障害項目のコピーが成功しなかった場合は、詳しい理由を述べた画面が表示されます。

障害項目のコピーに必要な TSO ユーザーのアクセス許可レベルは READ です。



## ヒストリー・ファイル項目の移動

「障害エントリー・リスト」画面で M ライン・コマンドを使用して、障害項目を移動します。

2つの MM 範囲マーカを使用して、一連の障害ヒストリー項目を移動することもできます。一連の突き合わせた項目セットを移動する場合、画面上に表示された項目のみが移動されます。オリジナルのヒストリー・ファイル上で、表示されていないが、表示された項目の間に配置されている項目があるとしても、その項目は移動されません。

Enter キーを押した後、M または MM コマンドで移動用に選択された障害項目のリストを示す画面が提示されます。C または CC コマンドでコピー用に選択された障害項目があれば、それらも含まれます。画面には、障害項目の移動先にするヒストリー・ファイルを指定するフィールドがあります。このフィールドは、最後に指定したヒストリー・ファイルの名前で初期化されます。Enter キーを押した後に表示される Specify Move/Copy Options 画面の例については、[ヒストリー・ファイル項目のコピー ページ 143](#)を参照してください。

移動/コピー処理を開始する前に、宛先として指定されたヒストリー・ファイルに対する UPDATE アクセス許可の有無が検査されます。指定されたヒストリー・ファイルが存在しない場合、New History File Allocation 画面が表示されます (詳細については、[新しいヒストリー・ファイルの割り振り ページ 93](#)を参照してください)。

可能な場合は、オリジナルの障害 ID が保存されます。ただし、同じ障害 ID が宛先ヒストリー・ファイルに既に存在する場合、障害番号は次の有効な ID に増分されます。

1つ以上の障害項目の移動が成功しなかった場合は、詳しい理由を述べた画面が表示されます。

ロックされた障害項目は移動できません。ロックされた障害項目を移動する必要がある場合には、最初に ? 行コマンドを発行して、Lock Flag (ロック・フラグ) 値をブランクに変更することによって Fault Entry Information (障害項目情報) 画面からその障害項目のロックを解除します。必要な場合には、Lock Flag (ロック・フラグ) 値をブランク以外の値に変更することによって、移動された障害項目をもう一度ロックします。Lock Flag (ロック・フラグ) に使用できる値については、[障害項目情報の表示 ページ 134](#)を参照してください。

障害項目の移動に必要な TSO ユーザーのアクセス許可レベルは ALTER です。

## ヒストリー・ファイル項目の送信

「障害エントリー・リスト」画面で X ライン・コマンドを使用して、障害項目を表示します。

2つの XX 範囲マーカを使用して、一連の障害ヒストリー項目を XMIT することもできます。一連の突き合わせた項目セットを XMIT する場合、画面上に表示された項目のみが XMIT されます。オリジナルのヒストリー・ファイル上で、表示されていないが、表示された項目の間に配置されている項目があるとしても、その項目は XMIT されません。

Enter キーを押した後、X または XX コマンドで XMIT 用に選択された障害項目のリストを示す画面が提示されます。障害項目の送信先の宛先ノードおよびユーザー ID と、送信されるデータ・セットのタイプ (シーケンシャル (SEQ) または区分 (PDS)) を指定できます。これらのフィールドは、最後に指定した値で初期化されます。以下に、Enter キーを押した後に表示される Specify XMIT Options 画面の例を示します。

図 77. Specify XMIT Options 画面の例

```

Specify XMIT Options                               Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR_

Verify or change XMIT options for the selected fault entries and press
Enter, or press PF3/PF12 to abort the XMIT request.

XMIT Options:
Destination Node. . . . . : PTHVM3
Destination User ID . . . . : NWILKES
Data Set Type . . . . . : SEQ (PDS/SEQ)

Selected Fault Entries:
BAT14648 BAT14649 BAT14650 BAT14651 BAT14652 BAT14653 BAT14654 BAT14655
BAT14672

*** Bottom of data.

F1=Help      F3=Exit      F7=Up        F8=Down      F12=Cancel

```

選択されたそれぞれの障害項目は、指定された宛先へ別々に送信されます。

1 つ以上の障害項目の XMIT が成功しなかった場合は、詳しい理由を述べた画面が表示されます。関連するダンプ・データ・セットは無視されます。

障害項目の XMIT に必要な TSO ユーザーのアクセス許可レベルは READ です。

## 障害項目のパッケージ化

IBM® サポートに送信する障害項目データをパッケージ化するには、Fault Entry List 画面で項目に P と入力します。これにより、JCL を生成し、関連する障害項目データを送信に最適な terse 形式データ・セットにパッケージ化します。JCL は編集セッションで表示されるため、修正および送信できます。

**i ヒント:** これは障害項目を送信するのに必ずしも十分ではありません。生成された JCL での IDIUTIL EXPORT ステップは、存在する場合、関連するダンプ・データ・セットを追加し、Fault Analyzer サポートがすべての適切なデータを受信できるようにします。Package オプションが容量の要件の見積もりを試みますが、正常な実行のために、容量の見積もりを修正する必要がある場合があります。

## セキュリティ上の考慮事項

履歴ファイルの障害項目へのアクセスを管理する最善の方法は、セキュリティー・サーバー XFACILIT リソース・クラスを使用することです。これについては、[履歴ファイル障害項目に対しての XFACILIT リソース・クラスの使用 ページ 366](#)で説明しています。

代わりに、データ・セット・プロファイル・セキュリティーと併せて、ビューを使用する方法 ([ビューの使用 ページ 66](#)を参照) も検討できます。

## 64 ビット・アドレスの指定

以下は、64 ビット・アドレスがサポートされている入力フィールドとコマンド・パラメーターを表示する場合に適用されます。

64 ビットに対応するアドレス・フィールドまたはコマンド・パラメーターは、16 桁の 16 進数字で指定でき、オプションでビット 31 と 32 の間に下線を含むことができます。下線を指定した場合、下線の右側の値は、8 桁になるように先行ゼロが自動的に埋め込まれます。例えば、以下はすべて同一のアドレス指定と見なされます。

```
100100000
1_100000
1_00100000
0000000100100000
00000001_00100000
```

## 2GB から 4GB のアドレスに関する特別な考慮事項

80000000 から 8FFFFFFF の範囲のアドレスは、一般的に 31 ビット・アドレスと見なされ、上位ビット (ビット 0) は無視されます。

この範囲のアドレス (仕様にアンダースコアを含まない) が選択または入力されると、上位ビットは暗黙的に 0 に設定され、結果の 31 バイト・アドレスが表示されます。

2GB から 4GB の範囲のストレージを表示するには、指定されたアドレスにアンダースコアを含める必要があります。例えば、アドレスを 0\_80000000 または単に \_80000000 と入力すると、64 ビット・アドレス 80000000 が表示されます。

## 対話式 IDIS サブシステム・インターフェースの使用

IDIS サブシステム情報オプションは IDIS サブシステムの状況を表示します。

この画面は、IDISINFO 基本コマンドを入力するか、アクション・バーから **Services > IDIS Subsystem Information** を選択することで表示されます。

図 78. IDIS サブシステム情報画面の例

```
File View Services Help
-----
IDIS Subsystem Information                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR_

Started Task Name . . . . . : IDISSE1
ASID. . . . . : X'0133'
Dispatching Priority. . . . . : X'FE'
User ID . . . . . : IDISS
PARM Field Options in Effect: UPDINDEX NOIMAGEFAST FASTEXCLUDE NOXCFGRPSUFFIX
Version . . . . . : V14R1M9 (PH15623 2019/11/19)

Select one of the following options for additional information:
  1. Managed History Files
  2. Excluded History Files

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F7=Up        F8=Down      F10=Left
F11=Right    F12=Cancel
```

IDIS サブシステムが始動されていない、または予想したように応答していない場合、サブシステムの状況に関する特定の情報が表示されます。

それ以外の場合は、以下の情報が表示されます。

**開始済みタスク名 / ジョブ名**

IDIS サブシステムの開始済みタスクまたはジョブの名前。

**ASID**

開始済みタスクまたはジョブの 16 進アドレス・スペース識別子。

**優先順位のディスパッチ**

優先順位をディスパッチする IDIS サブシステム。

**ユーザー ID**

IDIS サブシステムに関連付けられたユーザー ID。

**有効な PARM フィールド・オプション**

IDIS サブシステムで有効なオプション。これらは、デフォルト値または開始済みタスクまたはジョブの PARM フィールドで指定されます。

**バージョン**

IDIS サブシステムのバージョン、リリースおよびメンテナンス・レベル。

2 つの選択可能オプションで追加情報を指定します。

**1. 管理されているヒストリー・ファイル**

現在 IDIS サブシステムにより管理されているヒストリー・ファイルをリストします。

**2. 除外されたヒストリー・ファイル**

現在 IDIS サブシステムにより除外されているヒストリー・ファイルをリストします。

## 管理されているヒストリー・ファイル

**Managed History Files** 画面には、IDIS サブシステムにより現在管理されているヒストリー・ファイルがリストされます。

ALTER データ・セット・アクセス権限を持つヒストリー・ファイルが白で表示されます。S 行コマンドを入力してヒストリー・ファイルを選択できます。ALTER データ・セット・アクセス権限を持たないヒストリー・ファイルが青で表示され、行コマンド用の入力フィールドはありません。

```

IDIS Subsystem Managed History Files                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Line commands: S (Select). Press Enter to refresh.

  History File Name
-- CTEST.RERUN50.FAX1
  DA.DCAT
-- DA.HIST.FAX1
-- DA.PSTORE.HISTORY
  OKHAN.HIST.BATCH
  OKHAN.HIST.BATCH.FAX2
  OKHAN.HIST.CICS.FAX2
  OKHAN.TEST.HIST
-- KCOSMIC.HISTORY

F1=Help    F3=Exit    F5=RptFind  F7=Up      F8=Down    F12=Cancel

```

次のいずれかの方法を使用して、複数の履歴・ファイルを選択できます。

- SS 行コマンドを使用して、履歴・ファイルのブロックの最初と最後を選択します。
- 「repeat 行コマンド」を使用します (\*c)。例えば、\*s と入力して、ブロックの最後の行での \* または \*s で終了するまで、後続行で S コマンドを繰り返します。

履歴・ファイルの選択方法については、**Managed History Files** 画面で F1 を押ししてください。

選択した履歴・ファイルは「IDIS サブシステム選択履歴・ファイル」画面に表示され、ここでその状況を変更できます。例えば、再割り振りのため、または排他的アクセスを必要とするその他のアクティビティーのために、IDIS サブシステム管理履歴・ファイルを停止できます。履歴・ファイルをソートおよびフィルター処理することもできます。詳しくは、[選択された履歴・ファイル ページ 150](#)を参照してください。

## 除外された履歴・ファイル

**Excluded History Files** 画面には、IDIS サブシステムにより現在除外されている履歴・ファイルがリストされます。

ALTER データ・セット・アクセス権限を持つ履歴・ファイルが白で表示されます。S 行コマンドを入力して履歴・ファイルを選択できます。ALTER データ・セット・アクセス権限を持たない履歴・ファイルが青で表示され、行コマンド用の入力フィールドはありません。

```

IDIS Subsystem Excluded History Files                             Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Line commands: S (Select). Press Enter to refresh.

  History File Name
-- LJBERRY.DHIST
  JGALAGA.DHIST
  NWILKES.HIST

** Bottom of data.

F1=Help    F3=Exit    F5=RptFind  F7=Up      F8=Down    F12=Cancel

```

次のいずれかの方法を使用して、複数の履歴・ファイルを選択できます。

- SS 行コマンドを使用して、履歴・ファイルのブロックの最初と最後を選択します。
- 「repeat 行コマンド」を使用します (\*c)。例えば、\*s と入力して、ブロックの最後の行での \* または \*s で終了するまで、後続行で S コマンドを繰り返します。

履歴・ファイルの選択方法については、**Excluded History Files** 画面で F1 を押してください。

選択した履歴・ファイルは「IDIS サブシステム選択履歴・ファイル」画面に表示され、ここでその状況を変更できます。例えば、再割り振りのため、または排他的アクセスを必要とする他のアクティビティーのために、IDIS サブシステム管理履歴・ファイルを停止できます。履歴・ファイルをソートおよびフィルター処理することもできます。詳しくは、[選択された履歴・ファイル ページ 150](#)を参照してください。

## 選択された履歴・ファイル

この表示は、IDIS サブシステム **Managed History Files** またはサブシステム IDIS Subsystem **Excluded History Files** 画面から 1 つ以上の履歴ファイルを選択すると表示されます。

```

IDIS Subsystem Selected History Files                               Line 1 Col 1 80
Command ==>> ----- Scroll ==>> CSR

Line commands: R (Reset), X (Exclude). Press Enter to refresh.

  History File Name                                         Status
  ---              ---
  --- CTEST.DAIOSMVS.DCAT                                   Managed
  --- CTEST.DAIPPMVS.DCAT                                   Managed
  --- CTEST.DAI3AMVS.DCAT                                   Managed
  --- CTEST.DAI3PMVS.DCAT                                   Managed
  --- CTEST.DALEXASM.FAX1                                   Excluded
  --- CTEST.DALEXCBE.FAX1                                   Excluded
  --- CTEST.DALEXIFP.FAX1                                   Excluded
  --- CTEST.DALEXILC.FAX1                                   Excluded
  --- CTEST.DALEXPLI.FAX2                                   Managed
  --- CTEST.DANLEASM.FAX2                                   Managed
  --- CTEST.DANLECBL.FAX2                                   Managed
  --- CTEST.DANLEPLI.FAX2                                   Managed

** Bottom of data.

F1=Help   F3=Exit   F5=RptFind  F7=Up      F8=Down    F12=Cancel

```

各履歴・ファイルの現在の状況として、次のいずれかが表示されます。

### Managed

履歴・ファイルは現在 IDIS サブシステムにより管理されています。

### Excluded

履歴・ファイルは現在 IDIS サブシステムにより除外されています。

### n/a

履歴・ファイルは現在 IDIS サブシステムにより管理、除外のいずれもされていません。

選択した履歴・ファイルの状況を変更できます。例えば、再割り振りのため、または排他的アクセスを必要とする他のアクティビティーのために、IDIS サブシステムによる履歴・ファイルの管理を停止できます。

## 履歴・ファイルのリストのソートおよびフィルター処理

カーソルを「History File Name」ヘッダーに置いて Enter を押すことにより、履歴・ファイルのリストをソートまたは縮小できます。これにより「列属性」画面が表示され、そこでソート順序を選択でき、または \* および % 文字をワイルドカードとして使用して、フィルター条件を入力できます。

- アスタリスク (\*) を指定して、0、1つ、またはそれ以上の文字を示すことができます。
- パーセント記号 (%) を指定して、1つの文字のみを示すことができます。

フィルター処理のもう1つの方法は、前述のワイルドカードを含む、リスト内の文字を上書きして、Enter を押すことです。例えば、次の履歴・ファイルのリストがあるとします。

```
HIST.DEPA.PROD
HIST.DEPA.TEST
HIST.DEPB.PROD
HIST.DEPB.TEST
HIST.DEPC.PROD
HIST.DEPC.TEST
```

履歴・ファイル HIST.DEPB.PROD で、最後の修飾子をアスタリスクで置き換えます。

```
HIST.DEPA.PROD
HIST.DEPA.TEST
HIST.DEPB.*
HIST.DEPB.TEST
HIST.DEPC.PROD
HIST.DEPC.TEST
```

Enter を押した後で、一致した履歴・ファイル名のみが表示されます。

```
HIST.DEPB.PROD
HIST.DEPB.TEST
```

元のソート順序で完全なリストを再表示するには、基本コマンド行で RESET を入力します。

Enter を押して、IDIS サブシステム管理対象履歴・ファイルのリストを最新表示します。

PF3 を押して、「IDIS サブシステム情報」画面に戻ります。

## 行コマンド

選択した履歴・ファイルの状況を変更するのに、2つの行コマンドを使用できます。

### R

**リセット。**現在の状況が「管理」または「除外」の場合、R 行コマンドを入力すると状況 n/a となり、これは、履歴・ファイルが IDIS サブシステムにより、管理または除外されていないことを示します。

現在の状況が n/a の場合、R 行コマンドは無視されます。

### X

**除外。**現在の状況が「管理」または「n/a」の場合、X 行コマンドを入力すると状況 Excluded となり、これは、履歴・ファイルが IDIS サブシステムにより、除外されていることを示します。

現在の状況が「除外」の場合、X 行コマンドは無視されます。

## 現在有効なオプションの表示

Options in Effect 画面には、ISPF インターフェースを起動した際に有効になっているすべての Fault Analyzer オプションのほか、オプションまたはサブオプションが設定されている場所が表示されます。[CUROPTS 基本コマンド ページ 106](#) または [Options->Options in Effect ページ 102](#) を使用して、この画面を表示します。

この画面では、複数の行にまたがるオプションが (各行の右側に正符号を追加した) 継続文字で示されます。この形式は IDICNFxx parmlib メンバーなどの指定に有効な構文なので、この画面から直接オプションをコピーして貼り付けることもできます。

図 79. Options in Effect 画面の例

```

File View Services Help
Options in Effect                               Line 1 Col 1 80
Command ==>>                                   Scroll ==>> CSR

Option                                           Where Last Set
DATASETS(IDIADATA(DA.SYSADATA),IDIBOJPN(J4. + SYS1.PARMLIB.FAE1.USER(IDICNF00)
BOOKS),IDIBOKOR(K4.BOOKS),IDIDOC(A4.DOC),ID +
IDOJPN(J4.DOC),IDIDOKOR(K4.DOC),IDIDSECT(DA +
.DSECTS),IDIEXEC(DA.EXEC,DA.EXIT,EXEC,DA.T +
EST.CLIST),IDIHIST(DA.DCAT),IDILC(DA.LISTIN +
G.C),IDILCOB(DA.LISTING.COBOL),IDILPLI(DA.L +
ISTING.PLI),IDIMAPS(DA.SIDIMAPS),IDIVIEWS(& +
TSOPFX.VIEWS,DA.VIEWS,CTEST.VIEWS),DIVSEN +
U(A4.IDIHVENU),DIVSJP(JPN),DIVSK +
OR(A4.IDIHKOR))

DEFERREDREPORT(CICS(FATASKS(1,20)),NOIMS,NO + Default
BATCH)

DETAIL(MEDIUM)                                Default

DUMPREGISTRATIONEXITS(CONTROL(REXX(SLIPCTL) + SYS1.PARMLIB.FAE1.USER(IDICNF00)
),NOTIFY(REXX(SLIPNFY)))

```

## ユーザー・ノートのリカバリー

[ユーザー・ノートの作成と管理 ページ 218](#) で説明されているように、対話式分析を実行しながら、任意の保管場所に対してユーザー・ノートを記録できます。対話式再分析が正常に終了すると、ユーザー・ノートは履歴・ファイル障害項目に保管されます。対話式再分析が正常に終了しない場合は、ユーザー・ノートのリカバリーを有効にしない限り、セッション中に追加、削除、変更したユーザー・ノートは失われます。

ユーザー・ノートのリカバリーが有効な場合、指定されたデータ・セット内のすべてのユーザー・ノートのアクティビティが Fault Analyzer によって記録されます。対話式再分析セッションが異常終了すると、同一の履歴・ファイル障害項目またはシステム・ダンプを再分析する際に、Fault Analyzer によってユーザー・ノートのアクティビティが自動的に適用されます。

ユーザー・ノートのリカバリーを有効にするには、アクション・バーのプルダウン・メニューから **Options->Interactive Reanalysis Options** を選択して、**User Notes Recovery Data set name** フィールドに有効なデータ・セット名を指定します。(対話式再分析オプションの詳細については、[対話式再分析オプション ページ 162](#) を参照してください。)

ユーザー・ノートのリカバリーに使用するデータ・セットに、保存されていないユーザー・ノートのアクティビティに関する情報が含まれている場合、Fault Analyzer ISPF インターフェースには起動時にユーザー・ノートのリカバリー画面が表示されます。[図 80: ユーザー・ノートのリカバリー画面 ページ 153](#) に、この画面の例が表示されています。



図 80. ユーザー・ノートのリカバリー画面

```

File  View  Services  Help
-----
User Notes Recovery                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR

Line commands:
  S Perform reanalysis to recover unsaved user notes
  D Discard unsaved user notes
  PF3 to skip further user notes recovery at this time.

Dump or History File DSN                          Fault ID Status
---
PMR.T00541.S0045.HIST                            IM09293
EXAMPLE.HIST                                       F02489
EXAMPLE.HIST                                       F02491
EXAMPLE.HIST                                       F02492
EXAMPLE.HIST                                       F02493 Fault entry no
                                                    longer exists
EXAMPLE.HIST                                       F02494
EXAMPLE.SYSMDUMP
EXAMPL2.SYSMDUMP                                  No READ access

*** Bottom of data.

```

画面の各項目は、保存されていないユーザー・ノートを有する単一の対話式再分析セッションです。

- ユーザー・ノートのリカバリーするには、S 行コマンド (または SS ブロック行コマンド) を使用して項目を選択し、Enter キーを押します。
- ユーザー・ノートを破棄するには、D 行コマンド (または DD ブロック行コマンド) を使用します。

単一の項目に対して S 行または D 行コマンドを発すると、その項目は画面に表示されなくなります。

S 行または D 行コマンドを使用せずにこの画面から項目を選択する場合 (例えば、PF3 を押す場合) でも、同一のヒストリー・ファイル障害項目またはシステム・ダンプの対話式再分析を再度実行すると、ユーザー・ノートは自動的にリカバリーされます。この場合、「ユーザー・ノートのリカバリー」画面には、Fault Analyzer ISPF インターフェースの起動時に一致するエントリが表示されなくなります。

## 第 4 章. バッチ再分析の実行

バッチ再分析は、以下の操作のために使用します。

- 多数の障害に対して分析を再実行します。
- TSO セッションを中断するのではなくバッチで再分析を実行します。

バッチ・レポートは、バッチ・ジョブ・ステップの SYSPRINT DD ステートメントに送られます。

バッチ再分析レポートは、リアルタイム再分析レポートと同様の形をとります。詳しくは、[Fault Analyzer レポート ページ 275](#)を参照してください。

### バッチ再分析オプション

インストールに対して指定される汎用オプションの多くはバッチ再分析にも適用できます。使用可能なすべてのオプションと、そのオプションを指定できるさまざまな方法については、[オプション ページ 567](#) を参照してください。

バッチ再分析レポートを制御するために使用を検討すべきオプションをいくつか以下に示します。

#### Detail

このオプションは、バッチ再分析レポートに含める詳細のレベルを調整する場合に指定します。詳細については、[Detail ページ 586](#) を参照してください。

#### PrintInactiveCOBOL

PrintInactiveCOBOL オプションを使用すれば、非アクティブ COBOL プログラム (現行保管域チェーンにないプログラム) のストレージを再分析レポートに組み込むように要求できます。

バッチ・ジョブのみに適用されるバッチ再分析オプションを指定するには、**Fault Entry List** 画面の **Options** メニューから、**Batch Reanalysis Options** を選択します (メニュー・オプションの一般的な情報については、[アクション・バーのプルダウン・メニュー ページ 100](#) を参照してください)。これで、[図 81 : Batch Reanalysis Options 画面の例 ページ 155](#) に示されている **Batch Reanalysis Options** 画面が開きます。

図 81. Batch Reanalysis Options 画面の例

```

File  View  Services  Help
-----
Batch Reanalysis Options                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Press PF3 to save options or PF12 to cancel.

General Options:
Options line for batch
reanalysis. . . . . : Detail(Long)
Reanalysis report
destination . . . . . :
Redisplay this panel
before each reanalysis. : N (Y/N)
Display panel to edit
generated JCL . . . . . : N (Y/N)
Job card style. . . . . : P (P=Parameters, S=Statements)

Job Card Parameters:
Job name suffix . . . . . : A (A-Z, 0-9, @, #, or $)
Job class . . . . . : A (A-Z or 0-9)
Job notify. . . . . : Y (Y/N)
Job time minutes. . . . . : 10 (0-99)
Message class . . . . . : X (A-Z or 0-9)
Region megabytes. . . . . : 0 (0-2047)
Accounting info . . . . . :

Reanalysis Options Data Set Control:
Options data set name . . :
Options member name . . . : (If PDS or PDSE)
Use this data set during
reanalysis. . . . . : N (Y/N)
Edit the options data set
before reanalysis . . . : N (Y/N)

*** Bottom of data.
F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right   F12=Cancel

```

この画面を使用すれば、次のオプションを指定できます。

#### Options line for batch reanalysis

このフィールドで、ユーザーが実行するすべてのバッチ再分析ジョブに適用されるオプションを指定できます。ここで指定するオプションは、生成されるバッチ再分析ジョブの PARM フィールドで使用され、オプション・ファイルを使用して指定されるすべてのオプションに優先します (下記の「[Options data set name]」を参照してください)。

図 81 : Batch Reanalysis Options 画面の例 ページ 155 の Options Line には、例として Detail(Long) オプションが表示されています。

この行に収まるものより多くのオプションを指定する必要がある場合は、代わりに「[Display panel to edit generated JCL]」または「[Edit the options data set before reanalysis]」オプションを使用してください (どちらのオプションも以下に説明されています)。

オプション行で指定されるオプションは、ユーザー・プロファイルに保管されます。

## 再分析レポートの宛先

再分析レポートの宛先の指定 (オプション) です。指定されたオプションは、レポート用に生成された SYSPRINT DD ステートメントの DEST パラメーターに追加されるため、正確でなければなりません。

### Redisplay this panel before each reanalysis

このオプションが「Y」に設定される場合、JCL ストリームの生成前に、バッチ再分析が要求されるたびに、Batch Reanalysis Options 画面が表示されます。

変更を行います。その後で PF3 (現行オプションを使用して処理を続行する場合) または PF12 (変更した内容を取り消す場合) を押します。次に表示されるものは、下記の「[Display panel to edit generated JCL]」オプションにより決まります。

このオプションが「N」に設定される場合、Batch Reanalysis Options 画面は表示されません。

### Display panel to edit generated JCL

このオプションを「Y」に設定すると、[図 82: バッチ再分析 JCL ストリーム EDIT の例 ページ 156](#) の例のように、Fault Analyzer によって生成される疑似 JCL ストリームの ISPF EDIT 画面が表示されます。

図 82. バッチ再分析 JCL ストリーム EDIT の例

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          IBMUSER.SPFTEMP1.CNTL                      Columns 00001 00072
Command ==>  ----- Scroll ==> PAGE
***** ***** Top of Data *****
000001 //IBMUSERA JOB (),'FA - IDIVPCOB',
000002 // CLASS=A,MSGCLASS=X,TIME=10,NOTIFY=NWILKES,REGION=64M
000003 //RUNDA      EXEC PGM=IDIDA,
000004 // PARM=('/FAULTID(F16263)',
000005 //           )
000006 //STEPLIB DD DISP=SHR,DSN=CEE.SCEERUN
000007 //IDIHIST DD DISP=SHR,DSN=IDI.HIST
000008 //IDILCOB DD DISP=SHR,DSN=IBMUSER.IVPCB.LISTINGS
000009 //SYSPRINT DD SYSOUT=*
***** ***** Bottom of Data *****

```

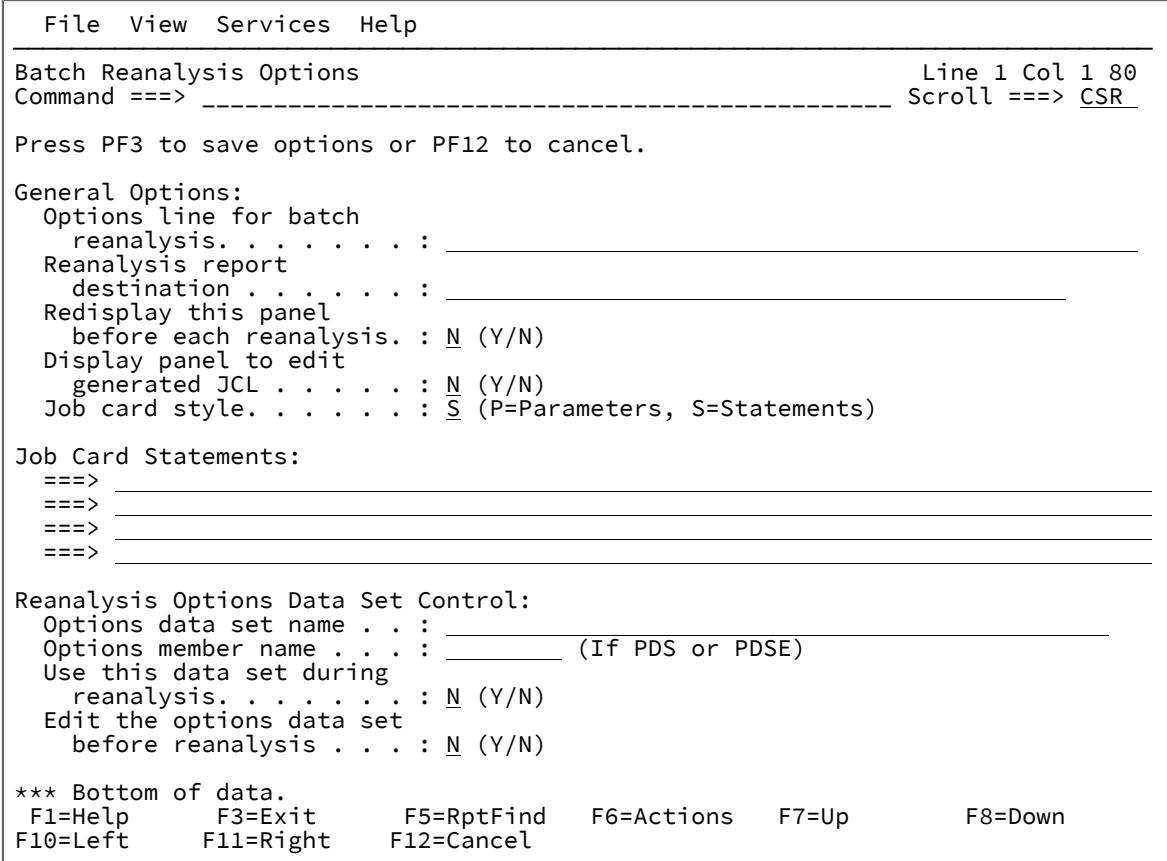
変更を行います。次に SUBMIT コマンドを発行してジョブを実行します。その後で EXIT (PF3) コマンドまたは CANCEL (PF12) コマンドを入力して Fault Entry List 画面に戻ります。

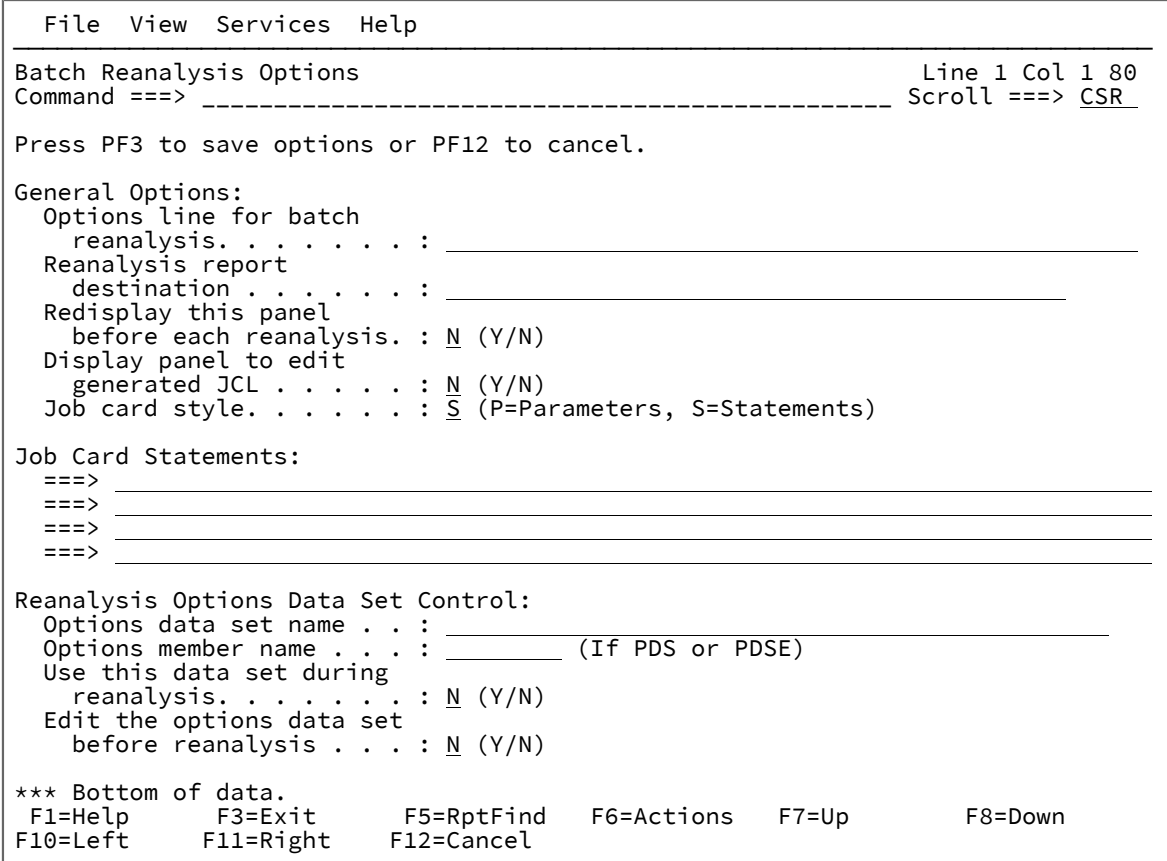
このオプションの使用について詳しくは、[バッチ再分析に使用されるデータ・セット ページ 159](#)を参照してください。

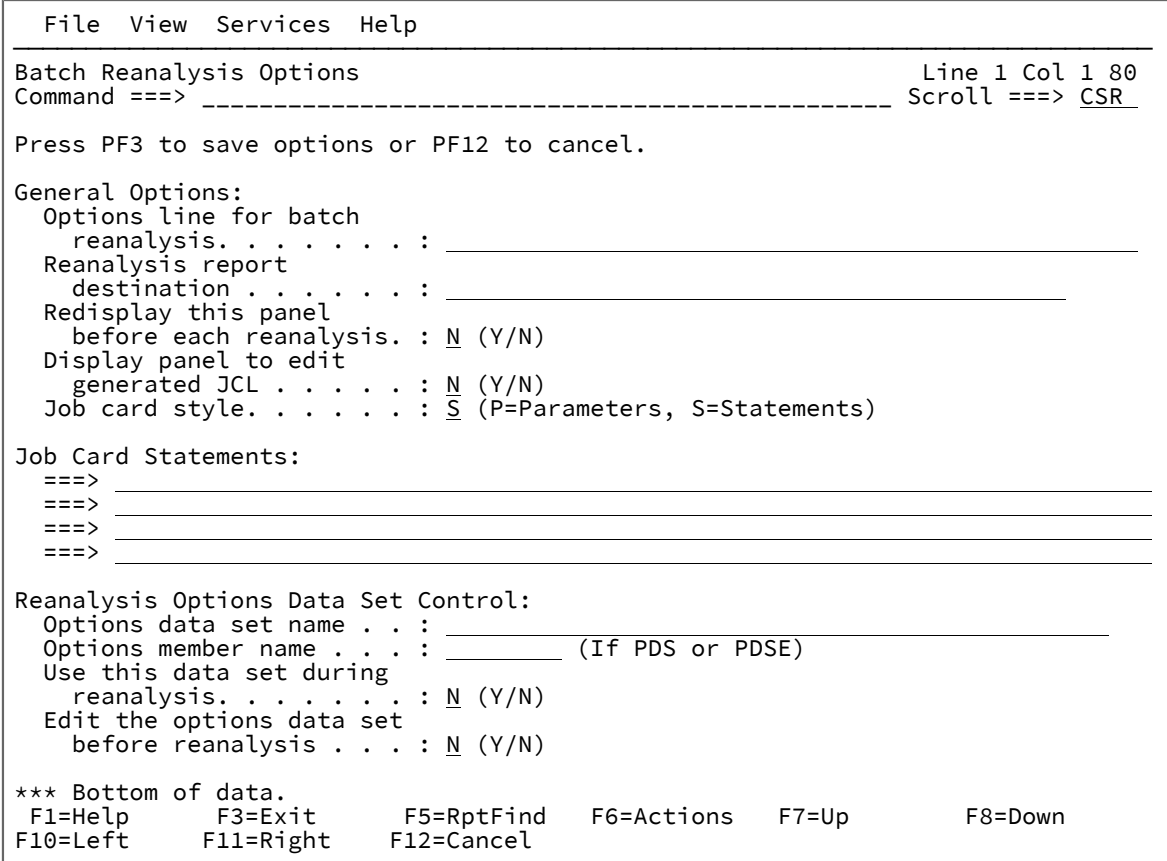
このオプションが「N」に設定される場合、生成される JCL ストリームは、JCL EDIT 画面が最初に表示されずに自動的に実行されます。

### ジョブ・カードのスタイル

次のように、単一文字 (P または S) でジョブ・カード指定のスタイルを制御します。

- 「P」(デフォルト)を指定し Enter キーを押すと、General Options セクションの後に  81 : Batch Reanalysis Options 画面の例 ページ 155 に示すように Job Card Parameters セクションが表示されます。
- 「S」を指定し Enter キーを押すと、General Options セクションの後に Job Card Statements セクションが表示されます。

これにより画面は、 81 : Batch Reanalysis Options 画面の例 ページ 155 に示されたようなものから、次のようなもの変わります。

 83. Batch Reanalysis Options 画面の例

```

File  View  Services  Help
-----
Batch Reanalysis Options                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR

Press PF3 to save options or PF12 to cancel.

General Options:
Options line for batch
  reanalysis. . . . . : _____
Reanalysis report
  destination . . . . . : _____
Redisplay this panel
  before each reanalysis. : N (Y/N)
Display panel to edit
  generated JCL . . . . . : N (Y/N)
Job card style. . . . . : S (P=Parameters, S=Statements)

Job Card Statements:
====> _____
====> _____
====> _____
====> _____

Reanalysis Options Data Set Control:
Options data set name . . : _____
Options member name . . . : _____ (If PDS or PDSE)
Use this data set during
  reanalysis. . . . . : N (Y/N)
Edit the options data set
  before reanalysis . . . : N (Y/N)

*** Bottom of data.
F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right    F12=Cancel

```

このオプションで選択したジョブ・カード仕様のスタイルが、いずれを選択した場合にも、バッチ再分析ジョブ用のジョブ・カードを生成する際に使用するメソッドを決定することになります。

#### Job name suffix

この接尾部は、生成されるバッチ再分析 JCL ストリームに使用されるジョブ名を作成するためにユーザー ID に付加される文字です。デフォルトは A です。

ユーザー ID の長さが 8 文字の場合は、サフィックスが追加される前に 7 文字に切り捨てられます。

#### Job class

このジョブ・クラスは、生成される JOB カードの CLASS パラメーターで使用されるジョブ・クラスです。デフォルトは A です。

**Job notify**

このフィールドが Y に設定される場合、NOTIFY=*userid* パラメーターが、生成される JOB カードに追加されます。N に設定されると、NOTIFY パラメーターは追加されません。デフォルトは Y です。

**Job time minutes**

この値は、JOB カードの TIME パラメーターで使用される分数です。有効な値の範囲は、1 から 30 です。デフォルトは 10 です。

**Message class**

このクラスは、生成される JOB カードの MSGCLASS パラメーターで使用されるメッセージ・クラスです。デフォルトは X です。

**Region megabytes**

この値は、生成される JOB カードの REGION パラメーターで使用される値です。有効な範囲は 0 から 2047 までです。デフォルトは 0 です。

**Accounting info**

このフィールドに指定されるものはすべて、生成される JOB カードでアカウントリング情報として使用されます。デフォルトでは、アカウントリング情報を提供しません。

**Options data set name**

このフィールドでは、オプションとして、メンバー ("*Options member name*" を参照) に Fault Analyzer オプションが含まれている PDS または PDSE データ・セットの名前を指定できます。このデータ・セットとメンバーの名前は、IDIOPTS ユーザー・オプション・ファイルとして使用されます。例えば、この画面の一番上にあるオプション行に収まるものより多くのオプションが必要な場合、このデータ・セットを使用できます。

**注:**

1. オプション・データ・セットが使用されるのは、“ [Use *this data set during reanalysis*] ”オプションが [Y] に設定される場合のみです。
2. オプション行で指定されるオプションは、このデータ・セットで指定されるオプションより優先されます。

**Options member name**

この名前は、“ [Options *data set name*] ”で指定されるデータ・セットのメンバー名です。

**Use this data set during reanalysis**

このオプションが [Y] に設定される場合、上記で指定されるデータ・セットとメンバーの名前が、対話式再分析時に Fault Analyzer により使用されます。このオプションが [N] に設定される場合は、データ・セットとメンバーの名前は使用されません。

### Edit the options data set before reanalysis

このフィールドが「Y」に設定される場合、バッチ再分析 JCL ストリームの生成前に、上記で指定されたオプション・データ・セット内のメンバーの ISPF EDIT 画面が表示されます。図 84: バッチ再分析用のオプション・ファイル EDIT の例 ページ 159 に例を示します。

図 84. バッチ再分析用のオプション・ファイル EDIT の例

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT MY.OPTIONS(SAMPCNF) - 01.02 Columns 00001 00072
Command ==> _____ Scroll ==> PAGE
***** Top of Data *****
000001 detail(1)
***** Bottom of Data *****

F1=Help      F2=Split    F3=Exit     F4=Return   F5=Rfind    F6=Rchange
F7=Up        F8=Down     F9=Swap     F10=Left    F11=Right   F12=Cancel

```

オプション・データ・セットを変更します (必要な場合)。次に EXIT コマンド (通常は PF3 にマップされている) を入力します。

## バッチ再分析の開始

バッチ再分析を開始するには、障害ヒストリー項目に対して **B** を入力します。

指定したバッチ・オプションに応じて、Fault Analyzer により生成される JCL ストリームの実行依頼の前に、1 つ以上の画面が表示される場合があります。詳しくは、[バッチ再分析オプション ページ 154](#)を参照してください。

## バッチ再分析に使用されるデータ・セット

ISPF インターフェースを使用してバッチ再分析を実行する場合、生成される JCL には、任意の JOBLIB、STEPLIB、Fault Analyzer コンパイラ・リストまたはサイド・ファイルのデータ・セットに必要な DD ステートメントが含まれます。Fault Analyzer データ・セット用の DD ステートメントは、リアルタイム JCL に明示的に含まれているのではなく、DataSets オプションまたは Analysis Control ユーザー出口を使用して指定されている場合であっても、追加されます。これらのデータ・セットは、リアルタイムで使用されたのと同じ実行環境を再作成するために、再分析ジョブに追加されます。

\_IDI\_OPTS ユーザー・オプション・ファイルまたは PARM フィールドを使用して指定された DataSets オプションにより、これらのデータ・セットは、リアルタイム実行からのデータ・セットに論理的に連結されます。

Batch Reanalysis Options 画面の「[Display panel to edit generated JCL]」オプションが「Y」に設定される場合 (バッチ再分析オプション ページ 154 を参照)、再分析ジョブを実行する前に、リアルタイム・データ・セット指定に変更を加えることが可能です。また、リアルタイムで使用されたにもかかわらず、再分析環境に存在しないデータ・セットや、READ アクセスが禁止されているデータ・セットはすべて、次の IDILCOB の例に表示されているように、コメントにより識別されません。

```
//IDILCOB DD DISP=SHR,DSN=CTEST.DUMPA.LISTING.CICCS.COBOL
//          DD DISP=SHR,DSN=DA.LISTING.COBOL
//* The following IDILCOB data set is unavailable:
//*          DD DISP=SHR,DSN=CTEST.DUMPA.LISTING.CICCS.COBOSVS
//* The following IDILCOB data set is READ protected:
//*          DD DISP=SHR,DSN=CTEST.PROTECT.LISTINGS
```



**注:** GenerateSavedReport オプションが有効な場合、リアルタイム処理中に使用されるすべてのデータ・セットまたはパス名は、再分析の実行前に生成済み JCL から削除された場合でも、自動的に含まれます。

## 独自のバッチ再分析ジョブの作成

Fault Analyzer によって以前に生成されたジョブから、ユーザーが作成または保管し、必要に応じて後で変更した JCL を介して、バッチ・モードを起動することもできます。障害項目の再分析を実行する場合は、FaultID オプションを指定します。MVS ダンプ・データ・セットの分析を実行する場合は、DumpDSN オプションを指定します。

JCL は、Fault Entry List 画面から B 行コマンドを使用して生成できます。この方法では、サイド・ファイル・データ・セットの JCL DD ステートメントとリアルタイム処理中に有効であったすべての STEPLIB 連結が自動的に追加されます。このため、この方法はバッチ再分析を実行する場合に推奨されます。

例えば、ジョブはこのようになります。



図 85. サンプル・バッチ再分析ジョブ

```

//RTURNERA JOB (), 'FAULT ANALYZER', CLASS=A, MSGCLASS=X, NOTIFY=&SYSUID
//*
//* Allocate a PDSE for compiler listings
//*
//ALLOC EXEC PGM=IEFBR14
//DD1 DD DISP=(,CATLG), DSN=&SYSUID..COBLIST, SPACE=(CYL,(1,1,5)),
// DCB=(RECFM=FBA, LRECL=133), DSNTYPE=LIBRARY
//*
//* Recompile MYPGMA
//*
//CBLCOMP EXEC IGYWC, PARM.COBOL='LIST,MAP,Source,XREF'
//COBOL.SYSIN DD DISP=SHR, DSN=MY.COBOL.Source(MYPGMA)
//COBOL.SYSPRINT DD DISP=SHR, DSN=&SYSUID..COBLIST(MYPGMA)
//*
//* Recompile MYPGMB
//*
//CBLCOMP EXEC IGYWC, PARM.COBOL='LIST,MAP,Source,XREF'
//COBOL.SYSIN DD DISP=SHR, DSN=MY.COBOL.Source(MYPGMB)
//COBOL.SYSPRINT DD DISP=SHR, DSN=&SYSUID..COBLIST(MYPGMB)
//*
//* Reanalyze SYSMDUMP data set
//*
//RUNDA EXEC PGM=IDIDA, PARM=(' /DumpDSN(MY.DUMPDS) ')
//SYSPRINT DD SYSOUT=*
//IDILCOB DD DISP=SHR, DSN=&SYSUID..COBLIST
//*
//* Delete temporary compiler listings PDSE
//*
//DELETE EXEC PGM=IEFBR14
//DD1 DD DISP=(OLD,DELETE), DSN=&SYSUID..COBLIST

```

このジョブには、2つの COBOL プログラム、MYPGMA と MYPGMB の再コンパイルが含まれています。これらは再分析中の障害に関連すると想定されますが、このプログラムのコンパイラ・リストは、リアルタイム分析中に Fault Analyzer に対して使用不可だった可能性があります。これらを障害再分析に指定することにより、エラーが発生したソース・コードの行を識別できます。

各ジョブ・ステップでコンパイルされるプログラムは1つのみであるということに注意してください。この制限事項は、[コンパイラ・リストまたはサイド・ファイルの命名 ページ 382](#)で概説されている規則に基づいたコンパイラ・リストのデータ・セット・メンバーの命名を容易にするためのものです。

オプションとして、IDIOPTS DD ステートメントを JCL に追加できます。このステートメントは、Fault Analyzer オプションを含む順次ファイルの名前を提供し、ジョブ・ステップに対して製品およびインストール・システムのデフォルトの変更を提供します。

JCL に追加できるその他の DD ステートメントについては、[JCL DD ステートメントによるリストのポインティング ページ 36](#)を参照してください。

JCL EXEC ステートメントの PARM フィールドに指定されたオプションは、IDIOPTS ファイルにより設定されたオプションをオーバーライドします。

詳しくは、[User options file IDIOPTS ページ 571](#)を参照してください。

## 第 5 章. 対話式再分析の実行

対話式再分析は、次のように、バッチ再分析より優れた利点がいくつかあります。

- 興味のあるレポートのセクションを選択し、別々に調べることができる。
- 関連するミニダンプまたは MVS ダンプ・データ・セットに含まれているすべてのストレージ域を、そのストレージ域が Fault Analyzer レポートに含まれているかどうかにかかわらず表示できます。
- ソース・コード情報 (コンパイラ・リストまたはサイド・ファイルにより提供される場合) 全体を表示できる。
- この方法が、CICS® システムの異常終了を分析する唯一の方法です。



**注:** 本章の情報は、対話式再分析を ISPF で実行することを想定して書かれていますが、これは CICS® で実行することもできます。その場合は、いくつかの制約事項が適用されることがあります。これらの制約事項は、[以下の環境での対話式再分析の実行 CICS ページ 286](#)で説明します。

### 対話式再分析オプション

インストール済み環境用に指定されている汎用オプションの多くは、対話式再分析にも適用できます。使用可能なすべてのオプションと、そのオプションを指定できる様々な方法については、[オプション ページ 567](#)を参照してください。

対話式再分析セッションにのみ適用される対話式再分析オプションを指定するには、**Fault Entry List** 画面の **Options** メニューから、**Interactive Reanalysis Options** を選択します (メニュー・オプションの一般的な情報については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。このオプションを選択すると、[図 86 : Interactive Reanalysis Options 画面の例 ページ 163](#)に示すように、Interactive Reanalysis Options 画面が開きます。

図 86. Interactive Reanalysis Options 画面の例

```

File View Services Help
-----
Interactive Reanalysis Options                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Press PF3 to save options or PF12 to cancel.

General Options:
Options line for
  interactive reanalysis. . . : GSR(O)
Redisplay this panel
  before each reanalysis. . . : N (Y/N)
Display panel to alter
  allocated data sets . . . : N (Y/N)
Prompt before opening an
  MVS dump data set . . . . : Y (Y/N)
Always prompt to select
  TCB for MVS dump data
  set analysis. . . . . : N (Y/N)
Prompt for missing side
  files . . . . . : Y (Y/N)
Current list of excluded programs ( Edit ):
  (Empty)

Reanalysis Options Data Set Control:
Options data set name . . : JCLLIB
Options member name . . . : IDICNF0@ (If PDS or PDSE)
Use this data set during
  reanalysis. . . . . : N (Y/N)
Edit the options data set
  before reanalysis . . . . : N (Y/N)

Deferred Breakpoints Repository:
Data set name (PDS/E) . . : PRINT.PDS

User Notes Recovery:
Data set name . . . . . : 'EXAMPLE.NOTES.RECOVERY'

*** Bottom of data.

```

以下の実現性をこの画面で指定できます。

#### Options line for interactive reanalysis

開始するすべての対話式再分析セッションに適用するオプションをここで指定できます。これらのオプションは、オプション・ファイルで指定されるどのオプションよりも優先されます（「[Options data set name ページ 167](#)」を参照）。これらのオプションは、バッチ再分析ジョブで使用される **PARM** フィールド・オプションと同等です。

図 86 : Interactive Reanalysis Options 画面の例 ページ 163 の Options Line には、例として `GSR(O)` オプションが表示されています。

#### Redisplay this panel before each reanalysis

このオプションを「Y」に設定すると、対話式再分析が要求されるたびに Interactive Reanalysis Options 画面が表示されます。

必要な変更を加えてください。次に、PF3（現行オプションを使用して続行）または PF12（すべての変更を取り消す）を押します。次に表示されるものは、**Display panel to alter allocated data sets** オプションに応じて異なります。

このオプションを「N」に設定すると、Interactive Reanalysis Options 画面は表示されません。

### Display panel to alter allocated data sets

このオプションを「Y」に設定すると、Fault Analyzer によって生成される疑似 JCL ストリームの ISPF EDIT 画面が表示されます。

図 87. 対話式再分析疑似 JCL ストリーム EDIT の例

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          IBMUSER.SPFTEMP1.CNTL                      Columns 00001 00072
Command ==> ----- Scroll ==> PAGE
***** ***** Top of Data *****
===== Type "RESET" on the command line and press Enter to see edit
===== instructions.
- - - - - 50 Line(s) not Displayed
000051 //IDIADATA DD DISP=SHR,DSN=DA.SYSADATA
000052 //IDILCOB DD DISP=SHR,DSN=DA.LISTING.COBOL
000053 //* The following IDILCOB data set is unavailable:
000054 //* DD DISP=SHR,DSN=DA.LISTING.COBOL.OLD
000055 //IDIJAVA DD PATH='/u/temp/payroll/directory171/DEPT64directory/accoun
000056 //          tingDIR1'
000057 //          DD PATH='/u/temp/payroll/directory171/DEPT64directory/accoun
000058 //          tingDIR2'
000059 //* The following IDIJAVA HFS path is unavailable:
000060 //* DD PATH='/u/temp/payroll/directory171/DEPT64directory/accoun
000061 //*          tingDIR3'
000062 //          DD PATH='/u/a+'
000063 //IDILC DD DISP=SHR,DSN=DA.LISTING.C
000064 //IDILPLI DD DISP=SHR,DSN=DA.LISTING.PLI.PROD
000065 //          DD DISP=SHR,DSN=DA.LISTING.PLI.TEST
***** ***** Bottom of Data *****

```

コマンド行に「RESET」を入力して、Enter

キーを押します。表示された指示に従って変更を行います。次に、必要に応じて EXIT (PF3) コマンドまたは CANCEL (PF12) コマンドを入力し、対話式再分析を開始します。

このオプションについて詳しくは、[対話式再分析に使用されるデータ・セット ページ 244](#)を参照してください。

このオプションを「N」に設定すると、疑似 JCL EDIT

画面が最初に表示されることなく、対話式再分析が開始されます。

### MVS ダンプ・データ・セットを開く前にプロンプトを表示

以下のような場合には、保管されたミニダンプには含まれていない保管場所へのアクセスが必要になることがあります。

- 対話式再分析中。
- 対話式レポート内から保管場所を表示した結果として。

この場合、フィールドが「Y」に設定されていると、関連する MVS ダンプ・データ・セットがオープンされる前に、欠落した保管場所を検索するための画面が表示されます。

図 88 : Confirm MVS Dump Open 画面の例 ページ 165 に、この画面の例が表示されています。

図 88. Confirm MVS Dump Open 画面の例

```

File  Options  View  Services  Help
Confirm MVS Dump Open
Command ==> -----

Fault Analyzer has determined the need to open an MVS DUMP data set:
JKATNIC.CICS53.SOS.DUMP

Permitting this might cause delays, however, if the open is not permitted,
Fault Analyzer cannot access important storage information.

Press Enter to confirm the data set open.

Press Cancel or Exit to cancel the data set open and attempt to continue
without access to missing storage locations.

F1=Help      F3=Exit      F12=Cancel

F00286 CICS53  n/a      MVS2      S122  2019/05/22 10:49:44
F00325 DAAMB022 n/a      MVS2      S0C6  2019/04/27 11:03:48
F00111 CICS53  n/a      MVS2      S08E  2019/03/22 13:12:23
F00272 CICS53  n/a      MVS2      S08E  2019/03/22 13:12:23
F00328 CICS53  n/a      MVS2      S08E  2019/03/22 13:12:23
F1=Help      F3=Exit      F4=MatchCSR  F5=RptFind  F6=Actions  F7=Up
F8=Down      F10=Left     F11=Right   F12=MatchALL

```

対話式再分析セッション中、プロンプトが出されるのは1回のみです。CANCEL または EXIT

を入力してオープンを取り消すと、MVS ダン

プ・データ・セットのオープンはそれ以上試行されません。同様に、オープンを許可すると、Fault Analyzer はミニダンプに含まれていない保管場所へのすべての参照について MVS ダンプを調べます。

このフィールドを「N」に設定すると、最初にプロンプトが出されることなく、必要に応じて関連ダンプ・データ・セットがオープンされます。

#### Always prompt to select TCB for MVS dump data set analysis

MVS ダンプ・データ・セットの分析を実行する前に、TCB 選択画面を表示するかどうかを制御する単一文字（「Y」または「N」）。

このオプションを「N」に設定すると、Fault Analyzer が TCB を選択できない場合にのみ TCB 選択画面が表示されます。この状態は、例えば異常終了したタスクがない場合にコンソール・ダンプで発生することがあります。

このオプションを「Y」に設定すると、TCB 選択画面は常に表示されます。Fault Analyzer が TCB を選択できる場合、この TCB は画面上で事前に選択されます。この TCB を受け入れるか、別の TCB に変更し、分析を実行するには Enter を押します。

#### Prompt for missing side files

プログラムのコンパイラー・リストまたはサイド・ファイルが見つからない場合に、Compiler Listing Not Found 画面を表示するかどうかを制御する単一文字（Y または N）。

このフィールドのデフォルト設定は、DataSets オプションの指定が、次のリストの DDname のいずれかに対する IDICNFxx parmlib メンバー内に存在するかどうかに基づきます。

IDIADATA  
 IDILANGX  
 IDILC  
 IDILCOB  
 IDILCOBO  
 IDILPLI  
 IDILPLIE  
 IDISYSDB

これら DDname のいずれかに対するデータ・セットが IDICNFxx parmlib  
 メンバー内で指定されていない場合は、このフィールドのデフォルト設定は「N」です。それ以外の場合は  
 「Y」です。

このフィールドをクリアして Enter キーを押すと、値はデフォルトの設定値に再度初期化されます。

### Current list of excluded programs (Edit)

「Prompt for missing side files」オプションを「Y」に設定すると、除外対象プログラム名  
 (コンパイラー・リストまたはサイド・ファイルの検索を実行しないプログラム名)  
 の現行リストが表示されます。「Edit」ポイント・アンド・シュート・フィールドにカーソルを置いて Enter  
 を押せば、リストを変更できます。これにより、Excluded Program Names 画面が表示されます。図 89：  
 「Excluded Program Names」画面 ページ 166 に、この画面の例が表示されています。

図 89. 「Excluded Program Names」画面

```

Excluded Program Names                               Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR

Enter D on the line to delete a program.
Type ADD followed by a list of program names to add them to the list.

Program  Source
ZERR     GLOBAL
PROG11B  GLOBAL
FRED     USER
COBBL    USER

* * Bottom of data.

F1=Help   F3=Exit   F5=RptFind  F7=Up      F8=Down    F12=Cancel

```

以下の 2 つのプログラム除外リストがあり、これらは画面に実際に表示される結合されたリストを導きます。

- Fault Analyzer ISPF インターフェイスで作成され、ユーザーの ISPF  
プロファイルに保管されたユーザー指定リスト。
- IDICNFxx parmlib メンバーの FAISPFOPPTS(GlobalExclude(...))  
オプションで指定されたグローバル・リスト。

特殊なプログラム名「-DROPCNF-」をユーザー指定リストに組み込むことにより、ユーザーは存在する可能性のあるグローバル・プログラム名をすべて除外できます。

PF3 を押して画面を終了し、行った変更をすべて保管します。

コマンド行に ADD

と入力し、その後続けて、ユーザー指定の除外として追加するブランクで区切られたプログラム名を 1 つ以上入力して、Enter

を押します。必要な命名規則にプログラム名が準拠していない場合は、メッセージが出されます。

プログラム名は、名前をより汎用的なものにするために、オプションでワイルドカード文字「\*」（ゼロ個以上の文字）および「%」（単一の必須文字）を追加した、有効な PDS または PDSE メンバー名でなければなりません。指定するプログラム名の大/小文字は区別されません。

有効なプログラム名の指定例を以下に示します。

```
*XMAI*
PAYROLL0
SELOPT%
SUBRTN*
TZ%%C*
```

D 行コマンドを使用して、個々の項目をユーザー指定リストから削除します。この画面で、個々の項目をグローバル・リストから削除することはできません。

プログラム名除外リストは、Compiler Listing Not Found 画面から編集することもできます。詳しくは、[コンパイラー・リストまたはサイド・ファイルのプロンプト ページ 239](#)を参照してください。

### Options data set name

このフィールドでは、オプションとして、メンバー（“*Options member name*”を参照）に Fault Analyzer オプションが含まれている PDS または PDSE データ・セットの名前を指定できます。このデータ・セットとメンバーの名前は、IDIOPTS ユーザー・オプション・ファイルとして使用されます。例えば、この画面の一番上にあるオプション行に収まるものより多くのオプションが必要な場合、このデータ・セットを使用できます。



注:



1. オプション・データ・セットが使用されるのは、“ [Use *this data set during reanalysis*] ”オプションが [Y] に設定される場合のみです。
2. オプション行で指定されるオプションは、このデータ・セットで指定されるオプションより優先されます。

**Options member name**

このフィールドは、“Options data set name” で指定されるデータ・セットのメンバー名です。

**Use this data set during reanalysis**

このオプションを [Y] に設定すると、以前に指定したデータ・セットとメンバーの名前が、対話式再分析時に Fault Analyzer で使用されます。このオプションが [N] に設定される場合は、データ・セットとメンバーの名前は使用されません。

**Edit the options data set before reanalysis**

このフィールドを [Y] に設定すると、対話式再分析の開始前に、上で指定されたオプション・データ・セットにあるメンバーの ISPF EDIT 画面が表示されます。図 90: 対話式再分析用のオプション・ファイル EDIT の例 ページ 168 に例を示します。

図 90. 対話式再分析用のオプション・ファイル EDIT の例

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          USER.JCLLIB(IDIOPTS) - 01.02          Columns 00001 00072
Command ==> ----- Scroll ==> PAGE
***** ***** Top of Data *****
000001 datasets(idiexec(user.exec))
000002 exits(listing(rexx(lx)))
***** ***** Bottom of Data *****

F1=Help      F2=Split    F3=Exit     F4=Return   F5=Rfind    F6=Rchange
F7=Up        F8=Down     F9=Swap     F10=Left    F11=Right   F12=Cancel
    
```

オプション・データ・セットを変更します (必要な場合)。次に EXIT コマンド (通常は PF3 にマップされている) を入力します。

**Data set name (PDS/E)**

COBOL Explorer によって設定された z/OS® Debugger Deferred Breakpoints に使用するデータ・セットの名前を指定します。



#### 標準の ISPF

データ・セット名指定規則が適用されます。つまり、データ・セット名が単一引用符で囲まれていない場合は、現行 TSO 接頭部が前に付けられます。

指定した名前のデータ・セットは、必要になったときに自動的に割り振られるため、存在している必要はありません。既存のデータ・セット名を指定する場合は、RECFM=VB および LRECL=255 で割り振られた PDS データ・セットの名前でなければなりません。

メンバー名を指定してはなりません。

#### ユーザー・ノートのリカバリーに使用するデータ・セット名

ユーザー・ノートのリカバリーに使用するデータ・セットの名前を指定します。

- 標準の ISPF データ・セット名の指定規則が適用されます:  
データ・セット名が単一引用符で囲まれていない場合は、現行の TSO 接頭部が前に付けられます。
- RECFM=VB および LRECL=1024  
で割り振られた順次データ・セットに限り、既存のデータ・セット名を指定できます。データ・セットが空でない場合、有効なユーザー・ノートのリカバリーに関する情報を含める必要があります。
- 指定されたデータ・セットが存在しない場合、[図 91: 「ユーザー・ノートのリカバリーに使用するデータ・セットの作成」画面 ページ 170](#) で示す例のような「ユーザー・ノートのリカバリーに使用するデータ・セットの作成」画面が表示されます。
- TSO/ISPF  
の各ユーザーは、固有のユーザー・ノートのリカバリーに使用するデータ・セットを利用する必要があります。ユーザーのノートに機密情報が含まれている可能性がある場合、適切な方法でデータ・セットへのアクセスを保護してください。

詳しくは、「[ユーザー・ノートのリカバリー ページ 152](#)」を参照してください。

図 91. 「ユーザー・ノートのリカバリーに使用するデータ・セットの作成」画面

```

File View Services Help
-----
User Notes Recovery Data Set Create                               Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR

The specified user notes recovery data set does not exist. Press Enter to
create, or press PF3/PF12 to cancel.

Data Set Name . . . . . : 'EXAMPLE.NOTES.RECOVERY'
Primary Space . . . . . : 5_____ Cylinders
Secondary Space . . . . . : 5_____ Cylinders

*** Bottom of data.

MVS dump data set . . . . . : Y (Y/N)
Always prompt to select
TCB for MVS dump data
set analysis. . . . . : N (Y/N)
Prompt for missing side
files . . . . . : Y (Y/N)
Current list of excluded programs ( Edit ):
(Empty)

Reanalysis Options Data Set Control:
Options data set name . . : JCLLIB
Options member name . . . : IDICNF00 (If PDS or PDSE)
Use this data set during
reanalysis. . . . . : N (Y/N)
Edit the options data set
before reanalysis . . . . : N (Y/N)

Deferred Breakpoints Repository:
Data set name (PDS/E) . . : PRINT.PDS

User Notes Recovery:
Data set name . . . . . : 'EXAMPLE.NOTES.RECOVERY'

*** Bottom of data.

```

必要に応じて、デフォルトの1次および2次スペース値を変更できます。

## 対話式再分析の開始

対話式再分析を開始するには、障害ヒストリー項目に対してIを入力します。

## 状況ポップアップ画面

対話式再分析レポートが表示される前に行われる対話式再分析中には、次の例のような状況ポップアップ画面が表示されることがあります。

図 92. Interactive Reanalysis Status 画面の例

```

File  Options  View  Services  Help
-----
IBM Fault Analyzer - Fault Entry List                               Line 1 Col 1 80
Command ==>                                                       Scroll ==> CSR

Fault History File or View : 'NWILKES.HIST'

{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).}

  Fault_ID Dups  Job/Tran User_ID  Sys/Job  Abend  Date      Time      MD_Pag
i  SW00341    RSHPAR2  B12128  SY2     S06F   2019/07/12 14:34:06
   SW00154    PG70688P PG7068  DEV     n/a    2019/07/07 13:00:56
   SW00137    PRGMN01  PST0123 SYSN    S0C4   2019/07/07 09:36:13
S
S      _____ Interactive Reanalysis Status _____
S
S      Current Function      : le_heap
S      Total Elapsed Time   : 11 Seconds
S      Total CPU Time       : 8.201 Seconds
S
F1=H   F2=Split  F3=Exit   F9=Swap   F12=Cancel
F8=D

```

この画面は、対話式再分析の開始から経過した時間が10秒を超えると表示され、それ以降、10秒経過するたびに更新されます。画面に示される現行機能は、Fault Analyzer 内部のプロセスを表しますが、関数 ID が変わらない場合は、ループ条件の標識として機能します。また、再分析が実行されたシステムで過負荷のために CPU リソースが不足している場合は、CPU 時間に対する経過時間が表示されるため、通常より分析に時間がかかる理由がわかります。

Interactive Reanalysis Status 画面では、ユーザー対話が許可されず、分析完了時にこの画面は自動的に除去されます。

## 対話式レポートについての一般的な解説

対話式再分析レポートは、リアルタイム障害分析レポートと類似していますが、問題の原因の詳細を参照するための機能が備わっています。

図 93 : [Interactive Reanalysis Report 画面の例](#) ページ 172 は、最初の対話式レポート画面の例を示しています。この画面から、対話式レポートの他のすべての部分を選択できます。

図 93. Interactive Reanalysis Report 画面の例

```

File View Services Help
-----
Interactive Reanalysis Report                               Line 1 Col 1 80
Command ==>                                               Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7                      MVS2      2019/12/21 13:02:25

Fault Summary:
Module IDISCBL1, program IDISCBL1, source line # 31 : Abend S0C7 (Data
Exception).

Select one of the following options to access further fault information:
 1. Synopsis
 2. Event Summary
 3. Open Files
 4. Storage Areas
 5. Messages
 6. Language Environment Heap Analysis
 7. Abend Job Information
 8. User Notes
 9. Fault Analyzer Options

{Fault Analyzer maximum storage allocated: 1.47 megabytes.}

*** Bottom of data.

```

初期表示画面の一番上に障害の要約が表示されます。この要約は、障害のリアルタイム分析時に出される [IDI0002I ページ 695](#) メッセージに表示される要約と同じです。

初期表示画面から選択できる個々のオプションについては、以下のセクションで説明しています。オプションは、分析する障害によって変わる可能性があります。選択可能なオプションは、コマンド行から入力することも、カーソルをオプション番号の上に置いて Enter キーを押すこともできます。

ヘルプ・テキストを表示するオプションが選択される場合 ([ヘルプ・テキストの追加または除去 ページ 127](#)を参照)、割り振られたストレージの中で Fault Analyzer が分析時に使用した最大量についての情報が、画面の一番下に表示されます。このストレージ量は明示的割り振りのみに関するものです。ロードされたモジュールなどのストレージは含まれていません。

対話式レポートは、使用される論理画面サイズに応じてフォーマットが異なります。本書の例はすべて、24 行 x 80 列の画面サイズに基づいています。ただし、ご使用の画面がこれより大きい場合、Fault Analyzer は適宜にレポートをフォーマットします。この状態は、画面サイズが動的に変更される場合にも当てはまります。Enter キーを押すだけで、表示されるレポート・セクションが、画面サイズに合わせて再フォーマットされます。

対話式レポート内のどこでも、UP (PF7)、DOWN (PF8)、LEFT (PF10)、または RIGHT (PF11) コマンドを使用して、現在選択されているレポート・セクション全体を表示できます。(Dump Storage 画面の PF10 および PF11 は、この画面が水平スクロールを必要としないため、それぞれ PREV および NEXT コマンドにマップされます。)

対話式レポート全体で、タブ・キーで移動できるフィールドは黄色で示されます。これらのフィールドは、ポイント・アンド・シュート方式のフィールドです。そのフィールドにカーソルを置いて、Enter キーを押すと応答します。次に表示される画面は、選択された情報のタイプによって異なります。レポートの他の部分に進む場合もあれば、選択された項目についての詳細情報を表示する場合があります。例:

#### ソース・コード行またはステートメントの番号

コンパイラー・リストまたはサイド・ファイルから取得された、プログラム全体のソース・コードを表示し、選択された行またはステートメントの番号が強調表示されます。さらに、マシン・インストラクションの逆ア

サンプルも提供されます。この画面の詳細および例については、[ソース・コードの表示 ページ 212](#) を参照してください。

### ストレージ・アドレス

16 進数と変換 EBCDIC

の両方で、この位置のストレージを表示します。この画面の詳細および例については、[保管場所の表示 ページ 215](#) を参照してください。

### プログラム状況ワード (PSW)

PSW は半分ずつ表示されます。

- 高位ワードを選択すると、PSW ビット設定に関する情報が表示されます。詳しくは、[PSW 情報の表示 ページ 225](#) を参照してください。
- 低位ワードを選択すると、PSW アドレスのストレージが表示されます。詳しくは、[保管場所の表示 ページ 215](#) を参照してください。

ポイント・アンド・シュート方式のフィールドは、ISPF カラー属性 YELLOW を使用して定義されますが、実際には、ユーザー設定値に応じて別の色で表示される場合があります。ただし、本書では、このフィールドを黄色のフィールドと呼びます。

ストレージ・アドレスは別として、すべてのポイント・アンド・シュート・フィールドは、画面のコマンド行からも入力できます。このポイント・アンド・シュート機能は特に、項目をオプションのリストから選択するときに便利です。

## 対話式レポートの終了

EXIT (PF3) または CANCEL コマンドを Interactive Reanalysis Report 画面で使用した対話式レポートの終了時、[図 94: Confirm Exit 画面の例 ページ 173](#) の例に示す確認プロンプトが表示されることがあります。

図 94. Confirm Exit 画面の例

```

File View Services Help
-----
Interactive Reanalysis Report                               Line 1 Col 1 80
Command ==>                                              Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7                     MVS2      2019/12/21 13:02:25

----- Confirm Exit -----
Are you sure you want to exit the interactive report ? Press Enter to
confirm exit or press PF12 to return to the interactive report.

F1=Help   F12=Cancel

3. Open Files
4. Storage Areas
5. Messages
6. Language Environment Heap Analysis
7. Abend Job Information
8. User Notes
9. Fault Analyzer Options

{Fault Analyzer maximum storage allocated: 1.47 megabytes.}

*** Bottom of data.

```

この画面は、例えば PF3 を 1 回以上押すことで表示され、対話式再分析レポートの予期しない終了を示します。終了を確認し、Fault Entry List 画面に戻るには、Enter キーを押してください。終了を中止し、対話式再分析レポートを再開するには、PF12 を押してください。

終了のプロンプト・パネルは、対話式再分析の経過時間が InteractiveExitPromptSeconds オプションの有効な秒数を超過した場合、またはそれと等しい場合のみ表示されます。このオプションの詳細については、[InteractiveExitPromptSeconds ページ 605](#)を参照してください。

ユーザー情報が変更されていると、追加のプロンプトが表示されることがあります。詳しくは、[リフレッシュ処理 ページ 244](#)を参照してください。

ISPF ジャンプ・コマンドを使用して、任意のタイミングで対話式レポートを終了できます (例えば、コマンド行で `EX` を入力し、Enter を押します)。

## 基本オプション: Synopsis

対話式レポートの初期表示画面から「Synopsis」オプションを選択すると、[図 95: Synopsis 画面の例 ページ 174](#)に示す例のように、レポートの「Synopsis」セクションが表示されます。

図 95. Synopsis 画面の例

```

File  View  Services  Help
-----
Synopsis                                         Line 1 Col 1 80
Command ==>>                                   Scroll ==>> CSR
JOBNAME: COBUNI      SYSTEM ABEND: 0C9          FAE1      2019/07/21  14:36:43

A system abend 0C9 occurred in module COBUNI program COBUNI at offset X'5C8'.

A program-interruption code 0009 (Fixed-Point-Divide Exception) is associated
with this abend and indicates that:

    The divisor was zero in a signed binary division.

The cause of the failure was program COBUNI in module COBUNI.  The COBOL source
code that immediately preceded the failure was:

Source
Line #
000036          IF UTF-16 NOT EQUAL ' ' AND V1 / V2 > 0

The COBOL source code for data fields involved in the failure:

Source
Line #
000018          01  UTF-16 PIC N(30) USAGE NATIONAL.
000020          01  V1 PIC 9(9) BINARY VALUE 357.
000021          01  V2 PIC 9(9) BINARY VALUE 0.

Data field values at time of abend:

UTF-16 = UTF-16 UNICODE DATA
V1      = 357
V2      = 0  *** Cause of error ***

*** Bottom of data.

```

## 基本オプション: イベント・サマリー

対話式レポートの初期表示画面から「Event Summary」オプションを選択すると、[図 96: Event Summary 画面の例 ページ 175](#)に示す例のように、レポートの「Event Summary」セクションが表示されます。

図 96. Event Summary 画面の例

```

File View Services Help
-----
Event Summary                               Line 1 Col 1 80
Command ==>                               Scroll ==> CSR
Full Application only - TRANID: FRED   CICS ABEND: AEIL   2019/06/13  10:42

{The following events are presented in chronological order.}

Event      Fail  Module  Program  EP
#  Type    Point  Name     Name     Name     Event Location (*)  Loaded
1  Call    DFHAPLI DFHAPLI1 n/a      P+27C4  CICS.C
2  Call    CEEPLPKA n/a      CEECRINI E+8B0   Not de
3  Call    CEEPLPKA n/a      CEECRINV E+42E   Not de
4  Call    CEEEV005 IGZCEV5 IGZCEV5  E+672   CEE.SC
5  EXEC CICS ***** CICFRED CICFRED  L#69 E+436  DA.TES
6  Abend AEIL DFHAIP  DFHEIP  n/a      P+1A92  CICS.C

(*) One or more of the following abbreviations might appear in the "Event
Location" column:

F#n  Source file number (refer to detailed event information for file
      identification)
L#n  Source file line number
S#n  Listing file statement number (refer to detailed event information
      for file identification)
M+x  Offset from start of load module
P+x  Offset from start of program
E+x  Offset from start of entry point

*** Bottom of data.
F1=Help   F3=Exit   F4=Dsect   F5=RptFind  F6=Actions  F7=Up
F8=Down   F10=Left  F11=Right

```

イベント番号にカーソルを置き Enter キーを押すと、このサマリーから個々のイベントを選択できます。この操作を行うと表示される詳細イベント画面のタイプは、[詳細なイベント情報 ページ 176](#)に表示されている画面とほぼ同じです。

Event Location 列の情報のほとんどに、ポイント・アンド・シュート・フィールドがあります。

- オフセット・タイプの情報を選択 (M+x、P+x、または、E+x) すると、対応するアドレスの Dump Storage 画面が表示されます。

Dump Storage 画面についての詳細は、[保管場所の表示 ページ 215](#)を参照してください。

- ソース情報またはリスト情報を選択 (L#n または S#n) すると、関連する行またはステートメントの Compiler Listing 画面が表示されます。

Compiler Listing 画面についての詳細は、[ソース・コードの表示 ページ 212](#)を参照してください。

パネル・タイトルの下には、画面の状態を切り替えに使用できる次の 2 つの point-and-shoot フィールドも表示されます。

- 「Full」には、すべてのアプリケーション・イベントと、アプリケーション以外のイベントが表示されます。
- 「Application only」は、イベント・サマリーを、発生元がユーザー・アプリケーションであるイベントのみに限定します。このオプションを選択すると、個々のイベント番号は完全イベント・サマリーとは違ったものになります。

選択したオプションは、対話式分析セッション間で記憶されます。

この画面は、標準の UP、DOWN、LEFT、および RIGHT コマンドに応答します。これらのコマンドは、デフォルトでそれぞれ PF7、PF8、PF10、および PF11 ファンクション・キーに割り当てられます。これらのキーを使用すると、必要に応じて横方向または縦方向に画面をスクロールして、入手可能な情報をすべて表示できます。

オプションのヘルプ・テキストは、画面の最上位の行が表示される場合のみ表示されます。表示が任意の行数分スクロールダウンされると、ヘルプ・テキストは消去されますが、表示が先頭にスクロールされると再表示されます。ヘルプ・テキストの一般情報については、[ヘルプ・テキストの追加または除去 ページ 127](#) を参照してください。

列見出しはビューからスクロールアウトしません。しかし、水平方向にスクロールすると、列見出しはその下のデータとともにスクロールされます。

## 詳細なイベント情報

Event Summary 画面でイベントを選択すると、Event Details 画面が表示されます。[図 97: 障害点の Event Details 画面の例 \(1/2\) ページ 177](#)に例を示します。



図 97. 障害点の Event Details 画面の例 (1/2)

```

File View Services Help
-----
Event 1 of 1: Abend S0C7 *** Point of Failure ***                               Line 1 Col 1 80
Command ==>                                                                    Scroll ==> CSR
JOBNAME: COBPERF6  SYSTEM ABEND: 0CF                                           FAE1      2019/07/21  16

Abend Code. . . . . : S0CF
Program-Interruption Code . : 000F (HFP-Divide Exception)
  The divisor in an HFP division had a zero fraction.

The source code below was executed via the following sequence of PERFORM
statements: ①
Source
Line #
000041          PERFORM CALC THRU CALC-EXIT
000069          PERFORM REDO THRU REDO-EXIT.
000082          PERFORM UNDO THRU UNDO-EXIT
000091          PERFORM ABEND.

COBOL Source Code:
Source
Line #
000097          ABEND.
000098          COMPUTE FIELD-4  ROUNDED =
000099          ELEMENT-4(3) / ELEMENT-2(5)

Data Field Declarations:
Source
Line #
000016          05 ELEMENT-2                      COMP-2.
000018          05 ELEMENT-4                      PIC 999999 COMP-4.
000030          01 FIELD-4  PIC 999999.

Data Field Values:
ELEMENT-2(5) = 0.000000e+00  *** Cause of error ***
ELEMENT-4(3) = 222
FIELD-4      = X'00000000000000'

The listing file used for the above was found in
LJBERRY.LISTING.COBOLE(COBPERF6).

Load Module Name. . . . . : SYS09202.T161638.RA000.COBOLE(COBPERF6.GOSET.H01(COBPER
At Address. . . . . : 16B00988
Load Module Length. . . . : X'1678'
Link-Edit Date and Time . : 2009/07/21  16:16:40

Program and Entry Point Name: COBPERF6
At Address. . . . . : 16B00988 (Module COBPERF6 offset X'0')
Program Length. . . . . : X'A4A'
Program Language. . . . . : COBOL (Compiled using IBM Enterprise COBOL for
z/OS and OS/390 V4 R1 M0 on 2009/07/21 at
16:16:39)
Compiler Options Used . . : NOADATA ADV APOST ARITH(COMPAT) NOAWO
BUFSIZE(4096) NOCICS CODEPAGE(1140) NOCOMPILE(S)
NOCURRENCY DATA(31) NODATEPROC DBCS NODECK
NODIAGTRUNC NODLL NODUMP DYNAM NOEXIT
NOEXPORTALL NOFASTSRT FLAG(I,I) NOFLAGSTD
INTDATE(ANSI) LANGUAGE(EN) LIB LINECOUNT(60)
LIST MAP NOMDECK NONAME NSYMBOL(NATIONAL)
NONUMBER NUMPROC(NOPFD) OBJECT NOOFFSET

```

図 98. 障害点の Event Details 画面の例 (2/2)

```

NOOPTIMIZE OUTDD(SYSOUT) PGMNAME(COMPAT) RENT
RMODE(AUTO) SEQUENCE SIZE(MAX) SOURCE SPACE(1)
NOSQL SQLCCSID SSRANGE NOTERM NOTEST NOTHREAD
TRUNC(STD) NOVBREF NOWORD XMLPARSE(XMLSS)
XREF(FULL) YEARWINDOW(1900) ZWB
Binary Optimizer. . . . . : Automatic Binary Optimizer for z/OS V1 R0 M0 ②
                           optimized COBPERF6 on 2015/09/03 at 12:57:14
                           using ARCH(10)

Machine Instruction . . . . . : 6D008094      DD      FR0,148(,R8)
At Address. . . . . : 16B010D8 (Program COBPERF6 offset X'750')
AMODE . . . . . : 31
Failing Operand . . . . . : Second operand
First Operand (FR0) . . . . . : 42DE0000 00000000
Second Operand Address. . . : 16BB4164 (Module COBPERF6 program COBPERF6
WORKING-STORAGE SECTION BLW=0000 + X'94', symbol
ELEMENT-2, source line # 16 - 433820 bytes of
storage addressable)

Second Operand Length . . : 8
Second Operand Storage. . : 00000000 00000000 *.....*

Program Status Word (PSW) . : 078D2000 96B010DC

General Purpose Registers (AMODE: 64 31 24 , Bytes: Dec Hex ): ③
R0: 16B96190 (Module COBPERF6 program COBPERF6 LOCAL-STORAGE SECTION
BLK=0001 + X'F80')
R1: 16B00C6E (Module COBPERF6 program COBPERF6 + X'2E6')
R2: 800000DE (1826 bytes of storage addressable)
R3: 00000000 (2048 bytes of storage addressable)
R4: 16B00E66 (Module COBPERF6 program COBPERF6 + X'4DE')
R5: 40000000 (Storage invalid)
R6: 00000000 (2048 bytes of storage addressable)
R7: 00000000 (2048 bytes of storage addressable)
R8: 16BB40D0 (Module COBPERF6 program COBPERF6 WORKING-STORAGE SECTION
BLW=0000 + X'0', symbol FILLER, source line # 10 )
R9: 16B90448 (580536 bytes of storage addressable)
R10: 16B00ABC (Module COBPERF6 program COBPERF6 + X'134')
R11: 16B00C98 (Module COBPERF6 program COBPERF6 + X'310')
R12: 16B00A84 (Module COBPERF6 program COBPERF6 + X'FC')
R13: 16B94030 (565200 bytes of storage addressable)
R14: 96B00ECC (Module COBPERF6 program COBPERF6 + X'544', source line # 39 )
R15: 96B577F0 (Module IGZCPAC + X'40C48')

Floating-Point Registers:
R0: 42DE0000 00000000 R2: 00000000 00000000
R4: 00000000 00000000 R6: 00000000 00000000

Associated Messages

CEE3215S The system detected a floating-point divide exception (System
Completion Code=0CF).

Associated Storage Areas

*** Bottom of data.

```



注:

①

COBOL PERFORM トレースバックは、実行されてイベントのソース行に到達した、ネストされている PERFORM を含む、PERFORM



のシーケンスを表示します。トレースバックはコンパイラーによりプロシージャラーがインライン化されている場合には表示されません。これは OPT が有効な場合に発生することがあります。

②

Automatic Binary Optimizer for z/OS® (ABO) 情報は、COBOL プログラムが ABO によって最適化されている場合にのみ、ここに表示されます。

③

汎用レジスターは、イベント AMODE に従って最初に表示されるか、イベント AMODE が決定されていない場合はデフォルトで AMODE 31 に設定されます。ただし、AMODE 64、31、または 24 のポイント・アンド・シュート・フィールドを選択すると、レジスター画面はそれに応じて変わります。また、レジスター説明のバイト数のデフォルト表示は 10 進数ですが、「Dec」または「Hex」の各ポイント・アンド・シュート・フィールドを選択することにより、16 進数に変更したり、再度 10 進数に戻したりすることができます。

現在選択されているイベントに関連するすべての情報は、表示された情報に既に含まれているか、その情報へのポイント・アンド・シュート・リンクが黄色で表示されています。このリンクには、メッセージと異常終了コード (選択すると、説明を表示できます (メッセージおよび異常終了コードの拡張表示 ページ 212 を参照))、および関連したストレージ域 (関連したストレージ域の表示 ページ 208 を参照) が含まれています。

現在表示されているイベントから前または次のイベントを選択するために (障害に複数のイベントが含まれている場合)、黄色のポイント・アンド・シュート・リンクが画面の下部に表示されます。これらのイベントのいずれかにカーソルを置き、Enter キーを押してください。

## 基本オプション: ファイルを開く

“ファイルを開く” オプションを選択すると、特定のイベントに関連付けることができなかったすべてのオープン・ファイルをリストする画面のみならず、個々のイベントに対するレポートの詳細セクションにリストされる可能性のあるファイルも表示されます。図 99: System-Wide Open Files 画面の例 ページ 180 では、オープン・ファイル・リストの例を示しています。

図 99. System-Wide Open Files 画面の例

```

File View Services Help
-----
System-Wide Open Files                                     Line 1 Col 1 80
Command ==>----- Scroll ==> CSR
JOBNAME: COBTSTC   SYSTEM ABEND: 0C9                     FAE1      2019/09/18  09:02:09

Event 1 Program COBTSTC Open Files
File Name . . . . . : OUTDD
Non-Event-Related Open Files
File Name . . . . . : SYSOUT
*** Bottom of data.

```

リストされるファイル名は、ポイント・アンド・シュート・フィールドです。これらのフィールドにカーソルを置き、Enter キーを押すと、関連した詳細ファイル情報が表示されます。例えば、上記の画面の例から OUTDD ファイルを選択すると、[図 100 : File Information 画面の例 ページ 181](#) に示すような「ファイル情報」画面が表示されます。

(Open Files 画面から元の画面に戻るには、Exit (PF3) コマンドを入力してください。)

図 100. File Information 画面の例

```

File View Services Help
-----
File Information                                     Line 1 Col 1 80
Command ==>                                         Scroll ==> CSR
JOBNAME: COBTSTC  SYSTEM ABEND: 0C9                FAE1    2019/09/18  09:02:09

File Name . . . . . : OUTDD
Data Set Name . . . . . : NWILKES.OUT80S ①
File Attributes . . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=SEQUENTIAL,
                           RECFM=VARIABLE BLOCKED SPANNED
Last I/O Function . . . . . : WRITE
Open Status . . . . . : OUTPUT
File Status Code. . . . . : 0

Last Record Written -2. . . : RDW=001C0100 Segment data length 24, variable re
Address  Offset      Hex      EBCDIC / ASCII
16599FC0          C9C9C9F9 F9F94040 40404040 40404040 *III999      *
16599FD0      +10 40404040 40404040          *              *

Last Record Written -1. . . : RDW=00240300 Segment data length 32, variable re
Address  Offset      Hex      EBCDIC / ASCII
16599FE0          40404040 40404040 40404040 40404040 *              *
      Line 16599FF0 same as above

Last Record Written . . . : RDW=001C0200 Segment data length 24, variable re
Address  Offset      Hex      EBCDIC / ASCII
16599EF0          40404040 40404040 40404040 40404040 *              *
16599F00      +10 40404040 40404040          *              *

Current Record Build Area : RDW is zero, no length assigned yet
Address  Offset      Hex      EBCDIC / ASCII
1659AFA4          D1D1D1C1 C1C14040 40404040 40404040 *JJJAAA      *
1659AFB4      +10 40404040 40404040 40404040 40404040 *              *
      Lines 1659AFC4-1659AFD4 same as above
1659AFE4      +40 40404040 40404040 40404040 40406E6E *              >>*
1659AFF4      +50 40404040          *              *

Associated File Control Blocks ②

*** Bottom of data.

```

① で表示されるデータ・セットが QSAM、VSAM、または IAM であり、存在する場合、それはポイント・アンド・シュート・フィールドとして表示されます。これを選択すると、選択メニューが表示され、そこから「編集」、「表示」、または「参照」を選択できます。File Manager for z/OS® が使用可能な場合は、データ・セットに対して要求された機能を実行します。使用不可の場合、データ・セットが QSAM 非スパン・レコード・フォーマットの場合は、ISPF が起動します。

オープン・ファイル・バッファの表示の詳細については、[オープン・ファイルのレコード情報 ページ 275](#)を参照してください。

「Associated File Control Blocks」ポイント・アンド・シュート・フィールドが、上記の②に示すように、File Information 画面の下部で使用できることがあります。このフィールドにカーソルを置いて Enter を押すと、Associated File Control Blocks 画面が表示されます。[図 101 : Associated File Control Blocks 画面の例 ページ 182](#)に、この画面の例が表示されています。

図 101. Associated File Control Blocks 画面の例

```

File View Services Help
Associated File Control Blocks                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
JOBNAME: COBTSTC   SYSTEM ABEND: 0C9                      FAE1      2019/09/18  09:02:09

Data Extent Block (DEB) at Address 008BCD84 :
Address  Offset   Hex                                     EBCDIC / ASCII
008BCD84          038C2CF0 10000000 E8000000 * ...0...Y...*
008BCD90      +C  0F001100 01000000 FF000000 8F01A038 *.....*
008BCDA0     +1C  048BCD60 10F41AE0 000000FF 000000FF *...-.4.\.....*
008BCDB0     +2C  000E000F 00010001 00000000 00000000 *.....*
008BCDC0     +3C  00000054 F3C2C1D9 C2D70000 00000000 *...3BARBP.....*
008BCDD0     +4C  00000000 00000000 00000000 00000000 *.....*
008BCDE0     +5C  00000000 00000000 E2E6C1D4 00000000 *.....SWAM.....*
008BCDF0     +6C  00000218 008BCE08 500000E6 008BD6B0 *.....&..W..0.*
008BCE00     +7C  008BD140                                     *..J *

Data Control Block (DCB) at Address 0001A038 :
Address  Offset   Hex                                     EBCDIC / ASCII
0001A038          1656C858 00000000 * ..H....*
0001A040      +8  00FF0000 0EF15026 002FBE96 07022FE8 *....1&....o...Y*
0001A050     +18  00004000 00006B70 E6000001 5801E8B4 *.. ...,W....Y.*
0001A060     +28  007C0048 008BCD84 92C9D870 00C8B348 *.@.....dkIQ..H.*

```

### 基本オプション: CICS 情報

["CICS Information"] オプションを選択すると、[図 102 : CICS Information 画面の例](#) ページ 182に示されている例のように、CICS® に関連した情報の画面が表示されます。

図 102. CICS Information 画面の例

```

File View Services Help
CICS Information                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: MXT1      CICS ABEND: ABC1                      FAE1      2022/01/16  12:06:24

CICS Release. . . . . : 0730 (TS 5.6)
Application ID. . . . . : C73E1FA1
CICS Transaction ID . . . . : MXT1
CICS Task Number. . . . . : 02432
CICS Terminal ID. . . . . : 1006
CICS Terminal Netname . . . : TCPE1006

Select one of the following:
  1. CICS Control Blocks
  2. CICS Transaction Storage Summary
  3. CICS Transaction Storage
  4. Last CICS 3270 Screen Buffer
  5. Summarized CICS Trace
  6. CICS Trace Formatting
  7. CICS Task Trace Table
  8. CICS Recovery Manager
  9. CICS Levels, Commareas, and Channels
 10. CICS Monitoring Data
 11. CICS Event Program Information
 12. CICS Asynchronous Information
 13. CICS System Initialization Parameters
*** Bottom of data.

```

CICS information 画面から、カーソルを使用してオプションを選択するか、またはコマンド行に入力することにより、追加の CICS 情報を表示することができます。

## CICS Control Blocks

トランザクション異常終了に関連する CICS® 制御ブロックを表示するには、このオプションを選択します。

## CICS Transaction Storage Summary

「CICS® トランザクション・ストレージ要約」リンクを選択すると画面が表示され、[図 103 : CICS Transaction Storage Summary 画面の例](#) ページ 183 に例示されているように、現行 CICS® トランザクションで所有されているストレージ域に関する情報が示されます。

図 103. CICS Transaction Storage Summary 画面の例

File View Services Help					
CICS Transaction Storage Summary - All Tasks					Line 1 Col 1 80
Command ==>					Scroll ==> CSR
Current All - TRANID: CFA CICS ABEND: FLT1					FAE1 2019/07
Address	Length	Type	Tran ID	Task #	Possible Overlay
00100008	00002D60		NEWC	0002915C	n/a
00103008	00000650		CECI	0002916C	n/a
00104000	000029F0	USER24			None detected
Total:	000029F0	USER24			
19320008	00000030		NEWC	0002915C	n/a
193201E8	000000C0		NEWC	0002915C	n/a
193202B8	00003AD0		NEWC	0002915C	n/a
19323D98	0000A770		NEWC	0002915C	n/a
19330008	000001C0		CECI	0002916C	n/a
193301E8	000000C0		CECI	0002916C	n/a
193302B8	00007D00		CECI	0002916C	n/a
19337FC8	00000120		CECI	0002916C	n/a
193380F8	000004F0		CECI	0002916C	n/a
1934B170	00000FD0	USER31			None detected
19340660	000000E0	USER31			None detected
19340910	0000A860	USER31			None detected

「CICS® トランザクション・ストレージ要約」画面は、トランザクションのストレージが別のタスクによってオーバーレイされているときに重要になります。この場合、所有しているストレージがオーバーレイに近かった、または隣接していたトランザクションを識別するときに役立ちます。CICS® トランザクション異常終了を分析するために Fault Analyzer が起動されると、同時に実行されているその他すべてのタスクのストレージ割り振りが収集されます。対話式再分析の間、この情報は「CICS® トランザクション・ストレージ要約」画面に表示されます。

画面には、2つのポイント・アンド・シュート・フィールド「Current」および「All」があります。これらを選択すると、該当する画面に切り替わります。

CICSSTG コマンドを使用して、CICS® トランザクション・ストレージを表示することもできます。[CICSSTG ページ 105](#) を参照してください。

## CICS Transaction Storage

異常終了したトランザクションによって割り振られた CICS® ストレージを表示するには、このオプションを選択します。この画面には、CICS® Transaction Storage Summary 画面で個別に選択できる同じストレージ域の複合ビューが表示されません。

または、CICS®STG コマンドを使用して CICS® トランザクション・ストレージを表示することもできます。CICSSTG ページ 105 を参照してください。

### Last CICS 3270 Screen Buffer

["Last CICS 3270 Screen Buffer"] リンクを選択すると、図 104 : Last CICS 3270 Screen Buffer 画面の例 ページ 184 の例のように、CICS により最後に書き込まれた 3270 画面バッファが表示されます。

図 104. Last CICS 3270 Screen Buffer 画面の例

```

File View Services Help
-----
Last CICS 3270 Screen Buffer                               Line 1 Col 1 80
Command ==>                                             Scroll ==> CSR
Normal Hex - TRANID: SK00   CICS ABEND: ASRA           CICS02   2019/06/07 11:10:32

      Column
Row  +---+---+---+---+---+---+---+---+---+---+---+---+
  1  sk00*** CICB0CB0 - START OF PROGRAM.....
  2  .....
  3  .....
  4  .....
  5  .....
  6  .....
  7  .....
  8  .....
  9  .....
 10  .....
 11  .....
 12  .....
 13  .....
 14  .....
F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right
    
```

印刷不能文字はすべてピリオド (.) で示されます。

Fault Analyzer は常に CICS 3270 画面バッファ情報を提供できるわけではありません。例えば、端末で VTAM® セッション・エラーが通知された場合、または以下のいずれかの CICS® 異常終了が発生した場合 (\* はワイルドカード文字を示します)、3270 画面バッファ情報は提供されません。

- AEXZ
- AKC3
- AKCT
- AKK\*
- ATC\*
- ATN\*
- AZCT
- AZI\*
- AZTS

パネル・タイトルの下には、画面の状態を切り替えに使用できる次の 2 つのポイント・アンド・シュート・フィールドが表示されます。





図 106. Summarized CICS Trace 画面の例

```

File View Services Help
-----
Summarized CICS Trace                                     Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: CS31      CICS ABEND: ASRA                        FAE2      2019/08/16 12:06:24

00027 QR    AP EA00 TMP    ENTRY LOCATE                PFT,DFHCICST
00027 QR    XS 0701 XSRC   ENTRY CHECK_CICS_RESOURCE   CS31,TRANSATTACH,EXECUTE
00027 QR    PG 0901 PPGG   ENTRY INITIAL_LINK          CSCB0310
00027 QR    AP 1940 APLI   ENTRY ESTABLISH_LANGUAGE     CSCB0310,17F00160,97F001
00027 QR    AP 1940 APLI   ENTRY START_PROGRAM          CSCB0310,CEDF,FULLAPI,EX
00027 QR    AP 00E1 EIP    ENTRY SEND-TEXT
  Called-from-address 17F0051E : Module CSCB0310 program CSCB0310 + X'396', sou
00027 QR    AP 00E1 EIP    EXIT  SEND-TEXT                OK

00027 QR    AP 00E1 EIP    ENTRY HANDLE-ABEND
  Called-from-address 17F0055C : Module CSCB0310 program CSCB0310 + X'3D4', sou
00027 QR    PG 0701 PGHM   EXIT  SET_ABEND/OK
00027 QR    AP 00E1 EIP    EXIT  HANDLE-ABEND            OK

00027 QR    AP 0790 SRP    *EXC* PROGRAM_CHECK
00027 QR    DU 0601 DUTM   EXIT  INQUIRE_SYSTEM_DUMP CODE/EXCEPTION DUMPCODE_NOT
00027 QR    DU 0601 DUTM   EXIT  LOCATE_SYSTEM_DUMP CODE/OK FFFFFFFF,1,0,0,NO,YE
00027 QR    DU 0601 DUTM   EXIT  COMMIT_SYSTEM_DUMP CODE/OK

```

標準の CICS® トレース・テーブルは、さらに使いやすくするために、Fault Analyzer により拡張されます。呼び出し点の起  
点アドレスを示す情報が追加されます。この情報には、モジュール名、CSECT 名、CSECT 内のオフセット、およびソース  
行またはリスト・ステートメントの番号 (使用可能な場合) が含まれます。

対話式レポートの他のすべてのセクションのように、ソース行番号またはリスト・ステートメント番号を選択するには、  
カーソルを番号の上に置いて Enter キーを押します。選択すると、[ソース・コードの表示 ページ 212](#)に示されているよう  
に、ソース・リスト画面全体が表示されます。

## CICS Trace Formatting

“[\[CICS® Trace Formatting\]](#)”リンクを選択すると、[図 107 : CICS Trace Selection Parameters 画面の例 ページ 187](#) の例  
のように、特定の選択パラメーターを CICS® トレースに入力するための画面が表示されます。

図 107. CICS Trace Selection Parameters 画面の例

```

File View Services Help
S CICS T CICS Trace Selection Parameters
Specify CICS trace selection parameters and press Enter.
Format . . . . . A (Abbrev/Short/Full)
Exception Only . . N (Yes/No)
Sequence Start . . 000001
                  End . . 000375
Highlight Interval 0.128 (0-99.9999999999 secs)
Task IDs . . . . . 00027
KE Task Numbers
Terminal IDs . . . . . Caps Y
Transaction IDs . . . . . Caps Y
Time Start . . . . . (HHMMSS)
End . . . . . (HHMMSS)
Domain/Point IDs
Exclude IDs

```

7. CICS Levels, Commareas, and Channels

\*\*\* Bottom of data.

Line 1 Col 1 80  
roll ==> CSR  
5/22 14:22:19

以下のパラメーターを指定できます。

#### フォーマット

これは、トレース・フォーマット設定のレベルを指定します。

##### A

トレースの簡略形式 (項目ごとに 1 行)。

##### S

短いフォーマット設定表示。これは、簡略項目に加えて、完全にフォーマット設定されたパラメーター・リスト、戻りアドレス、時刻、および間隔で構成されます。

##### F

各項目における全データの完全フォーマット設定表示。

#### Exception Only

内部トレースの例外項目のみを表示することを指定します。

#### Sequence Start/End

選択するシーケンス番号を指定します (シーケンス番号は、1 から始まり、最大 6 桁の長さです)。開始、終了、または両方を指定できます。

#### Highlight Interval

内部トレース項目間のインターバルを指定します。このインターバルの後、項目が (アスタリスクを付けて) 強調表示されます。

#### Task IDs

トレース項目を表示する最大 5 個のタスク ID を指定します。ID 値は以下のいずれかにすることができます。

- 最大 5 桁の長さとなる任意の数字。
- J01 から J99 までの任意の JAS。
- III、TCP、または DST。
- 接続ドメインの 2 文字のドメイン ID (非 TCA タスクの場合)。

### KE Task Numbers

これは、表示される 4 桁の 16 進数字のカーネル・タスク番号を 5 個まで指定します。

### Terminal IDs

トレース項目を表示する最大 5 個の端末の端末 ID を指定します。いずれかの端末 ID に小文字が含まれている場合、Caps を N に設定し、すべての ID を大/小文字の区別をして入力します。Caps が Y の場合、すべての項目が大文字に変換されます。

### トランザクション ID

トレース項目を表示する最大 5 個のトランザクションのトランザクション ID を指定します。いずれかのトランザクション ID に小文字が含まれている場合、Caps を N に設定し、すべての ID を大/小文字の区別をして入力します。Caps が Y の場合、すべての項目が大文字に変換されます。

### Time Start/End

トレース項目を表示する期間を指定します。開始、終了、または両方を指定できます。

### Domain/Point IDs

これは、2 文字のドメイン ID を 5 個まで指定します (オプションでドメイン内のトレース・ポイント ID をドメイン ID の後に指定します)。これらの IDs は含まれています。このドメイン ID には次のものがあります。

```
AP BA BR CC DD DE DH DM DP DS DU EJ EM EX GC IE II IS KE LC
LD LG LM ME ML MN NQ OT PA PG PI PT RL RM RS RX RZ SH SJ SM
SO ST TI TR TS US WB W2 XM XS
```

トレース・ポイント ID は最大 4 桁の長さの 16 進数字にすることができます (4 桁未満の長さが入力されている場合、トレース・ポイント ID は総称値として扱われます)。

この選択パラメーター・フィールドにドメイン ID もトレース・ポイント ID も指定されない場合は、デフォルトですべてのトレース項目が組み込まれます。ただし、「Exclude IDs」選択パラメーターで指定されたトレース項目はすべて例外として除外されます。それ以外の場合は、指定された ID に適合するトレース・レコードのみ (この場合も、Exclude IDs 選択パラメーターで除外されたトレース・レコードを除く) が表示されます。

### Exclude IDs

これは、2 文字のドメイン ID (Domain/Point IDs の場合と同じ値) を 5 個まで指定します (オプションでドメイン内のトレース・ポイント ID をドメイン ID の後に指定します)。指定されたドメイン ID は処理対象から除外されます。トレース・ポイント ID は最大 4 桁の長さの 16 進数字にすることができます (4 桁未満の長さが入力されている場合、トレース・ポイント ID は総称値として扱われます)。

オプションでパラメーターを変更します。Enter を押して CICS® トレースを表示します。CICS® Trace 画面の例を、[図 108 : CICS Trace 画面の例 ページ 189](#) に示します。

図 108. CICS Trace 画面の例

File View Services Help										
CICS Trace										
Command ==>										
Abbrev	Short	Full	TRANID: CS65				IDISNAP	CALL	Entry 1 Col 1 80	Scroll ==> CSR
-----										
FAC2 2										
00027	QR	SM	0D02	SMMF	EXIT	FREEMAIN/OK		CICS storage at	174E7748	
00027	QR	SM	0C01	SMMG	ENTRY	GETMAIN		2000,NO,00,CICS		
00027	QR	SM	0C02	SMMG	EXIT	GETMAIN/OK		174CF028		
00027	QR	SM	0D01	SMMF	ENTRY	FREEMAIN		174CD018,CICS		
00027	QR	SM	0D02	SMMF	EXIT	FREEMAIN/OK		CICS storage at	174CD018	
00027	QR	SM	0C01	SMMG	ENTRY	GETMAIN		2000,NO,00,CICS		
00027	QR	SM	0C02	SMMG	EXIT	GETMAIN/OK		174D1038		
00027	QR	SM	0D01	SMMF	ENTRY	FREEMAIN		174CF028,CICS		
00027	QR	SM	0D02	SMMF	EXIT	FREEMAIN/OK		CICS storage at	174CF028	
00027	QR	SM	0C01	SMMG	ENTRY	GETMAIN		2000,NO,00,CICS		
00027	QR	SM	0C02	SMMG	EXIT	GETMAIN/OK		174CD018		
00027	QR	SM	0D01	SMMF	ENTRY	FREEMAIN		174D1038,CICS		
00027	QR	SM	0D02	SMMF	EXIT	FREEMAIN/OK		CICS storage at	174D1038	
00027	QR	SM	0C01	SMMG	ENTRY	GETMAIN		2000,NO,00,CICS		
00027	QR	SM	0C02	SMMG	EXIT	GETMAIN/OK		174CF028		
00027	QR	SM	0D01	SMMF	ENTRY	FREEMAIN		174CD018,CICS		
00027	QR	SM	0D02	SMMF	EXIT	FREEMAIN/OK		CICS storage at	174CD018	
00027	QR	SM	0C01	SMMG	ENTRY	GETMAIN		2000,NO,00,CICS		
00027	QR	SM	0C02	SMMG	EXIT	GETMAIN/OK		174E7748		

最初、トレースは、CICS® Trace Selection Parameters 画面での設定値に従ってフォーマット設定されています。しかし、トレースの表示中はいつでも、画面の上部に配置されているポイント・アンド・シュート・フィールドの1つを選択してフォーマットを変更することができます。表示された最上位のトレース項目は、選択されたフォーマットに関係なく、上部に残ります。



**注:** Fault Analyzer は CICS® 提供のトレース・フォーマット設定モジュールを使用して CICS® トレース項目をフォーマットします。これらのモジュールは DFHTRnnn と呼ばれます。nnn は CICS® バージョンです。障害項目に複数バージョンの CICS® の CICS® トレース・レコードが含まれる場合、Fault Analyzer が必要なすべてのバージョンの DFHTRnnn を使用できることを確認する必要があります。

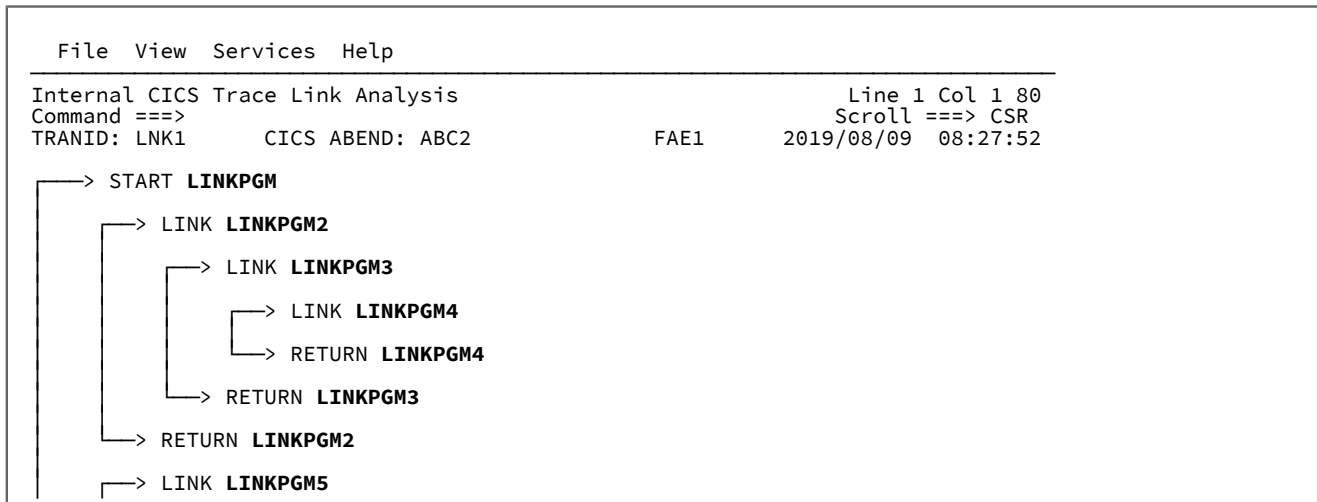
## 内部 CICS トレース・リンク分析 (CICSLINK)

フォーマット設定された CICS® トレースを表示するときに CICSLINK コマンドを使用すると、CICS® LINK 項目および出口が階層で表示されます。CICS® フォーマット設定トレース画面では CICSLINK コマンドはデフォルトで PF4 に割り当てられています。CICSLINK コマンドについて詳しくは、「[CICSLINK ページ 105](#)」を参照してください。

この機能は、CICS® システム・ダンプまたは補助トレース・データ・セットによって提供されるフォーマット設定済み CICS® トレースにも使用できます。これらのいずれかのソースから提供されるフォーマット設定済みトレースに対して CICSLINK コマンドを使用する場合は、分析対象のタスク番号を指定するようにプロンプトが出されます。フォーマット設定済みトレースでタスク番号上にカーソルを置いて **PF4** を押すと、そのタスク番号に関する「CICS® Trace Link Analysis」が開きます。

以下の図に画面の例を示します。

図 109. サンプル 「CICS Trace Link Analysis」 画面



この図は、EXEC CICS® LINK 項目の CICS® トレースを解析することで生成されます。表示されている階層には、以下の見出しが示されています。

#### START

この項目は CICS® トレースの開始を示します。この項目は、最初のプログラムが実行され始めたリンク分析の先頭に常に表示されます。

#### LINK

この項目は、LINK CICS® トレース項目が発生したポイントを示します。この項目は、CICS® トランザクションにおいて、ネストされたプログラムの実行の開始をマークします。

#### RETURN

この項目は、現行最高レベルのリンク済みモジュールが終了するポイントを示します。

#### ABEND

この項目は、トランザクションにおいて異常終了が発生したポイントを示します。

階層におけるいずれかのポイント・アンド・シュート・プログラム名フィールドを選択すると、選択したプログラムと、そのプログラムの実行時に発生したイベントについて、さらに詳しい情報が表示されます。例として下図を参照してください。

図 110. サンプル「CICS Trace Link Analysis」追加情報画面

```

File View Services Help
Internal CICS Trace Link Analysis
Command ==>
TRANID: LNK1          CICS A
Line 1 Col 1 80
Scroll ==> CSR
/09 08:27:52

-> START LINKPGM
  -> LINK LINKPGM2
    -> LINK LINK
      -> LINK
        -> RETU
      -> RETURN LI
    -> RETURN LINKPGM
  -> LINK LINKPGM5

Module Name: LINKPGM2
Address.....: 1B1E4150
Size.....: 0000377C (14204)
DDname.....: DFHRPL
DSN.....: MIFROEN.TEST.LOAD

CICS EXEC Call List
EXEC Command Name  #
GETMAIN            1
SEND TEXT          1
ASKTIME            1
RETURN             1
FREEMAIN           1

```

**Module Name**

リンクされたロード・モジュールの名前。

**Address**

プログラムのアドレス。CICS® 補助トレース・データ・セットまたは CICS® システム・ダンプが分析されているときは、このアドレスは使用できません。

**Size**

プログラムのサイズ。CICS® 補助トレース・データ・セットまたは CICS® システム・ダンプが分析されているときは、このサイズは使用できません。

**DDname**

ロード・モジュール起点 ddname。CICS® 補助トレース・データ・セットまたは CICS® システム・ダンプが分析されているときは、この DDname は使用できません。

**DSN**

ロード・モジュール起点データ・セット名。CICS® 補助トレース・データ・セットまたは CICS® システム・ダンプが分析されているときは、DSN は使用できません。

**CICS® EXEC Call List**

当該プログラムの実行時に CICS® トレースで記録されるすべての CICS® EXEC コマンドのリスト。このリストには、EXEC コマンド名と、個々のコマンド呼び出しの数が含まれます。このリストには、ネストされたプログラムから呼び出された EXEC コマンドは含まれません。

**Storage Violations Detected**

当該プログラムの実行時に CICS® トレースの SM トレース項目から検出されたすべての CICS® 記憶保護違反のリスト。このリストには、違反のタイプと、個々の違反の数が含まれます。このリストは、記憶保護違反が発生しなかった場合は表示されません。また、ネストされたプログラムで発生した記憶保護違反は含まれません。

## CICS タスク・トレース・テーブル

CICS® タスク・トレース・ブロックを表示するには、このオプションを選択します。このトレース・ブロックには、タスクによって作成された最後の 20 個のトレース項目が含まれています。

## CICS Recovery Manager

CICS® Recovery Manager に固有の情報を表示するには、このオプションを選択します。

## CICS Levels, Commareas, and Channels

["CICS® レベル、コマンド域、およびチャネル"] ポイント・アンド・シュート・フィールドを選択すると、[図 111 : CICS Levels, Commareas, and Channels 画面の例](#) ページ 192 に例示されている画面が表示されます。

図 111. CICS Levels, Commareas, and Channels 画面の例

File View Services Help									
CICS Levels, Commareas, and Channels									
Command ==> ----- FAE1 2019/09/17 14:39:58									
TRANID: CONT CICS ABEND: CVER									
Lnk	Ev	Commaarea			Created	Container name		Passed f	
#	#	Type	Program	Address	Len	Channel(s)		Channel	
1	4	EXEC	CONTEST			PASS1001	CONT1002		
						LOCAL102	CONT1001		
						COMMONCHANNEL	CONT1002		
						LOCAL101	CONT1001		
2	8	EXEC	CONTEST2			PASS2001	COMMONCONTAINER		PASS1001
						LOCAL202	CONT1002		
						COMMONCHANNEL	CONT2002		
						LOCAL201	CONT2001		
						COMMONCHANNEL	CONT2002	①	
						LOCAL201	CONT2001		
						COMMONCHANNEL	COMMONCONTAINER		
3	12	EXEC	CONTEST3			PASS3001	CONT2002		PASS2001
							CONT2001		
							CONT3001-1MB		

ポイント・アンド・シュート・フィールドであるコンテナ名を選択すると、コンテナ・タイプに関する情報が表示されます。例えば、[図 111 : CICS Levels, Commareas, and Channels 画面の例](#) ページ 192 の ① にある CONT2002 ポイント・アンド・シュート・フィールドを選択すると、次の画面が表示されます。

図 112. サンプル CICS コンテナ画面

CICS Container									
Command ==> ----- FAE1 2019/09/17 14:39:58									
TRANID: CONT CICS ABEND: CVER									
Container CONT2002 at address <b>177E33A0</b> has a length of X'1024'									
Segment Address	Segment Offset	Segment (EBCDIC)	Preview	Segment (Hex)	Preview				
<b>17912028</b>	X'000000'	CONT2002/L202...	C3D6D5E3 F2F0F0F2 61D3F2F0 F24B						
<b>1790C328</b>	X'000FD8'	-----	60606060 60606060 60606060 6060						

一部の標準 CICS® コンテナに関しては、コンテナの説明も表示されます。例については、[図 113 : サンプル標準 CICS コンテナ画面](#) ページ 193 を参照してください。



図 113. サンプル標準 CICS コンテナー画面

```

Container Data
Command ==> ----- Line 1 Col 1 80
TRANID: COBA CICS ABEND: ATCV FAE1 2019/04/06 12:32:44 Scroll ==> CSR

DFHWS-BODY Contains the body section of the SOAP envelope. Typically, the
application will modify the contents.

Data Length X'3FE' Format XML Data

Address  Offset      Hex                               ASCII / EBCDIC
16A4B788                3C534F41 502D454E * <SOAP-EN*
16A4B790          +8  563A426F 64793E20 20202020 20202020 *V:Body> *
16A4B7A0          +18 20202020 20202020 20202020 20202020 * *
      Lines 16A4B7B0-16A4B7C0 same as above
16A4B7D0          +48 20202020 20200D0A 3C434943 53564552 * ..<CICSVER*
16A4B7E0          +58 324F7065 72617469 6F6E3E20 20202020 *20peration> *
16A4B7F0          +68 20202020 20202020 20202020 20202020 * *
      Lines 16A4B800-16A4B810 same as above
16A4B820          +98 20202020 20202020 0D0A3C63 6F6D6D61 * ..<comma*
16A4B830          +A8 7265613E 20202020 20202020 20202020 *rea> *
16A4B840          +B8 20202020 20202020 20202020 20202020 * *
      Lines 16A4B850-16A4B860 same as above

```

16進データの右側にある文字で表現されるセクションは、自動的に EBCDIC または ASCII のいずれかで表示されます。

[Format XML Data] ポイント・アンド・シュート・フィールドを ② で選択すると、フォーマット設定された XML データが表示されます。

図 114. CICS XML Formatter 画面の例

```

XML Formatter
Command ==> ----- Line 1 Col 1 80
TRANID: COBA CICS ABEND: ATCV FAE1 2019/04/06 12:32:44 Scroll ==> CSR

<SOAP-ENV:Body>
  <CICSVER20peration>
    <commarea>
      <DisplayOrUpdate>
        T
      </DisplayOrUpdate>
      <LinkRC>
        0
      </LinkRC>
      <msg1>
        Message 1
      </msg1>
      <msg2>
        Message 2
      </msg2>
      <msg3>
        Message 3
      </msg3>

```

## CICS モニター・データ

CICS®トランザクション・モニター域 (TMA) 制御ブロックからの詳細情報を表示するには、このオプションを選択します。

## CICS イベント・プログラム情報

この障害項目の CICS® 処理プログラム・テーブル (PPT) で検出されたすべての項目のリストを表示するには、このオプションを選択します。

## CICS システム初期化パラメーター

CICS 障害項目の対話式再分析中に、CICS® システム初期化パラメーターの値を表示できます。

- 対話式レポートの「CICS 情報」セクションから、「CICS システム初期化パラメーター」オプションを選択できます。
- ISPF コマンド行から SIT コマンドを入力できます。

値はアルファベット順にリストされます。ポイント・アンド・シュート・フィールドを使用して、値をカテゴリ別に表示できます。

## 基本オプション: メッセージ

システム・コンソールに書き込まれ、特定のイベントに属するものと識別されなかったメッセージは、「メッセージ」見出しの下にリストされます。特定のイベントに対して識別される LE メッセージも含まれます。メッセージ番号にカーソルを置き、Enter キーを押すと、個々のメッセージを選択して説明を表示できます。

“「メッセージ」”リンク選択時に表示される画面の例を [図 115: System-Wide Messages 画面の例](#) ページ 194 に示します。

図 115. System-Wide Messages 画面の例

```

File View Services Help
-----
System-Wide Messages                               Line 1 Col 1 80
Command ==>                                         Scroll ==> CSR
JOBNAME: P00398  SYSTEM ABEND: 0CB  MVS2  2019/03/24 13:54:05

Event 5 Messages
CEE3211S Job-specific text not available

Non-Event-Related Messages
$HASP375 P00398 ESTIMATED LINES EXCEEDED
$HASP375 P00398 ESTIMATE EXCEEDED BY           10,000 LINES
$HASP375 P00398 ESTIMATE EXCEEDED BY           20,000 LINES

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

## 基本オプション: DB2 Information

「DB2 Information」オプションを選択すると、[図 116: DB2 Information 画面の例 \(1/3\)](#) ページ 195 に示されている例のように、DB2® に関連した情報の画面が表示されます。

図 116. DB2 Information 画面の例 (1/3)

```

File View Services Help
-----
DB2 Information                                     Line 1 Col 1 80
Command ==>                                     Scroll ==> CSR
JOBNAME: DAC2DH07  SYSTEM ABEND: 806             MVS2      2019/11/26 09:44:50

DB2 Subsystem DBT6

DB2 Version . . . . . : V6R1M0
Plan Name . . . . . : DAC2DH (Bound 2019/04/25 12:13:14)
Plan Owner. . . . . : HARRIDA
Package Name. . . . . : DAPNDH11 (Created 2019/07/30 10:06:08, bound
                        2004/04/25 12:13:13)
Package Collection ID . . . : DAC2DHPK
Package Owner . . . . . : HARRIDA
Package Creator . . . . . : HARRIDA
Package Version . . . . . : HV01
Package Qualifier . . . . . : DSN8610
Database Request Module Name: CTEST.DB2.DBRMLIB.DATA(DAPNDH11) (Precompiled
                        2019/04/25 12:07:57)
Consistency Token . . . . . : X'177522E41D026D08'
Primary Authorization ID. . : HARRIDA
Current SQL ID. . . . . : HARRIDA

```

図 117. DB2 Information 画面の例 (2/3)

```

Last Executed SQL Statement:
List
  Stmt #
  000227          EXEC SQL FETCH HVAR1 INTO :HVTABLE

Fault Analyzer Event #. . . : 7 (Program DAPNDH11)
Declare Cursor Stmt No. . . : 133
Declare Cursor Stmt . . . : DECLARE HVAR1 CURSOR FOR SELECT HVARKEY ,
                          VCHAR300 , DEC9 , LINT , CHARHEX , TIMESTMP
                          FROM HVAR5
Open Cursor Stmt No . . . . : 166
Open Cursor Stmt. . . . . : OPEN HVAR1

Browse Table Contents in File Manager:
HVAR5

Output Host Variables:
Name and Data Type. . . . : HVTABLE.HVARKEY CHARACTER(6)
  At Address. . . . . : 0007EA26
  Data Value. . . . . : 000003

Name and Data Type. . . . : HVTABLE.VCHAR300 VARCHAR(300)
  At Address. . . . . : 0007EA2C
  Data Value. . . . . : This is record 3
                          =====
                          =====101
                          =====
                          201
                          =====
                          =====>

Name and Data Type. . . . : HVTABLE.DEC9 DECIMAL(9,4)
  At Address. . . . . : 0007EB5A
  Data Value. . . . . : 12345.6789

Name and Data Type. . . . : HVTABLE.LINT INTEGER
  At Address. . . . . : 0007EB74
  Data Value. . . . . : 214748364

Name and Data Type. . . . : HVTABLE.CHARHEX CHARACTER(21)
  At Address. . . . . : 00082A8C
  Data Value. . . . . : *** Data format error: Character string contains
                          non-printable character at offset X'B' ***

Host Variable Storage:
  Address  Offset      Hex          EBCDIC / ASCII
  00082A8C          E4959799 8995A381 82938522 83888199 *Unprintable.char*
  00082A9C          +10      8183A385 99          *acter          *

Name and Data Type. . . . : HVTABLE.TIMESTMP CHARACTER(26)
  At Address. . . . . : 000869C9
  Data Value. . . . . : 2019-09-30-10.44.59.000001
    
```

図 117: DB2 Information 画面の例 (2/3) ページ 196 に示す画面から SQL 表のポイント・アンド・シュート・フィールドを選択すると、File Manager DB2 Browse パネルに選択した表の事前に入力された詳細が表示されます。SQL 表のポイント・アンド・シュート・フィールドは、File Manager がインストールされていない限り表示されません。

図 118. DB2 Information 画面の例 (3/3)

```

DB2 Control Blocks

SQL Communications Area (SQLCA) for Event # 7 Program DAPNDH11 at Address
1747E9F8 :
Offset   Field      Value
Dec  Hex  Name      Hex      EBCDIC / ASCII
-----
  0      (0)  SQLCAID  E2D8D3C3 C1404040  *SQLCA      *
  8      (8)  SQLCABC  00000088          *...h      *
 12      (C)  SQLCODE  00000064          *....      *
          SQLCODE 100 Explanation
 16      (10) SQLERRML 0000              *..        *
 18      (12) SQLERRMC 40404040 40404040 40404040 40404040 *          *
 34      (22)          40404040 40404040 40404040 40404040 *          *
 50      (32)          40404040 40404040 40404040 40404040 *          *
 66      (42)          40404040 40404040 40404040 40404040 *          *
 82      (52)          40404040 4040      *          *
 88      (58) SQLERRP  C4E2D5E7 D9C6D540 *DSNXRFN   *
 96      (60) SQLERRD  FFFFFFFF92 00000000 00000000 FFFFFFFF *...k.....*
112     (70)          00000000 00000000 *.....   *
120     (78) SQLWARN  40404040 40404040 404040 *          *
131     (83) SQLSTATE F0F2F0F0 F0      *02000    *
          SQLSTATE 02000 Explanation

SQL Communications Area (SQLCA) for subsystem DB31 not shown as it is
identical to the SQLCA for event # 7 program DAPNDH11.

*** Bottom of data.
  F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
 F10=Left     F11=Right

```

1つ以上のDB2®サブシステム情報が表示されます。DB2®サブシステムのSQLCAデータ域がEvent Details画面から使用可能なものと同一である場合、上記の例のように、この画面には、そのイベントへの参照のみが表示されます。それ以外の場合は、SQLCAの内容がこの画面に表示されます。

## 基本オプション: IMS Information

["IMS™ Information"] オプションを選択すると、IMS™に関連した情報の画面が表示されます。図 119: IMS Information 画面の例 (1/4) ページ 197 に画面の例があります。

図 119. IMS Information 画面の例 (1/4)

```

File View Services Help
-----
IMS Information                               Line 1 Col 1 80
Command ==>                                Scroll ==> CSR
JOBNAME: IBCB0030  USER ABEND: 4036          MVS2      2019/11/29 13:51:55

Summary Detail - JOBNAME: IDCB0070  SYSTEM ABEND: 0CB          FAE1 2015

IMS Version . . . . . : V13R1M0
IMS Region Type . . . . . : Batch Message Processing Region
IMS Subsystem Name. . . . . : IB81
Application Program Name. . : IDCB0070
PSB Name. . . . . : DFHSAM25

```

図 120. IMS Information 画面の例 (2/4)

```

Last DL/I Call Parameter List

Note that storage addressed by individual parameters might no longer be valid.

Parameter 1 . . . . . : 0002BD68
  DL/I Call Function. . . . : GU (Get Unique)

Parameter 2 . . . . . : 0001A098
  (See "IMS Control Blocks" for details of this PCB)

Parameter 3 . . . . . : 0002BCA8

Parameter 4 . . . . . : 8002BD48
  SSA # 1 . . . . . : PARTROOT(           =           )

IMS Control Blocks

Input/Output Program Communications Blocks (IOPCBs)

IOPCB:
  At Address. . . . . : 0001A020
  PCB Name. . . . . : IOPCB
  Relative PCB Number . . . . : 1
  PCB Type. . . . . : I/O
  Logical Terminal ID . . . . : n/a
  Status Code . . . . . : ' ' (Normal status)
  User ID . . . . . : DFHSAM25
  Group Name. . . . . : n/a
  Formatting Module Name. . . : n/a

Data Base Program Communications Blocks (DBPCBs)

DBPCB (** Current/Last Used **):
  At Address. . . . . : 0001A098
  PCB Name. . . . . : n/a
  Relative PCB Number . . . . : 2
  PCB Type. . . . . : Data Base or Online
  Data Base Name. . . . . : DI21PART
  Segment Level . . . . . : 01
  Status Code . . . . . : ' ' (Normal status)
  Processing Options. . . . . : A
  Segment Name. . . . . : PARTROOT
  Number of Segments. . . . . : 5
  Key Feedback Length . . . . : 17

Key Feedback Data:
  Address  Offset      Hex                               EBCDIC / ASCII
  0001A0F4                F0F2C1D5 F9F6F0C3 F1F04040 40404040 *02AN960C10 *
  0001A104          +10  40                               *                               *
    
```

図 121. IMS Information 画面の例 (3/4)

```

JCB DL/I Call Trace (Most recent call first):
  Call
  # Code Description      Status Code Description
  1 01  GHU or GU          ' '      Status good.
  2 03  GHN or GN          AB       Segment I/O area required; none specified
                                     in call. Only applies to full-function DEQ
                                     calls.
                                     -or-
                                     BB       Call could not be completed because data
                                     was unavailable and updates are backed out
                                     only since the last commit point.
                                     -or-
                                     GB       End of database.
  3 03  GHN or GN          ' '      Status good.
  4 03  GHN or GN          ' '      Status good.
  5 03  GHN or GN          ' '      Status good.
  6 03  GHN or GN          ' '      Status good.

IMS Accounting Information

DL/I Data Base Calls:
GU Calls. . . . . : 151
GN Calls. . . . . : 10050
GNP Calls . . . . . : 0
GHU Calls . . . . . : 0
GHN Calls . . . . . : 0
GHNP Calls. . . . . : 0
ISRT Calls. . . . . : 0
DLET Calls. . . . . : 0
REPL Calls. . . . . : 0
Total Calls . . . . . : 10201

DL/I Message Calls:
GU Calls. . . . . : 0
GN Calls. . . . . : 0
ISRT Calls. . . . . : 0
PURG Calls. . . . . : 0
CMD Calls . . . . . : 0
GCMD Calls. . . . . : 0
CHNG Calls. . . . . : 0
AUTH Calls. . . . . : 0
SETO Calls. . . . . : 0

```

図 122. IMS Information 画面の例 (4/4)

```

DL/I System Service Calls:
APSB Calls. . . . . : 0
DPSB Calls. . . . . : 0
GMSG Calls. . . . . : 0
ICMD Calls. . . . . : 0
RCMD Calls. . . . . : 0
CHKP Calls. . . . . : 0
XRST Calls. . . . . : 0
ROLB Calls. . . . . : 0
ROLS Calls. . . . . : 0
SETS Calls. . . . . : 0
SETU Calls. . . . . : 0
INIT Calls. . . . . : 0
INQY Calls. . . . . : 0
LOG Calls . . . . . : 0
DB DEQ Calls. . . . . : 0
VSAM I/O Count (READs) . . : 1800
VSAM I/O Count (WRITEs) . . : 0
OSAM I/O Count (READs) . . : 0
OSAM I/O Count (WRITEs) . . : 0
Total DL/I I/O Count
(OSAM+VSAM) . . . . . : 1800
Total ESAF Calls. . . . . : 0
FP FLD Calls. . . . . : 0
FP POS Calls. . . . . : 0
RLSE Calls. . . . . : 0
SAVE Calls (XQUERY) . . . : 0
RSTR Calls (XQUERY) . . . : 0
COPY Calls (XQUERY) . . . : 0
DL/I ICAL Calls . . . . . : 0

IMS Parameter Modules

Module DFSPRPX0 Address . . : 000078B0

*** Bottom of data.
F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

「IMS™ 情報」画面には次のものが表示されます。

- IMS™ 領域についての一般的な解説
- Last DL/I Call Parameter List
- すべての PCB に関する情報。

すべての PCB が、現行 (または最近使用された) PCB の識別番号とともに、相対的な PCB 番号の順に表示されま  
す。

JCB 呼び出しトレース情報が入手可能な場合、この情報が各データ・セット PCB の後に表示され、最新の呼び出し  
と最大 5 つの以前の呼び出しを示します。

- IMS™ アカウント情報
- DFSPRPX0 パラメーター・モジュールのアドレス

アドレスのポイント・アンド・シュート・フィールドを選択すると、16 進ダンプ・フォーマットでモジュールのス  
トレージを表示できます。



デフォルトでは、「IMS™ 情報」画面は“「詳細」”形式で表示されます。この形式では、すべての PCB が完全に展開されて表示されます。代わりに画面の見出しから“「Summary」”ポイント・アンド・シュート・フィールドを選択すると、[図 123 : Summary IMS Information 画面 \(1/2\) ページ 201](#) に示されている例のように、PCB が要約されて表形式で表示されます。

図 123. Summary IMS Information 画面 (1/2)

```

File View Services Help
-----
IMS Information
Command ==>
JOBNAME: IBCB0030 USER ABEND: 4036 MVS2 2019/11/29 13:51:55
Line 1 Col 1 80
Scroll ==> CSR

Summary Detail - JOBNAME: IDCB0070 SYSTEM ABEND: 0CB FAE1 2015

IMS Version . . . . . : V13R1M0
IMS Region Type . . . . . : Batch Message Processing Region
IMS Subsystem Name. . . . . : IB81
Application Program Name. . . : IDCB0070
PSB Name. . . . . : DFHSAM25

Last DL/I Call Parameter List

Note that storage addressed by individual parameters might no longer be valid.

Parameter 1 . . . . . : 0002BD68
DL/I Call Function. . . . : GU (Get Unique)

Parameter 2 . . . . . : 0001A098
(See "IMS Control Blocks" for details of this PCB)

Parameter 3 . . . . . : 0002BCA8

Parameter 4 . . . . . : 8002BD48
SSA # 1 . . . . . : PARTROOT( = )

IMS Control Blocks

Input/Output Program Communications Blocks (IOPCBs)

PCB # Address Cur PCB PCB Name Terminal Status User Group Module
# Address PCB Name ID Code ID Name Name Name
1 0001A020 IOPCB n/a DFHSAM25 n/a n/a

Data Base Program Communications Blocks (DBPCBs)

PCB # Address Cur PCB PCB Name DataBase Status Segment Process Segment Key # of
# Address PCB Name Name Code Level Options Name Length Segments
2 0001A098 *** n/a DI21PART 01 A PARTROOT 17 5

```

図 124. Summary IMS Information 画面 (2/2)

```

IMS Accounting Information

DL/I Data Base Calls:
GU Calls. . . . . : 151
GN Calls. . . . . : 10050
GNP Calls . . . . . : 0
GHU Calls . . . . . : 0
GHN Calls . . . . . : 0
GHNP Calls. . . . . : 0
ISRT Calls. . . . . : 0
DLET Calls. . . . . : 0
REPL Calls. . . . . : 0
Total Calls . . . . . : 10201
DL/I Message Calls:
GU Calls. . . . . : 0
GN Calls. . . . . : 0
ISRT Calls. . . . . : 0
PURG Calls. . . . . : 0
CMD Calls . . . . . : 0
GCMD Calls. . . . . : 0
CHNG Calls. . . . . : 0
AUTH Calls. . . . . : 0
SETO Calls. . . . . : 0
DL/I System Service Calls:
APSB Calls. . . . . : 0
DPSB Calls. . . . . : 0
GMSG Calls. . . . . : 0
ICMD Calls. . . . . : 0
RCMD Calls. . . . . : 0
CHKP Calls. . . . . : 0
XRST Calls. . . . . : 0
ROLB Calls. . . . . : 0
ROLS Calls. . . . . : 0
SETS Calls. . . . . : 0
SETU Calls. . . . . : 0
INIT Calls. . . . . : 0
INQY Calls. . . . . : 0
LOG Calls . . . . . : 0
DB DEQ Calls. . . . . : 0
VSAM I/O Count (READs) . . : 1800
VSAM I/O Count (WRITEs) . . : 0
OSAM I/O Count (READs) . . : 0
OSAM I/O Count (WRITEs) . . : 0
Total DL/I I/O Count
(OSAM+VSAM) . . . . . : 1800
Total ESAF Calls. . . . . : 0
FP FLD Calls. . . . . : 0
FP POS Calls. . . . . : 0
RLSE Calls. . . . . : 0
SAVE Calls (XQUERY) . . . : 0
RSTR Calls (XQUERY) . . . : 0
COPY Calls (XQUERY) . . . : 0
DL/I ICAL Calls . . . . . : 0

IMS Parameter Modules

Module DFSPRPX0 Address . . : 000078B0

*** Bottom of data.

```

この画面で、「PCB #」ポイント・アンド・シュート・フィールドの上にカーソルを置いて Enter キーを押すと、個々の PCB を選択できます。図 125 : Detailed PCB 画面 ページ 203 に、詳細 PCB 画面の例が示されています。

図 125. Detailed PCB 画面

```

File View Services Help
-----
IMS Input/Output Program Communications Block (IOPCB)..          Line 1 Col 1 80
Command ==>                                                    Scroll ==> CSR
JOBNAME: IDCBO070  SYSTEM ABEND: 0CB  FAE1  2019/01/19 14:17:15

IOPCB:
At Address. . . . . : 0001A020
PCB Name. . . . . : IOPCB
Relative PCB Number . . . . : 1
PCB Type. . . . . : I/O
Logical Terminal ID . . . . : n/a
Status Code . . . . . : ' ' (Normal status)
User ID . . . . . : DFHSAM25
Group Name. . . . . : n/a
Formatting Module Name. . . : n/a

*** Bottom of data.

```

この画面からアドレスのポイント・アンド・シュート・フィールドを選択すると、[図 143: サンプル「Dump Storage」抑止画面](#) ページ 218 に表示されている例のように、「Dump Storage」画面に PCB データ域が表示されます。

## 基本オプション: ストレージ域

“ストレージ域”オプションを選択すると、イベントに関するフォーマット済みストレージ域へのリンク、16 進ダンプ・ストレージ域へのリンク、また COBOL の場合は、DSA チェーンに存在しなくなったプログラム用の静的ストレージに関する情報へのリンクが、[図 126: System-Wide Storage Areas 画面の例](#) ページ 203 の例のように提供されます。

図 126. System-Wide Storage Areas 画面の例

```

File View Services Help
-----
System-Wide Storage Areas          Line 1 Col 1 80
Command ==>                        Scroll ==> CSR
TRANID: CD01  CICS ABEND: DSNC  CICS41  2019/04/28 13:22:19

Select one of the following:
  1. Event 1 Program COBMST3 Storage Areas
  2. Hex-Dumped Storage

The following list of COBOL programs have been called and returned during the
current execution, but do not have active register save areas:
  3. Module COBMST3 Program COBFIL2 Static Storage
  4. Module COBMST3 Program COBSUB2 Static Storage

*** Bottom of data.

```

以下のいずれかを行って、この画面からいずれかのイベント関連ポイント・アンド・シュート・リンク (上の「Event 1 Program COBMST3 Storage Areas」リンクなど) を選択します。

- コマンド行でオプション番号を入力する。
- オプション番号のポイント・アンド・シュート・フィールドにカーソルを置いて Enter を押す。

この結果の画面は、イベントの詳細セクションから「Associated Storage Areas」リンクを選択した場合に表示される画面に似たものになります。追加情報と例については、[関連したストレージ域の表示 ページ 208](#)を参照してください。

「Hex-Dumped Storage」リンクは、1つ以上のイベントに対するフォーマットされていない関連ストレージ域を表示します。Hex-Dumped Storage 画面の例を [図 127: Hex-Dumped Storage 画面の例 ページ 204](#) に示します。

図 127. Hex-Dumped Storage 画面の例

```

File View Services Help
-----
Hex-Dumped Storage                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: CD01      CICS ABEND: DSNC                CICS41    2019/04/28 13:22:19

Address  Offset   Hex                                     EBCDIC / ASCII
-----
Event 4 Program CDCB0010 BLL=0001 (Address 001410D0)
See Event 4 Program COBMAIN Storage Areas for address range
001410D0-001420CF formatted storage

Event 4 Program CDCB0010 GPR 12 (Address 18206138)
18206138          +8  18206C90 18207C90  00000800 00000000 * .....*
18206140          +18 00000000 00000000 00000000 00000000 *..%...@.....*
18206150          +18 00000000 00000000 00000000 00000000 *.....*
Lines 18206160-18206190 same as above
182061A0          +68 00000000 00000000 00000000 80028B20 *.....*
182061B0          +78 00000000 00000000 00000000 00000000 *.....*
Lines 182061C0-18206240 same as above
18206250          +118 00000000 00000000 18203FF0 00000000 *.....0....*
18206260          +128 00000000 00000000 00000000 00000000 *.....*
Lines 18206270-182062C0 same as above
F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

## 基本オプション: Java Information

[Java 分析の実行 ページ 262](#)を参照してください。

## 基本オプション: Language Environment Heap Analysis

「*Language Environment@ Heap Analysis*」オプションを選択すると、LE ヒープに関連する要約情報が表示されます。 [図 128: Language Environment Heap Analysis の概要画面の例 ページ 204](#)には、概要画面の例が示されています。

図 128. Language Environment Heap Analysis の概要画面の例

```

File View Services Help
-----
Language Environment Heap Analysis               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
Summary Full -  JOBNAME: LEHEAP      SYSTEM ABEND: 0C4      FAE1      2019/0

Segment  Segment      Segment  Root      Root      No.Free  No.Alloc  Alloc
Address  Type                Length   Address   Length   Elements Elements  Bytes
17E59018 User Heap          00008000 17E59E68 000071B0         1         5 00000D50 E
17E35000 Anywhere Heap    00004000 17E37580 00000E20         4         30 00002DA8
17E7B000 Anywhere Heap    00002000 17E7B4E8 00001A80         3         10 00000440
17E7D000 Anywhere Heap    00004A50 00000000 00000000         0         1 00004A30

Additional Heap Control Blocks

There are no additional heaps

*** Bottom of data.

```

この画面から、**Segment Address**ポイント・アンド・シュート・フィールドを使用して、個別のヒープ・セグメントを表示します。**Segment Type**ポイント・アンド・シュート・フィールドを使用して、指定したタイプのすべてのセグメントを表示します。

**Full**ポイント・アンド・シュート・フィールドを選択して、LE ヒープに関する情報をすべて表示します。

図 129. Language Environment Heap Analysis の全画面の例

```

File View Services Help
-----
Language Environment Heap Analysis                               Line 1 Col 1 80
Command ==>----- Scroll ==> CSR
Summary Full --JOBNAME: LEHEAP  SYSTEM ABEND: 0C4             FAE1      2019/0

Errors were found in one or more segments

Enclave-Level Storage
  Management (ENSM) Address : 17613268
Heap allocation
  initialization value
  specified . . . . . : No
Heap free initialization
  value specified . . . . . : No

User Heap Analysis

Heap Anchor Node (HANC) . . : 17E59018
Heapid. . . . . : 00000000
Root Address. . . . . : 17E59E68
Segment Length. . . . . : 00008000
Root Length . . . . . : 000071B0

```

必要に応じて、スクロール・コマンド、UP (PF7) および DOWN (PF8)、LEFT (PF10) および RIGHT (PF11) を使用し、画面全体を表示します。

## 基本オプション: MTRACE レコード

“ [MTRACE レコード] ”オプションを選択すると、[図 130 : MTRACE Records 画面の例](#) ページ 206の例のように、MVS™ マスター・トレースの画面が表示されます。

図 130. MTRACE Records 画面の例

File View Services Help									
MTRACE Records									
Command ==> _____ Line 1 Col 1 80									
Scroll ==> CSR									
SLIP DUMP ID=F092									
M	0100000	F AE1	06233	18:10:40.30	<b>STC21507</b>	00000090	F AE1	2019/08/21	18:11:11
D					016	00000090		IST530I	AM GBIND PENDI
D					016	00000090		IST1051I	EVENT CODE =
E					016	00000090		IST1062I	EVENT ID = 00
E					016	00000090		IST314I	END
M	0100000	F AE1	06233	18:10:40.30	<b>STC21507</b>	00000090		IST530I	AM GBIND PENDI
D					017	00000090		IST1051I	EVENT CODE =
D					017	00000090		IST1062I	EVENT ID = 00
E					017	00000090		IST314I	END
N	4000000	F AE1	06233	18:10:40.63	<b>JOB08017</b>	00000090		IEF677I	WARNING MESSAG
N	0020000	F AE1	06233	18:10:40.67	<b>JOB08017</b>	00000090		ICH70001I	ANDYMEL LAS
N	4000000	F AE1	06233	18:10:40.68	<b>JOB08017</b>	00000090		\$HASP373	DACBB012 STAR
N	0000000	F AE1	06233	18:10:40.71	<b>JOB08017</b>	00000090		IEF403I	DACBB012 - STA
N	0004000	F AE1	06233	18:10:40.99	<b>JOB08017</b>	00000290		-	-
-PAGING COUNTS---									
N	0004000	F AE1	06233	18:10:40.99	<b>JOB08017</b>	00000290		-JOBNAME	STEPNAME PRO
AGE SWAP VIO SWAPS STEPNO									
N	0004000	F AE1	06233	18:10:41.01	<b>JOB08017</b>	00000290		-DACBB012	V00
	0	0	0	0	1				
N	0004000	F AE1	06233	18:10:41.25	<b>JOB08017</b>	00000290		-DACBB012	D00

「MTRACE レコード」画面は、必要なストレージ域が含まれる MVS™ ダンプの分析を実行している場合のみ使用可能になるので、注意してください。

カーソルをポイント・アンド・シュート・フィールドであるジョブ ID のどれか 1 つに置いて、Enter キーを押すと、選択されたジョブ ID の項目のみが含まれる MTRACE 画面が表示されます。完全な MTRACE に戻るには、PF3 を押してください。

## 基本オプション: Abend Job Information

対話式レポートの初期表示画面から「Abend Job Information」オプションを選択すると、[図 131: Abend Job Information 画面の例](#) ページ 207 に示す例のように、レポートの「Abend Job Information」セクションが表示されます。

図 131. Abend Job Information 画面の例

```

File View Services Help
-----
Abend Job Information                                     Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: FRED          CICS ABEND: AEIL                 2019/05/24 13:49:18

IBM Fault Analyzer Abend Job Information:

Abend Date. . . . . : 2002/05/24
Abend Time. . . . . : 13:49:18
System Name . . . . . : n/a
Job Type. . . . . : CICS Transaction
Job ID. . . . . : STC01869
Job Name. . . . . : CICS04
Job Step Name . . . . . : CICS04
ASID. . . . . : 33
Job Execution Class . . . . : n/a
Region Size . . . . . : 4M
EXEC Program Name . . . . . : DFHSIP
User ID . . . . . : CICSUSER
Accounting Information. . . : n/a

Data Sets:
  DDname  Data Set or Path Name

```

この画面には、障害がリアルタイムで分析されたときに存在していた環境についての情報が表示されます。表示される情報は、分析された障害のタイプによって異なります。

## 基本オプション: ユーザー・ノート

初期対話式レポートから「User Notes (ユーザー・ノート)」オプションを選択する操作は、NOTELIST コマンドを発行する操作に相当し (詳しくは、[NOTELIST ページ 114](#) を参照)、その結果として User Note List 画面が表示されます ([図 144: User Note List 画面の例 ページ 222](#) を参照)。

「User Notes」オプションは、指定された障害項目に 1 つ以上のユーザー・ノートが存在する場合は必ず、Interactive Reanalysis Report 画面に動的に追加されます。

## 基本オプション: Fault Analyzer オプション

対話式レポートの初期表示画面から「*Fault Analyzer*」オプションを選択すると、[図 132: 「Fault Analyzer オプション」画面の例 ページ 208](#)のように、レポートの「Fault Analyzer」セクションが表示されます。

図 132. 「Fault Analyzer オプション」画面の例

```

File View Services Help
-----
Fault Analyzer Options                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: FRED          CICS ABEND: AEIL              CICS04   2019/09/25 11:06:56

IBM Fault Analyzer Options in Effect:

{These are the options that were used to generate the current interactive
reanalysis report. To change any options, first return to the Fault Entry List
display and select "Interactive Reanalysis Options" from the "Options"
action-bar pull-down menu; then perform interactive reanalysis again.}

FaultID(F00066)
Language(ENU)
NoLocale

Data Sets:

{The following Fault Analyzer data set or path names were either
preallocated, specified via DataSets options, or provided as defaults.}

DDname   Data Set or Path Name
-----
IDIADATA DA.SYSADATA

```

## 関連したストレージ域の表示

対話式レポートでは、使用するプログラム言語に応じて、詳細イベント画面に「関連したストレージ域」リンクが表示されます。このリンクを選択するときに表示される内容は、プログラム言語に応じて、およびプログラム用に Fault Analyzer がコンパイラー・リストまたはサイド・ファイルを使用可能であるかどうかに応じて異なります。

- コンパイラー・リストもサイド・ファイルもない COBOL プログラムの場合、ベース・ロケーターが 16 進ダンプ・フォーマットで表示されます。
  - COBOL V4 以前の場合、TGT はフォーマット設定されます。
  - COBOL V5 以降の場合、PPA はフォーマット設定されます。
- コンパイラー・リストまたはサイド・ファイルが提供されている COBOL プログラムの場合、すべてのフィールドのソース宣言が、現行の内容とともに提示されます。

図 133 : Associated Storage Areas 画面の例 ページ 209 では、ソース・リストが提供されている COBOL プログラムの Associated Storage Areas 画面の例を示しています。



図 133. Associated Storage Areas 画面の例

File View Services Help			
Associated Storage Areas			Line 1 Col 1 80
JOBNAME: COBLVL88 SYSTEM ABEND: 0CB FAE1			2021/09/20 15:48:38
<b>Program Prolog Areas</b>			
WORKING-STORAGE SECTION			
- Collapse hex - Collapse level 88			
Offset	Hex Value	Data Value	Source (Starti
STATIC storage at address 0000B3F0			
A8	E6D6D9D2 C9D5C760 E2E3D6D9 C1C7C540	*WORKING-STORAGE	* 01 FILLER
B8	40404040	*	*
<b>Suppressed Copybook</b>			
390	00000000	*....	* 01 FIELD-1
398	00000000	0	01 FIELD-2
3A0	00000000 0000	*.....	* 01 FIELD-3
			01 TABLE-4.
			03 TABLE
			04 TABL
3A8	426F1C29	1.111100e+02	05 ELEM
3AC	00000000	*....	* 05 ELEM
			88 J
			88 J
			ELEM
3BC	00000000	*....	* 05 ELEM

表示された情報をスクロールダウンすると、現行のプログラムに応じて、「Linkage Section」情報、「File Section」情報なども表示されます。

対話式再分析レポートに固有の、次のようないくつかの機能があります。

- 16 進値列を非表示にする機能。
- レベル 88 の項目を省略する機能。
- すべての COBOL ベース・ロケータを表示する機能。

これらについては、この後のトピックで説明します。

## 16 進値列の非表示

16 進値列の非表示は、狭い (80 列) 画面のユーザーにとって役立つことがあります。図 133 : Associated Storage Areas 画面の例 ページ 209 の「16 進値 (Hex Value)」見出しの上にある負符号 (-) にカーソルを合わせて Enter キーを押すと、図 134 : 16 進値列が縮小表示された Associated Storage Areas 画面の例 ページ 210 に示されている画面に変わります。

図 134. 16 進値列が縮小表示された Associated Storage Areas 画面の例

```

File View Services Help
Associated Storage Areas                               Line 1 Col 1 80
JOBNAME: COBLVL88  SYSTEM ABEND: 0CB                FAE1      2021/09/20 15:48:38

Program Prolog Areas
WORKING-STORAGE SECTION
+ Expand hex    - Collapse level 88
  Offset Data Value      Source (Starting at Line # 000010 )
STATIC storage at address 0000B3F0
  A8 *WORKING-STORAGE * 01 FILLER                      PIC X(20)  VALUE 'WORK
  B8 *
Suppressed Copybook
  390 *....          * 01 FIELD-1                      PIC 999999 COMP-3.
  398 0              01 FIELD-2                      PIC 999999 COMP-4.
  3A0 *.....        * 01 FIELD-3                      PIC 999999.
                        01 TABLE-4.
                        03 TABLE-8 OCCURS 6 TIMES.
                        04 TABLE-8A OCCURS 3 TIMES.
  3A8 1.111100e+02   05 ELEMENT-1                      COMP
  3AC *....          * 05 ELEMENT-2 OCCURS 4 TIMES     PIC
                        88 JACK VALUE 'JACK'.
                        88 JILL VALUE 'JILL'.
                        ELEMENT-2(1,1,2) to ELEMENT-2(1,1,4) sam
  3BC *....          * 05 ELEMENT-3                      PIC 999999 CO

```

ここで、「Data Value」列の上にある負符号 (-) のポイント・アンド・シュート・フィールドが、正符号に変わっています。このポイント・アンド・シュート・フィールドにカーソルを合わせると、「16 進値 (Hex Value)」が可視だった最初の画面に戻ります。

最後に選択した「16 進数の縮小表示/拡張 (Collapse/Expand hex)」オプションは、ユーザーの ISPF プロファイルに保管され、それ以降に Associated Storage Areas 画面を開くときのデフォルト表示モードとして使用されます。

## レベル 88 項目の縮小表示

レベル 88 項目の縮小表示は、潜在的に多数あるが役立つことが少ない宣言を画面から抑止するのに使用できます。図 133: Associated Storage Areas 画面の例 ページ 209 の「レベル 88 項目の縮小表示」見出しの直前にある負符号 (-) にカーソルを合わせて Enter キーを押すと、図 135: レベル 88 項目を縮小表示した「Associated Storage Areas」画面の例 ページ 211 に示されている画面に変わります。

図 135. レベル 88 項目を縮小表示した「Associated Storage Areas」画面の例

```

File View Services Help
Associated Storage Areas                               Line 1 Col 1 80
JOBNAME: COBLVL88  SYSTEM ABEND: 0CB                FAE1      2021/09/20 15:48:38

Program Prolog Areas
WORKING-STORAGE SECTION
+ Expand hex    + Expand level 88
Offset Data Value Source (Starting at Line # 000010 )
STATIC storage at address 0000B3F0
  A8 *WORKING-STORAGE * 01 FILLER                    PIC X(20)  VALUE 'WORK
  B8 *
Suppressed Copybook
  390 *....          * 01 FIELD-1                    PIC 999999 COMP-3.
  398 0              * 01 FIELD-2                    PIC 999999 COMP-4.
  3A0 *.....        * 01 FIELD-3                    PIC 999999.
                      * 01 TABLE-4.
                      03 TABLE-8 OCCURS 6 TIMES.
                      04 TABLE-8A OCCURS 3 TIMES.
  3A8 1.111100e+02  * 05 ELEMENT-1                    COMP
  3AC *....          * 05 ELEMENT-2 OCCURS 4 TIMES    PIC
Level 88 Items
                      ELEMENT-2(1,1,2) to ELEMENT-2(1,1,4) sam
  3BC *....          * 05 ELEMENT-3                    PIC 999999 CO
  3C0 0              * 05 ELEMENT-4                    PIC 999999 CO

```

ここで、「Expand level 88」見出しの直前にある"負符号(-)のポイント・アンド・シュート・フィールドが正符号に変わっていること、およびこのポイント・アンド・シュート・フィールドにカーソルを合わせると、レベル 88 項目がインラインになっている最初の画面に戻ることに注意してください。

「Level 88 Items」ポイント・アンド・シュート・フィールドにカーソルを合わせて Enter キーを押すと、結果は [図 136: レベル 88 項目画面の例 ページ 211](#) に示す画面になります。

図 136. レベル 88 項目画面の例

```

File View Services Help
Level 88 Items                                       Line 1 Col 1 80
JOBNAME: COBLVL88  SYSTEM ABEND: 0CB                FAE1      2021/09/20 15:48:38

Source
  88 JACK VALUE 'JACK'.
  88 JILL VALUE 'JILL'.

*** Bottom of data.

```

associated storage areas 画面に戻るには PF3 を押します。

最後に選択した「Collapse/Expand level 88」オプションは、ユーザーの ISPF プロファイルに保管され、それ以降に「Associated Storage Areas」画面を開くときのデフォルト表示モードとして使用されます。

## すべての COBOL ベース・ロケーターを表示

ソース・コードがイベントに使用できず、複数の連続するベース・ロケーターが同じストレージ・タイプ (例えば WORKING-STORAGE) で存在する場合、これらは初めに要約されて表示されます。この場合、開始アドレスが示され、長さはそれに続く連続するすべてのベース・ロケーターの合計長です。

「Show all BLs」の前にある正符号にカーソルを置いて Enter を押します。画面が変わり、すべてのベース・ロケータが、それぞれの開始アドレスおよび長さとともに表示されます。同時に、画面上部の正符号が負符号に変わり、テキストが「Summarize BLs」に変わります。

デフォルトの要約ベース・ロケータ画面に戻すには、「Summarize BLs」の前にある負符号にカーソルを置いて Enter を押します。

## メッセージおよび異常終了コードの拡張表示

Fault Analyzer の対話式ダンプ再分析機能を使用するとき、メッセージまたは異常終了コードは初めは拡張表示されていません。拡張表示をしていないことにより、レポート項目を表示するために、その後ろに続く非常に長い可能性がある説明をスクロールする必要がなくなります。その代わりに、対話式レポートでメッセージや異常終了コードの説明を表示するには、メッセージ ID または異常終了コードにカーソルを置いて、Enter キーを押します。これで、[図 137 : Message Explanation 画面の例](#) ページ 212 の例に示されているように、バッチ・レポートと同様の内容が開きます。

図 137. Message Explanation 画面の例

```

File View Services Help
-----
Message Explanation                               Line 1 Col 1 80
Command ==>                                       Scroll ==> CSR
JOBNAME: DACBB045  USER ABEND: 4038             MVS2      2019/10/10 10:38:22

IGZ0035S There was an unsuccessful OPEN or CLOSE of file INDD in program
          DAVSA009 at relative location X'0380'.  Neither FILE STATUS nor an
          ERROR declarative were specified. The status code was 39. ❶

IGZ0035S
IGZ0035S There was an unsuccessful OPEN or CLOSE of file file-name in
          program program-name at relative location location. Neither FILE
          STATUS nor an ERROR declarative were specified. The status code was
          status-code.

Explanation: An error has occurred while opening or closing the named
          file. No file status or user error declarative was specified.

Programmer Response: Check to make sure there is a DDname defined for the
          indicated file.

F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

メッセージがレポートに表示されると、最初に現れるメッセージには、その発行時の実際のテキストが含まれます(上記画面の例の ❶ を参照してください)。インスタンス固有のテキストがなければ、コメント「[*job-specific text not available*]」がこれに置き換わります。

発行されたメッセージまたは異常終了コードの直後の拡張部分には、通常、Fault Analyzer とともに提供されるソフトコピーにある説明が含まれます。メッセージまたは異常終了コードの説明が見つからない場合は、テキスト「[*explanation not available*]」がこれに置き換わります。

## ソース・コードの表示

プログラム全体のソース・コードを表示するには、任意の黄色いポイント・アンド・シュート・ソース行番号またはリスト・ステートメント番号の上にカーソルを置き、Enter キーを押します。例えば、対話式レポート内のイベントから行番号 69 が選択された場合、[図 138 : Compiler Listing 画面の例](#) ページ 213 の画面が表示されます。

図 138. Compiler Listing 画面の例

```

File View Services Help
-----
Program CICFRED Compiler Listing                               Line 63 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: FRED          CICS ABEND: AEIL          CICS04      2019/01/09 15:37:46
000060              Move length of MSG1 to dfhb0020
000061              Call 'DFHEI1' using by content x'04043000070000008100004000
000062      -      '00f0f0f0f1f8404040' by content x'0000' by content x'0000'
000063              reference MSG1 by reference dfhb0020 end-call.
000065              ADD 1 TO COUNTER.
000066      *EXEC CICS READ FILE('NOTTHERE') RIDFLD(RID)
000067      *      INTO(REC-AREA) END-EXEC.
000068              Move length of REC-AREA to dfhb0020
000069      Call 'DFHEI1' using by content x'0602f0000700008000f0f0f2
000070      -      '404040' by content 'NOTTHERE' by reference REC-AREA by
000071              reference dfhb0020 by reference RID end-call.
000073      *EXEC CICS SEND FROM(MSG1) END-EXEC.
000074              Move length of MSG1 to dfhb0020
000075              Call 'DFHEI1' using by content x'04043000070000000100004000
000076      -      '00f0f0f0f2f2404040' by content x'0000' by content x'0000'
000077              reference MSG1 by reference dfhb0020 end-call.
000079      *EXEC CICS RETURN END-EXEC.
F1=Help           F3=Exit           F5=RptFind       F6=AddBkp       F7=Up           F8=Down
F10=Left          F11=Right

```

最初に選択されたソース行番号またはステートメント番号が、強調表示されます。

ここに示されている例では、疑似アセンブラー命令の表示は抑止されていると想定しています。この抑止については詳しくは、[ソース・コードの表示 ページ 212](#)を参照してください。

青色で表示されている情報(すべての行が列1から始まる)は、プログラム・ソース・コードに関連しています。画面の左側には、表示されているソースのソース行またはリスト・ステートメント番号が表示されます。この情報の後に、プログラム内のこの位置の実際のソース・コードが続きます。

リストにマシン・インストラクション情報を追加するには、「View」メニューの「疑似アセンブラー・インストラクションの追加」オプションを選択します(このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。このオプションを選択すると、[図 139: 「コンパイラー・リスト」画面の例: 疑似アセンブラー命令が有効 ページ 214](#)に示されているように、Compiler Listing 画面が再フォーマットされ、プログラムのソース・コードに疑似アセンブラー命令が挿入されます。

図 139. 「コンパイラー・リスト」画面の例: 疑似アセンブラー命令が有効

```

File View Services Help
-----
Program CICFRED Compiler Listing                               Line 316 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: FRED          CICS ABEND: AEIL          CICS04      2019/01/09 15:37:46
000003DA 1845          LR      R4,R5
000003DC 8E40 0020      SRDA   R4,32
000003E0 5D40 C000      D      R4,0(,R12)
000003E4 4040 8008      STH    R4,8(,R8)
000066      *EXEC CICS READ FILE('NOTTHERE') RIDFLD(RID)
000067      * INTO(REC-AREA) END-EXEC.
000068      Move length of REC-AREA to dfhb0020
000003E8 D201 8088 A01C MVC    136(2,R8),28(R10)
000069      Call 'DFHEI1' using by content x'0602f0000700008000f0f0f2
000070      - '404040' by content 'NOTTHERE' by reference REC-AREA by
000071      reference dfhb0020 by reference RID end-call.
000003EE D210 D0B8 A061 MVC    184(17,R13),97(R10)
000003F4 4130 D0B8      LA     R3,184(,R13)
000003F8 5030 D0A0      ST     R3,160(,R13)
000003FC D207 D0D0 A08E MVC    208(8,R13),142(R10)
00000402 4130 D0D0      LA     R3,208(,R13)
00000406 5030 D0A4      ST     R3,164(,R13)
F1=Help      F3=Exit      F5=RptFind  F6=AddBkp  F7=Up      F8=Down
F10=Left     F11=Right

```

緑色で表示されている情報(すべての行が列 15 から始まる)は、マシン命令の逆アセンブルに関連しています。この情報には、属するコードの行の後に、ソース・コード情報が散在しています。



**注:** TEST(NONE,SYM,SEPARATE) を使用してコンパイルされた COBOL プログラムでは、ソース・コードの最終行の後にすべての疑似アセンブラー命令が配置されます。

それ以外のすべてのプログラムでは、リスト画面に疑似アセンブラー命令を追加する前に、必ず画面をスクロールして関心のあるソース行を画面の最上部に置くようにします。追加の表示行により、ソース行が現在のビューに表示されなくなる可能性があります。

この画面から疑似アセンブラー命令を除去するには、「View」メニューの「疑似アセンブラー・インストラクションの除去」オプションを選択してください。

リストの上部の情報は、コンパイラー・リストまたはサイドのソースを示しています。図 140: 「コンパイラー・リスト」画面の例: 起点情報 ページ 215 に例を示します。例えば UP MAX コマンドを入力すると、サイド・ファイル・リストの上部にスクロールできます。

図 140. 「コンパイラー・リスト」画面の例: 起点情報

```

File View Services Help
-----
Program CICFRED Compiler Listing                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
TRANID: FRED          CICS ABEND: AEIL          CICS04      2019/01/09 15:37:46

The listing or side file used for the following was found in
DA.LISTING.COBOLE(CICFRED).

Source
Line #
000001          *****
000002          * TRANSACTION: FRED                                *
000003          *   EXPECTED OUTPUT:                                *
000004          *   'STARTED CICFRED' FOLLOWED BY AEIL ABEND      *
000005          *****
000006          IDENTIFICATION DIVISION.
000007          PROGRAM-ID. CICFRED.
000008          ENVIRONMENT DIVISION.
000009          DATA DIVISION.
000010          WORKING-STORAGE SECTION.
000011          77 UPPER-LIMIT PIC S9(4) COMP VALUE 255.
F1=Help      F3=Exit      F5=RptFind      F6=AddBkp      F7=Up      F8=Down
F10=Left     F11=Right

```

## 据え置きブレークポイント機能

コンパイラー・リスト表示では、IPVLANGP によく似た z/OS® Debugger 据え置きブレークポイントの設定がサポートされています。詳しくは、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*を参照してください。

据え置きブレークポイントを設定するには、ADFz 共通コンポーネントをインストールする必要があります。

## 保管場所の表示

保管場所を表示するには、任意の黄色いポイント・アンド・シュート・アドレス (例えば、レジスター値) にカーソルを置いて、Enter キーを押します。また、SHOW コマンドを使用することもできます (詳細については、[SHOW ページ 116](#)を参照してください)。

図 141 : Dump Storage 画面の例 ページ 216 には、ストレージ画面の例が示されています。

図 141. Dump Storage 画面の例

```

File View Services Help
-----
Dump Storage                               17C01380-17C013D7
Command ==>                               Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7      MVS2      2019/08/12 13:46:58

Address  Offset      Hex                                     EBCDIC / ASCII
-----
Module IDISCBL1 program IDISCBL1 + X'640', source line # 32
Event 1 Program IDISCBL1 GPR 1 + X'1051'
Event 1 Program IDISCBL1 GPR 10 + X'1300'
Event 1 Program IDISCBL1 GPR 11 + X'1008'
Event 1 Program IDISCBL1 GPR 12 + X'1348'
Event 1 Program IDISCBL1 GPR 14 + X'680'
Event 1 Program IDISCBL1 GPR 3 + X'600'
Event 1 Program IDISCBL1 GPR 4 + X'1544'
17C01380      20004160 61345060 5108D203 510C310A *...-/.&-..K.....*
17C01390      +10  D2035110 310E5860 20085060 5114D20B *K.....-&-..K.*
17C013A0      +20  51183112 58602004 50605128 58602008 *.....-&-..K.....*
17C013B0      +30  5060512C D2175130 311E98EC D00C07FE *&-..K.....q.}....*
Module IDISCBL1 CSECT CEESG005
17C013C0      +40  E2F0F0F5 00140001 00000000 00000000 *S005.....*
17C013D0      +50  00000000 00000000 *.....*
Module IDISCBL1 CSECT CEEBETBL
F1=Help  F3=Exit  F7=Up    F8=Down  F10=Prev  F11=Next

```

16進データの右側にある文字で表現されるセクションは、一般に EBCDIC で表示されます。代わりにデータを ASCII で表示するには、カーソルを ASCII ポイント・アンド・シュート・フィールドに置いて、Enter を押します。それからカーソルを EBCDIC ポイント・アンド・シュート・フィールドに置いて Enter をもう一度押すと、EBCDIC の表示に戻ります。

16進ストレージ表示域内の任意の場所にカーソルを置いて Enter キーを押すと、選択されたアドレスが表示されます。カーソルを置いたポイント・アンド・シュート・フィールドが 8 桁未満の場合は、8 桁の 31 ビット・アドレスになるように、先行ゼロが埋め込まれます。

8 桁のアドレスのポイント・アンド・シュート・フィールドで、最初の 2 桁にゼロを上書き入力してから直ちに Enter キーを押すと、アドレスは 24 ビット・アドレスとして解釈されるようになります。

64 ビット・アドレスを選択するには、64 ビット・アドレスの前半部 (ビット 0 から 31) を表しているポイント・アンド・シュート・フィールドの最後の桁に上書き入力するか、64 ビット・アドレスの後半部 (ビット 32 から 63) を表しているポイント・アンド・シュート・フィールドの最初の桁に上書き入力してから、Enter を押します。このプロセスにより、下線に最も近いポイント・アンド・シュート・フィールドが、下線が置かれているフィールドと論理的に「結合」されます。例えば、次のような 2 つの隣接するアドレスのポイント・アンド・シュート・フィールドがあるとします

```
00000001 80109020
```

これに、次のように最初のフィールドの最後の桁に上書き入力するか、

```
0000000_ 80109020
```

または、次のように 2 番目のフィールドの最初の桁に上書き入力すると、

```
00000001 _0109020
```

結果として 64 ビット・アドレスの 00000001\_80109020 が表示されます。31 ビット・アドレスの場合のように、64 ビット・アドレスの後半部には、8 桁になるように先行ゼロが埋め込まれます。



表示される最後の 10 個のアドレスのレコードが保持されます。PREV (PF10) または NEXT (PF11) コマンドを使用すると、以前に選択された領域を再表示できます。

表示される 1 行当たりのバイト数は、設定済みの現行のフォーマット幅ではなく、可視幅によって決まります。可視幅に許可されると、1 行当たり 32 バイトが表示され、そうでなければ 16 バイトが表示されます。

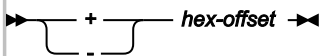
表示可能な場合、モジュール、プログラム、およびユーザー・ノート (図 141 : Dump Storage 画面の例 ページ 216 参照) など他のストレージ域の最初の説明とともに、最初に選択されたアドレスの説明が表示されます。

図 141 : Dump Storage 画面の例 ページ 216 の Event 1 Program IDISCBL1 GPR 1 + X'1051' などを含む説明内のイベント番号は、フル・イベント・サマリー内のイベント番号を参照することに注意してください。イベント・サマリーで Application Only 設定が有効な場合、イベント番号はストレージの説明と一致しないことがあります。異なるイベント・サマリリーの設定について詳しくは、基本オプション: イベント・サマリー ページ 175 を参照してください。

この画面で使用される FIND コマンドは、フォーマットされた画面そのものではなくミニダンプを検索対象とするため、他のすべての画面と異なる動作を行います。詳しくは、画面のタイプによる FIND コマンドの差異 ページ 110 を参照してください。

現行画面のストレージの前後にストレージを表示するには、必要に応じて UP/DOWN コマンド (通常は PF7/PF8 にマップされている) を使用します。また、次の形式でコマンド行にオフセットを入力することもできます。

図 142. Syntax



この *hex-offset* は、表示される左上のアドレスに対応する 16 進オフセットです。例:

+10C

または

-D4

## データ域の表示

対話型再分析レポート内からアドレスのポイント・アンド・シュート・フィールドを選択する場合、結果は一般的にすべての利用可能なストレージが表示されますが、選択したアドレスはオフセット 0 にあります。ただし、アドレスが指定された長さのデータ領域に関連付けられている場合、結果には、暗黙の長さのストレージのみが最初に表示され、それ以前または以降のすべてのストレージが「抑止されて」表示されます。この例を 図 143 : サンプル [Dump Storage] 抑止画面 ページ 218 に示します。

図 143. サンプル 「Dump Storage」 抑止画面

File View Services Help						
Dump Storage						17C01380-17C013D7
Command ==>						Scroll ==> CSR
<b>ShowAll</b> - JOBNAME: IDCB0070 SYSTEM ABEND: 0CB						FAE1 2019/01/19
Address	Offset	Hex	EBCDIC / ASCII			
Address range 00000000-0001A01F suppressed						
PCB #1 - Start						
0001A020		00400038 00010018 40404040 00000000	*	.	.	.*
0001A030	+10	008A1054 00000000 12A74084 C9D6D7C3	*	.	.	x dIOPC*
0001A040	+20	C2404040 00000000 00000000 00000000	*	B	.	.....*
0001A050	+30	00000000 0001A020	*	.	.	.*
Event 5 Program IDCB0070 BLL=00001						
0001A058	+38	40404040 40404040	*	.	.	.*
0001A060	+40	10004040 0000000F 0000000F 00000000	*	.	.	.....*
0001A070	+50	00000000 00000000 C4C6C8E2 C1D4F2F5	*	.	.	DFHSAM25*
0001A080	+60	40404040 40404040 00000000	*	.	.	.....*
PCB #1 - End						
Address range 0001A08C-FFFFFFFF_FFFFFFFF suppressed						

使用可能なすべてのストレージを参照するには、画面タイトルにある「ShowAll」ポイント・アンド・シュート・フィールドにカーソルを置いて Enter を押します。

または、任意の 16 進アドレス・ポイント・アンド・シュート・フィールドを選択しても、使用可能なストレージがすべて表示されます。

「ShowAll」の方式が使用された場合は、PF3 を押して元のアドレス・ポイント・アンド・シュート・フィールドを再選択する方法以外では、前後のストレージを再び抑止することはできません。

一方、任意の 16 進アドレス・ポイント・アンド・シュート・フィールドを選択する方式が使用された場合、アドレスは通常どおり「ダンプ・ストレージ」画面に「スタック」されるため、元のアドレスが再び表示されるまで PF10 を押せば、前後のストレージは再び抑止されます。この方式は、10 個を超える後続のアドレスが選択された場合を除いて使用可能です。

## ユーザー・ノート の作成と管理

ユーザー・ノートは、対話式ユーザーが任意の保管場所に対して追加できるコメントです。ヒストリー・ファイルの障害項目に保管され、すべてのユーザーが使用できます。

ユーザー・ノートは、Interactive Reanalysis Report 画面から NOTE コマンドを使用して作成できます。詳しくは、[NOTE ページ 113](#)を参照してください。

または、次の画面を上書きすることにより、ユーザー・ノートを作成することができます。

- ダンプ・ストレージ (Dump Storage)
- 関連ストレージ域 (Associated Storage Areas)
- イベント関連ストレージ域 (イベント *n* プログラム *name* のストレージ域)
- 16 進ダンプ・ストレージ (Hex-Dumped Storage)

以下では、これらを簡単に「ストレージ域」画面と呼びます。

ノートが表示されるストレージ域にカーソルを合わせ、16進数字と区別可能な1つ以上の文字を入力してEnterを押すと、ユーザー・ノートが作成されます。例えば、[図 141 : Dump Storage 画面の例](#) ページ 216 にある Dump Storage 画面の場合、アドレス 17C01389 ストレージにカーソルを置き、「This is」と入力すると、次のような画面が表示されます。

```

File View Services Help
-----
Dump Storage                               17C01380-17C013D7
Command ==>                               Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7      MVS2      2019/08/12 13:46:58

Address  Offset      Hex                               EBCDIC / ASCII
-----
Module IDISCBL1 program IDISCBL1 + X'640', source line # 32
Event 1 Program IDISCBL1 GPR 1 + X'1051'
Event 1 Program IDISCBL1 GPR 10 + X'1300'
Event 1 Program IDISCBL1 GPR 11 + X'1008'
Event 1 Program IDISCBL1 GPR 12 + X'1348'
Event 1 Program IDISCBL1 GPR 14 + X'680'
Event 1 Program IDISCBL1 GPR 3 + X'600'
Event 1 Program IDISCBL1 GPR 4 + X'1544'
17C01380      20004160 61345060 51This i s10C310A *...-/.&-..K....*
17C01390      +10 D2035110 310E5860 20085060 5114D20B *K.....-&-..K.*
17C013A0      +20 51183112 58602004 50605128 58602008 *...-..&-...-.*
17C013B0      +30 5060512C D2175130 311E98EC D00C07FE *&-..K.....q.}...*
Module IDISCBL1 CSECT CEESG005
17C013C0      +40 E2F0F0F5 00140001 00000000 00000000 *S005.....*
17C013D0      +50 00000000 00000000 *.....*
Module IDISCBL1 CSECT CEEBETBL
F1=Help  F3=Exit  F7=Up    F8=Down  F10=Prev F11=Next

```

Enter を押すと ISPF 編集パネルが表示され、入力したテキストで初期設定されます。

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
Volume:
EDIT      Note.17C01389                      Columns 00001 00072
Command ==>                               Scroll ==> PAGE
***** ***** Top of Data *****
000001 This is
***** ***** Bottom of Data *****

F1=Help  F2=Split  F3=Exit  F4=:tf  F5=Rfind  F6=Rchange
F7=Up    F8=Down   F9=Swap  F10=Left F11=Right F12=Cancel

```

必要な行を追加し、ノート作成が完了します。後でノート画面が縮小した場合、ノートはその行のみ表示されるため、最初の行は見出しとして使用する必要があります。

最後のノートは次のようになります。

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
Volume: _____
EDIT      Note.17C01389                      Columns 00001 00072
Command ==> ----- Scroll ==> PAGE
***** ***** Top of Data *****
000001 This is an important reminder!
000002 The contents of storage at this offset into this module could be
000003 significant for the understanding of the error that caused this fault.
***** ***** Bottom of Data *****

F1=Help      F2=Split      F3=Exit      F4=:tf      F5=Rfind      F6=Rchange
F7=Up        F8=Down       F9=Swap      F10=Left    F11=Right     F12=Cancel
    
```

PF3 を押すとストレージ域画面に戻ります。新規に作成され、属するストレージの前に挿入されたユーザー・ノートが表示されます。

```

File View Services Help
-----
Dump Storage                      17C01380-17C013D7
Command ==> ----- Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7      MVS2      2019/08/12 13:46:58

Address  Offset      Hex                                     EBCDIC / ASCII
-----
Module IDISCBL1 program IDISCBL1 + X'640', source line # 32
Event 1 Program IDISCBL1 GPR 1 + X'1051'
Event 1 Program IDISCBL1 GPR 10 + X'1300'
Event 1 Program IDISCBL1 GPR 11 + X'1008'
Event 1 Program IDISCBL1 GPR 12 + X'1348'
Event 1 Program IDISCBL1 GPR 14 + X'680'
Event 1 Program IDISCBL1 GPR 3 + X'600'
Event 1 Program IDISCBL1 GPR 4 + X'1544'
17C01380      20004160 61345060 51                *...-/.&-.*
- This is an important reminder!
  The contents of storage at this offset into this module could be
  significant for the understanding of the error that caused this fault.
17C01389      +9                08D203 510C310A *      .K.....*
17C01390      +10 D2035110 310E5860 20085060 5114D20B *K.....-..&-..K.*
17C013A0      +20 51183112 58602004 50605128 58602008 *.....-..&-.....*
17C013B0      +30 5060512C D2175130 311E98EC D00C07FE *&-..K.....q.}....*
F1=Help      F3=Exit      F7=Up        F8=Down      F10=Prev     F11=Next
    
```

デフォルトでは、すべてのユーザー・ノートは「展開」されて表示され、それはノートの前にある負符号 (-) のポイント・アンド・シュート・フィールドで示されます。このフィールドにカーソルを置いて Enter を押すと、ノートは次のように“縮小”されます。

```

File View Services Help
-----
Dump Storage                               17C01380-17C013D7
Command ==>                               Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7      MVS2      2019/08/12 13:46:58

Address  Offset      Hex                                     EBCDIC / ASCII
-----
Module IDISCBL1 program IDISCBL1 + X'640', source line # 32
Event 1 Program IDISCBL1 GPR 1 + X'1051'
Event 1 Program IDISCBL1 GPR 10 + X'1300'
Event 1 Program IDISCBL1 GPR 11 + X'1008'
Event 1 Program IDISCBL1 GPR 12 + X'1348'
Event 1 Program IDISCBL1 GPR 14 + X'680'
Event 1 Program IDISCBL1 GPR 3 + X'600'
Event 1 Program IDISCBL1 GPR 4 + X'1544'
17C01380      20004160 61345060 51                *...-/.&-.*
+ This is an important reminder!
17C01389      +9          08D203 510C310A *          .K.....*
17C01390      +10        D2035110 310E5860 20085060 5114D20B *K.....-..&-..K.*
17C013A0      +20        51183112 58602004 50605128 58602008 *.....-..&-.....*
17C013B0      +30        5060512C D2175130 311E98EC D00C07FE *&-..K.....q.}*
Module IDISCBL1 CSECT CEESG005
17C013C0      +40        E2F0F0F5 00140001 00000000 00000000 *S005.....*
F1=HeLp  F3=Exit  F7=Up    F8=Down  F10=Prev F11=Next

```

ポイント・アンド・シュート・フィールドが正符号(+)になり、「縮小」されていることを示します。このポイント・アンド・シュート・フィールドにカーソルを置くことで、縮小または展開ビューを切り替えられます。

以下の2つの追加のアクション文字(大/小文字は区別されません)を使用して、ポイント・アンド・シュート・フィールドを上書きすることも可能です。

#### D

ユーザー・ノートの削除に使用します。

#### E

ユーザー・ノートの編集に使用します。

現行の障害項目に存在するすべてのユーザー・ノートを表示するには、対話式レポートの任意の画面のコマンド行から NOTELIST コマンドを入力するか、View アクション・バーのプルダウン・メニューから「List User Notes」オプションを選択します。結果は、[図 144 : User Note List 画面の例](#) ページ 222 に示す例のように表示されます。

図 144. User Note List 画面の例

```

File View Services Help
-----
User Note List                               Line 1 Col 1 80
Command ==>                               Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7          MVS2    2019/08/12 13:46:58

- Collapse all / + Expand all
{The following line commands are available: E (Edit), D (Delete), S (Show), +
(Expand), - (Collapse). To enter a line command, overtype the +/- sign in
column 1, or simply place the cursor on the +/- sign and press Enter to
perform the default expand/collapse action indicated.}

  Address Text
-----
- 17C01389 This is an important reminder!
      The contents of storage at this offset into this module could be
      significant for the understanding of the error that caused this fa
- 17C01610 So is this!

F1=Help      F3=Exit      F5=RptFind  F6=Actions  F7=Up       F8=Down
F10=Left     F11=Right

```

この画面のオプションのヘルプ・テキストに示されるように、各ユーザー・ノートの前に示されるポイント・アンド・シュート・フィールドは、ストレージ域画面と同様に特定のアクションを要求するために上書き入力できます。

また、User Note List 画面では、最上部にある「Expand all」または「Collapse all」ポイント・アンド・シュート・フィールドを選択することで、同期的に展開または縮小できます。ノートの展開または縮小という状態は、User Note List 画面と Dump Storage 画面で共通です。そのため、一方の画面に加えられた変更は、もう一方の画面に反映されます。

ユーザー・ノートに関連するストレージを表示するには、S 行コマンドを使用するか、アドレスのポイント・アンド・シュート・フィールドにカーソルを置き、Enter キーを押します。

ユーザーが対話式レポートを終了する際、ユーザー・ノートはヒストリー・ファイルの障害項目に保管されます。このとき、ユーザー・ノートが追加または変更されていた場合、[図 145 : User Notes Update プロンプトの例 ページ 223](#) の例に示す画面のように、障害項目の更新の確認プロンプトが出されます。

図 145. User Notes Update プロンプトの例

```

File View Services Help
User Notes Update
User notes have been added or modified for the current fault entry. Press
Enter to update the fault entry with the current user notes, or press
PF3/PF12 to exit from the interactive report without updating the fault
entry.
History file DSN . . : 'IDI.HIST'
Fault ID . . . . . : F00264
F1=Help      F3=Exit      F12=Cancel

4. Storage Areas
5. Messages
6. Language Environment Heap Analysis
7. Abend Job Information
8. Fault Analyzer Options

{Fault Analyzer maximum storage allocated: 2.61 megabytes.}

*** Bottom of data.
F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

この画面で Enter を押すと、履歴ファイルの障害項目をリフレッシュします。PF3 または PF12 を押すと、ユーザー・ノートは更新せずに対話式レポートを終了します。

## CICS トランジション・ストレージの表示 (CICSSTG)

CICS Information 画面 (基本オプション: CICS 情報 ページ 182 を参照) から「CICS Transaction Storage Summary」または「CICS Transaction Storage」オプションを選択するための代替方法として、CICSSTG コマンド (CICSSTG ページ 105 を参照) を使用できます。

図 146: CICS Transaction Storage 画面の例 ページ 223 には、CICSSTG コマンドの CICS Transaction Storage 画面の例が示されています。

図 146. CICS Transaction Storage 画面の例

```

File View Services Help
CICS Transaction Storage
Command ==>
Previous Next Filter
Element: 1 of 3 (9), address 0_1AE00000, length 0x870 (USER31)
Filter: Storage classes(USER31)
Line 1 Col 1 80
Scroll ==> CSR

Address  Offset      Hex
1AE00000      E4F0F0F0 F0F1F0F8 00E86EC4 C6C8C5C9 *U0000108.Y>DFHEI*
1AE00010      +10 E4E24040 40404040 00000000 00000000 *US .....
1AE00020      +20 00000000 00000000 00000000 00000000 *.....
1AE00030      +30 00000000 00000000 00000000 1AE00100 *.....\..*
1AE00040      +40 00000000 1AE008A0 00000000 00000000 *.....\.....*
1AE00050      +50 00000000 0008B614 00000000 19C7D429 *.....GM.*
1AE00060      +60 00000000 00000420 00000000 1AE0003C *.....\..*
1AE00070      +70 00000000 00041800 00000000 1941DA98 *.....q*
1AE00080      +80 00000000 1AC96DF0 00000000 1AC9725C *.....I_0.....I.**
1AE00090      +90 00000000 19419C02 00000000 1AC97438 *.....I..*
1AE000A0      +A0 00000000 7F466948 00000000 1AC93000 *.....".....I..*
1AE000B0      +B0 00000000 1AE00008 00000000 1AFFB4A4 *.....\.....u*
1AE000C0      +C0 00000000 00882000 00000000 00000000 *.....h.....*

```

「前へ」および「次へ」ポイント・アンド・シュート・フィールドを使用して、個別のストレージ域をナビゲートします。  
「フィルター」ポイント・アンド・シュート・フィールドを使用して、CICSストレージ・クラス、ストレージ域の長さ、特定のテキストを含む領域に基づき、含まれているストレージ域をフィルターします。CICSSTG画面からPF1を使用して、追加のヘルプを表示できます。

## アドレス・スペース・ストレージ・マップの表示 (STGMAP)

現在のアドレス・スペースのレイアウトに関する情報を表示するには、以下のいずれかの方法で、Storage Map画面を呼び出します。

- 対話式再分析レポート内から STGMAP コマンドを発行する。
- アクション・バーから、**Services > Storage Map** を選択します。

画面には、以下の2種類があります。Summary と Details。

最初に、Summary画面が表示されます。Summary画面には、アドレス・スペースのメインのストレージ域と、ユーザー領域のサイズおよび残りのフリー・スペースに関する情報が表示されます。

図147に、Storage Map Summary画面の例を示します。



図 147. Storage Map Summary 画面の例

```

File  View  Services  Help
Storage Map                               Line 1 Col 1 80
Command ==>                               Scroll ==> CSR
Summary Details - JOBNAME: S0C7A          SYSTEM ABEND: 0C7          FAE1          202

High Private < 100000000_00000000
Shared Area <---- 20000_00000000
High Common <----- 200_00000000
Low Private <----- 1EF_80000000
Compressed references <----- 10_00000000
area
=====
Extended Private <----- 80000000 "The Bar"
(Free 31-Bit User Region <----- 18B00F7E Current Address
= 1006.70 MB)
Extended Common <----- 18B00000
Common <----- 01000000 "The Line"
Private <----- 00900000
(Free 24-Bit User Region <----- 00002000
= 436.00 KB)
Common <----- 00000000

USER REGION SIZES
Requested
REGIONX(400K,1G)
Actual
64-Bit. . . . . : X'FFFFFFFF00000000' (16.00 EB)
31-Bit. . . . . : X'40000000' (1.00 GB)
24-Bit. . . . . : X'00074000' (464.00 KB)

*** Bottom of data.

```

Details 画面は、Storage Map 画面の上部にあるポイント・アンド・シュート・フィールドから選択できます。Details 画面には、メインのストレージ域と各領域のサイズがさらに詳細に表示されます。

Dump Storage 画面からストレージ・マップを表示すると、Summary 画面と Details 画面の両方に現在のアドレスが示されます。

ストレージ域の名前は、Summary 画面と Details 画面の両方にポイント・アンド・シュート・フィールドとして表示されます。カーソルをストレージ域名に置き、Enter キーを押してストレージ域を選択し、そのダンプ・ストレージを表示します。詳しくは、[データ域の表示 ページ 217](#) を参照してください。

## PSW 情報の表示

下図は、対話式再分析レポートでの PSW の例を示しています。

図 148. プログラム状況ワード画面の例

```

File View Services Help
-----
Event 1 of 3: Abend S0C7 *** Point of Failure ***                               Line 69 Col 1 80
Command ==> _____ Scroll ==> CSR
JOBNAME: IDIVPCOB  SYSTEM ABEND: 0C7 -----FAE1 2019/05/31 12:04:53
  Second Operand Storage. . : C1C2C3CF      *ABC.*

Associated Messages

CEE3207S The system detected a data exception (System Completion Code=0C7).
                                     ① ②
Program Status Word (PSW) . . : 078D2000 98601172
PSW Summary . . . . . : Primary Space Mode, PSW Key 8, AMODE 31, Problem
                        State

General Purpose Registers (AMODE: 64 31 24 , Bytes: Dec Hex ):
R0: 10000_186970F0 (405,264 bytes of storage addressable)
R1: 0_18600FA1 (Module IDISCBL1 program IDISCBL1 + X'209')
R2: 0_000077FC (83,972 bytes of storage addressable)
R3: 0_18601136 (Module IDISCBL1 program IDISCBL1 + X'39E', source
              line # 26 )
R4: 0_18600DD0 (Module IDISCBL1 program IDISCBL1 + X'38')
R5: 0_1860EE90 (962,928 bytes of storage addressable)
R6: 0_00000000 (2,048 bytes of storage addressable)

```

PSWには2つのポイント・アンド・シュート・フィールドが含まれています。1つのフィールドは高位ワード ① 用であり、もう1つのフィールドは低位ワード ② 用です。

低位ワードのポイント・アンド・シュート・フィールドを選択すると、そのアドレスのストレージが表示されます。詳しくは、[保管場所の表示 ページ 215](#)を参照してください。

高位ワードのポイント・アンド・シュート・フィールドを選択すると、Program Status Word Breakdown Table 画面が表示されます。例として下図を参照してください。

図 149. Program Status Word Breakdown Table 画面の例

```

Program Status Word Breakdown Table
Command ==> ----- Line 1 Col 1 76
                               Scroll ==> CSR
PSW: 078D1000 A68011AA (ESA/390)
PSW Format:


|   |   |     |   |   |   |         |   |    |    |    |    |    |          |          |
|---|---|-----|---|---|---|---------|---|----|----|----|----|----|----------|----------|
| 0 | R | 000 | T | I | E | PSW KEY | 1 | M  | W  | P  | AS | CC | PGM MASK | 00000000 |
| 0 |   | 5   |   | 8 |   | 12      |   | 16 | 18 | 20 |    | 24 |          | 31       |



|    |                     |  |  |  |  |  |  |  |  |  |  |  |    |
|----|---------------------|--|--|--|--|--|--|--|--|--|--|--|----|
| A  | Instruction Address |  |  |  |  |  |  |  |  |  |  |  |    |
| 32 | 33                  |  |  |  |  |  |  |  |  |  |  |  | 63 |


Actual Values


|   |   |     |   |   |   |      |   |    |    |    |    |    |      |          |
|---|---|-----|---|---|---|------|---|----|----|----|----|----|------|----------|
| 0 | 0 | 000 | 1 | 1 | 1 | 1000 | 1 | 1  | 0  | 1  | 00 | 10 | 0000 | 00000000 |
| 0 |   | 5   |   | 8 |   | 12   |   | 16 | 18 | 20 |    | 24 |      | 31       |



|    |          |  |  |  |  |  |  |  |  |  |  |  |    |
|----|----------|--|--|--|--|--|--|--|--|--|--|--|----|
| 1  | 18601172 |  |  |  |  |  |  |  |  |  |  |  |    |
| 32 | 33       |  |  |  |  |  |  |  |  |  |  |  | 63 |



| Bit   | Meaning                      |               |
|-------|------------------------------|---------------|
| 1     | Program-Event-Recording Mask | (R)           |
| 5     | DAT Mode                     | (T = 1)       |
| 6     | Input/Output Mask            | (I)           |
| 7     | External Mask                | (E)           |
| 12    | One indicates ESA/390        |               |
| 13    | Machine-Check Mask           | (M)           |
| 14    | Wait State                   | (W = 1)       |
| 15    | Problem State                | (P = 1)       |
| 16-17 | Address Space Control        | (AS)          |
|       | xx Real Mode                 | (T = 0)       |
|       | 00 Primary-Space Mode        | (T = 1)       |
|       | 01 Access-Register Mode      | (T = 1)       |
|       | 10 Secondary-Space Mode      | (T = 1)       |
|       | 11 Home-Space Mode           | (T = 1)       |
| 18-19 | Condition Code               | (CC)          |
| 20    | Fixed-Point-Overflow Mask    |               |
| 21    | Decimal-Overflow Mask        |               |
| 22    | HFP-Exponent-Overflow Mask   |               |
| 23    | HFP-Significance Mask        |               |
| 32    | 31-Bit Addressing Mode       | (A = 1)       |
| 33-63 | Instruction Address          | (Hexadecimal) |


*** Bottom of data.

```

この画面には、選択した PSW 内にある各種ビット・フィールドの明細が示されます。

この画面には、対をなす 2 つの表があります。最初の表には汎用形式の PSW が示され、2 番目の表には選択した PSW の内容が示されます。画面の残りの部分には、フォーマット・テーブルと関連ビット・オフセットで参照されているビット・フィールドが含まれています。

## DSECT 情報を使用したストレージ域のマッピング

対話式レポートにおいて DSECT コマンドを使用すると、アセンブラー・マクロまたは DSECT サンプル集を含む PDS または PDSE データ・セット・メンバーに基づき、ストレージ域をマップできます。

DSECT コマンド (構文については [DSECT ページ 107](#) を参照) は、任意の画面のコマンド行または PF キー割り当てによって入力できます。デフォルトでは、DSECT コマンドは PF4 に割り当てられています。

呼び出すと、次のようなポップアップ・ウィンドウが表示されます。

図 150. Storage DSECT Mapping Entry 画面の例

```

File  View  Services  Help
----- Storage DSECT Mapping -----
Enter the name of the Dsect in the Dsect Name field to be used to map the
storage address provided in the Address field. Press PF4 to display a list
of all available Dsects.  Optionally a specific Dsect can be used by
supplying a Dataset and Member name in the DSN field.  In this case if a
Dsect name is not provided it will be made equal to the member name.

Address _____
Dsect Name _____
DSN . . . _____

F1=Help      F3=Exit      F12=Cancel

First Operand Address . . : 0002A120 (3808 bytes of storage addressable)
First Operand Length. . . : 8
First Operand Storage . . : 00000000 0986888C *.....fh.*
Second Operand Address. . : 0002A110 (3824 bytes of storage addressable)
Second Operand Length . . : 4
Second Operand Storage. . : C1C2C3CF *ABC.*
F1=Help      F3=Exit      F5=RptFind  F6=Actions  F7=Up      F8=Down
F10=Left     F11=Right

```



**注:** ポイント・アンド・シュート・フィールドのアドレス上にカーソルがあるときに DSECT コマンドを発行した場合、ポップアップ画面のアドレスは、そのアドレスで自動的に初期化されます。

開始アドレス (まだ入力済みでない場合)、およびストレージ域のマッピング時に使用する DSECT の名前を入力できます。Address フィールドは 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。

DSECT の名前は次のいずれかの方法で提供できます。

1. DSECT 名のみを入力します。この場合、一致する IDIDSECT 連結が検索されます (詳細については、[IDIDSECT 連結 ページ 230](#)を参照)。要求された DSECT が IDIDSECT 連結に複数ある場合、PF4 を押し、選択可能なすべての DSECT のリストから適切なものを選択します。「S」を使用すると DSECT をリストから選択でき、「E」を使用すると編集できます。
2. 指定された DSECT が保管されたデータ・セットおよびメンバー名を入力します。DSECT 名が指定されない場合、PDS または PDSE メンバー名と同様の名前がデフォルトとして使用されます

有効なストレージ・アドレスおよび DSECT 名が指定されると、次の例のようにストレージはマップされて表示されます。

図 151. Storage DSECT Mapping Map 画面の例

```

File View Services Help
-----
Dsect mapping for DFHCSADS at address 4de20                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

CICS DUMP: SYSTEM=QXPM2C61 CODE=ASRA      ID=      MVS2      2019/06/25 13:47:55

0004DE20 +0000                                DSECT DFHCSADS
                                                DFHCSABA EQU   *

0004DE20 +0000 00000248 0000D0A0 17EB4D00 983C1ECE
                80BF4DA8 80800000 18685160 18642330
                000003FD 18973FB8 00000BAF 00000000
                983C1A40 18973000 18685160 18684B70
                00051D80 17F90680
                                                CSAOSRSA DS    18F
                                                CSASOSI DS    0B
                                                CSASSI1 DS     B
                                                CSAFPURG EQU  X'80'
                                                CSAFTCAB EQU  X'40'
                                                CSASDTRN EQU  X'20'
                                                CSACSDOP EQU  X'02'
                                                CSASOSON EQU  X'01'
                                                CSAKCM1 DS    0B
                                                CSASSI2 DS     B
0004DE68 +0072 00
                                                F5=RptFind  F6=Actions  F7=Up      F8=Down

0004DE69 +0073 10
F1=Help      F3=Exit      F5=RptFind  F6=Actions  F7=Up      F8=Down
F10=Left     F11=Right

```

さらに DSECT 情報を表示する必要がある場合は、上下左右にスクロールします。

PF3 を押すと、Storage DSECT Mapping Map 画面から戻ります。

## IDISYSLB 連結

DSECT コマンドと IDIPDSCU ユーティリティはどちらも内部的に ADATA を生成するアセンブラーを使用します。ADATA はフィールド名とオフセットを決定するために使用されます。アセンブラーが使用する syslib DD は、ソース DSECT と同じデータ・セットに動的に割り振られます。ただし、この割り振りは、アセンブラー・マクロを使用して DSECT を生成し、そのマクロが DSECT データ・セットに存在しない場合、不十分に実行される可能性があります。このような状況に対応するために、追加の syslib データ・セットを IDISYSLB DD に割り振ることで指定できます。

例えば、データ・セット MY.DSECTS には、CICS DFHSIT DSECT を生成するための 1 つのメンバー CICSSIT が含まれません。

```

000001 * GENERATE THE CICS DFHSIT DSECT
000002         DFHSIT TYPE=DSECT
000003         END

```

DSECT コマンドまたは IDIPDSCU ユーティリティでこのデータ・セットまたはメンバーを使用しても、DFHSIT マクロが解決されないため、使用可能な DSECT は生成されません。このため、IDISYSLB DD を使用できます。例:

```

//UTIL1 EXEC PGM=IDIPDSCU,PARM=('MY.DSECTS')
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//IDISYSLB DD DISP=SHR,DSN=CICS.TS61.CICS.SDFHMAC

```

対話式再分析中に DSECT コマンドを使用する場合、または ISPF で IDIPDSCU を使用する場合は、次の TSO ALLOCATE コマンドを使用します。

```
ALLOCATE FILE(IDISYSLB) DS('CICS.TS61.CICS.SDFHMAC') SHR
```

## IDIDSECT 連結

IDIDSECT 連結は、オプション・データ・セットの DATASETS セクションにオプションで指定できます。指定した場合、DSECT コマンドの処理時に使用される DSECT ファイルを含む 1 つ以上の PDS または PDSE データ・セット名を指定する必要があります。

IDIDSECT データ・セット属性は以下のいずれかでなければなりません。

- RECFM=FB および LRECL≥80
- RECFM=VB および LRECL≥84

LRECL に関係なく、各レコードの最初の 80 バイトのみが読み取られます。推奨 IDIDSECT データ・セット属性は RECFM=FB および LRECL=80 です。

対話式レポートでは、DSECT コマンドが最初に発行されたとき、IDIDSECT 連結の各データ・セットを処理します。データ・セットに \$DINDEX メンバーが含まれる場合 ([DSECT データ・セットの索引作成 \(\\$DINDEX メンバー\) ページ 230](#)を参照)、このメンバーの DSECT の詳細が使用されます。含まれない場合、データ・セットの各メンバーは、同様の名前の DSECT を含むと考えられます。すべての DSECT の詳細がデータ・セットに判別されると、プロセスは、IDIDSECT 連結のすべてのデータ・セットが処理されるまで次のデータ・セットに繰り返されます。

このプロセスは、対話式レポート・セッションごとに一度だけ行われます。新しい DSECT が IDIDSECT 連結内のデータ・セットに追加された場合、または \$DINDEX メンバーが更新された場合は、以下のいずれかのアクションを行う必要があります。

- 対話式レポートを再開する。
- 含まれるデータ・セットおよびメンバー名を指定して新しい DSECT を明示的に識別する。

## DSECT データ・セットの索引作成 (\$DINDEX メンバー)

DSECT 名を最大 63 文字まで許可するには、また複数の DSECT を含む各メンバーを許可するには、IDIPDSCU ユーティリティを使用して \$DINDEX メンバーを作成します。

\$DINDEX メンバーは、PDS または PDSE の各メンバー内の、DSECT ごとに 1 行必要です。各行は、DSECT 名、次にスペース、さらに DSECT があるメンバーから構成される必要があります。例:

```
DSECT1 MEMBER1
DSECT2 MEMBER1
LONGDSECTNAME1 MEMBER2
LONGDSECTNAME2 MEMBER2
```

この例では、メンバー MEMBER1 が DSECT の DSECT1 および DSECT2 を含み、MEMBER2 が DSECT の LONGDSECTNAME1 および LONGDSECTNAME2 を含みます。

## DSECT 索引作成のユーティリティ (IDIPDSCU)

IDIPDSCU ユーティリティは、提供データ・セットの \$DINDEX メンバーの作成に使用されます。データ・セットの各メンバーに対してアセンブラを呼び出し、アセンブラ出力から埋め込まれた DSECT を抽出することでこれを実行します。

DSECT またはマクロ展開が特別なキーワードの指定を要求する場合、別のメンバーのコード化が必要な場合があります。このメンバーは、要求されたキーワードを提供して該当のマクロを呼び出す必要があり、呼び出すマクロと同じデータ・セットに保管する必要があります。例えば、CICS® はその SDFHMAC データ・セットに DFHTCTZE というメンバーを提供します。これは、複数のターミナル関連 DSECT を提供します。このメンバーが IDIPDSCU ユーティリティによって直接処理される場合、検出で特殊なマクロ・キーワードの指定が必要なため、TCTENIB DSECT は検出されません。メンバーが、次のソース行を含む SDFHMAC データ・セットに作成 (またはコピー) される場合、TCTENIB を含むすべての DSECT が検出されます。

```
DFHTCTZE CICSYST=YES
```

IDIPDSCU ユーティリティは、ISPF のデータ・セット名の横に IDIPDSCU を入力することで、またはバッチ・ユーティリティとして使用できます。この場合、処理されるデータ・セットはパラメーターとして渡されます。以下の例を参照してください。

```
//UTILJOB1 JOB ...
//RUNUTIL EXEC PGM=IDIPDSCU,PARM=('fully_qualified_PDS(E)_data_set_name')
//SYSPRINT DD SYSOUT=*
```

IDIPDSCU ユーティリティは、ターゲット・データ・セットに \$DINDEX メンバーを作成するため、このデータ・セットへの書き込みアクセス権が必要です。

マクロ解像度の SYSLIB データ・セットの指定については、[IDISYSLB 連結 ページ 229](#)を参照してください。

## チェーン・データ域の表示

RUNCHAIN コマンド ([RUNCHAIN ページ 115](#) を参照) を使用すると、ストレージにおいてリンクされた制御ブロックのチェーンをスキャンできます。RUNCHAIN コマンドは、任意の対話式レポートのコマンド行から RUNCHAIN と入力するか、RUNCHAIN を PF キーに割り当てることで呼び出せます。呼び出すと、次のようなポップアップ・パネルが表示されます。

図 152. Storage RUNCHAIN Command 入力画面の例

```

File View Services Help
Storage RUNCHAIN Command
Enter the required fields and press Enter .
Start Address . . . . .
Max Number Control Blocks : 9999 (Decimal)
Forward Pointer Offset . . . (Hex)
End of Chain Identifier . . . (Hex, Default Values: All
                                0's and all F's)
Eyecatcher Text . . . . .
Eyecatcher Offset . . . . . (Hex)

F1=Help      F3=Exit      F12=Cancel

000011      n/a          01 PARM1          PIC X(4) .
000012      n/a          01 PARM2.

Data Field Values:
PARM1      = 0001
PARM2      = HEADING FOR IDIXSNAP
F1=Help    F3=Exit      F5=RptFind    F6=Actions    F7=Up        F8=Down
F10=Left   F11=Right

```

Start Address は 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。

31 ビットの Start Address を指定した場合は、以下のようになります。

- Forward Pointer Offset の順方向ポインターは 31 ビット・アドレスであると想定されます。
- End of Chain Identifier は 32 ビットの 16 進値であると想定されます。8 桁未満の 16 進数字を指定した場合、32 ビットの 16 進値を形成するために、指定した値には先行ゼロが埋め込まれます。

64 ビットの Start Address を指定した場合は、以下のようになります。

- Forward Pointer Offset の順方向ポインターは 64 ビット・アドレスであると想定されます。
- End of Chain Identifier は 64 ビットの 16 進値であると想定されます。16 桁未満の 16 進数字を指定した場合、64 ビットの 16 進値を形成するために、指定した値には先行ゼロが埋め込まれます。

Start Address および Forward Pointer Offset で、RUNCHAIN コマンドは、次の終了条件のいずれかが満たされるまで制御ブロックのチェーンに従います。

1. スキャンされた制御ブロック数が、ユーザーによって設定された最大数を超えた場合 (デフォルト値は 9999)。
2. 現行の制御ブロックの順方向ポインターが、「チェーン終了」の値を含む場合。この値は次のようになります。
  - 00000000 (31 ビット Start Address の場合) または 0000000000000000 (64 ビット Start Address の場合)。
  - FFFFFFFF (31 ビット Start Address の場合) または FFFFFFFFFFFFFFFF (64 ビット Start Address の場合)。
  - 初期開始アドレス。チェーンがループ済みと暗黙指定する
  - ユーザー指定の End Of Chain Identifier。



- 指定された値には、8桁の長さまで (31 ビット Start Address の場合)、または 16桁の長さまで (64 ビット Start Address の場合)、自動的に先行ゼロが埋め込まれます。64 ビット・アドレスには下線が含まれる場合があります (詳しくは上記の Start Address を参照)。
- 31 ビット Start Address に指定された値が 8桁を超えた場合、値は 8桁まで左側が切り捨てられます。
- 64 ビット Start Address に指定された値が 16桁を超え、アドレスに下線が含まれていない場合、値は 16桁まで左側が切り捨てられます。

3. 現行の制御ブロックの順方向ポインターが、無効または使用不可のストレージをポイントした場合。

制御ブロックごとに、そのアドレスと初めの 32 バイトまでのデータが表示されます。

オプションで、制御ブロックに目印およびオフセットを設定できます。この場合、各制御ブロックでは、指定されたオフセットのテキストと指定されたテキストが比較され、一致しない場合、警告メッセージが発行されます。

RUNCHAIN コマンドの例のように、Storage RUNCHAIN Command 入力画面で次のように指定できます。

```

File View Services Help
Storage RUNCHAIN Command

Enter the required fields and press Enter .

Start Address . . . . . 0005C000
Max Number Control Blocks . 9999 (Decimal)
Forward Pointer Offset . . . 1C (Hex)
End of Chain Identifier . . . (Hex, Default Values: All
                                0's and all F's)
Eyecatcher Text . . . . . DFHSMQPH
Eyecatcher Offset . . . . . 2 (Hex)

F1=Help    F3=Exit    F12=Cancel

000011    n/a        01  PARM1        PIC X(4) .
000012    n/a        01  PARM2.

Data Field Values:
PARM1     = 0001
PARM2     = HEADING FOR IDIXSNAP
F1=Help   F3=Exit    F5=RptFind  F6=Actions  F7=Up       F8=Down
F10=Left  F11=Right

```

Enter を押すと、次の画面が表示されます。

```

File View Services Help
-----
Runchain starting at address 0005C000. 1 Control blocks          Line 1 Col 1 80
Command ==>                                                    Scroll ==> CSR
CICS DUMP: SYSTEM=QXPM2C61 CODE=ASRA      ID=      MVS2      2019/06/25 13:47:55

Address  Hex (First 32 Bytes Of Data)
0005C000 00306EC4 C6C8E2D4 D8D7C840 40404040 C1D76DE3 C3C1F2F4 002AA000 0005
0005B000 00306EC4 C6C8E2D4 D8D7C840 40404040 C1D76DE3 C3C1F2F4 0005C000 0005
0005A000 00306EC4 C6C8E2D4 D8D7C840 40404040 C1D76DE3 C3C1F2F4 0005B000 0005
00059000 00306EC4 C6C8E2D4 D8D7C840 40404040 C1D76DE3 C3C1F2F4 0005A000 2E80
2E80CCCC Invalid eyecatcher.  Expected: >DFHSMQPH
                          Found   : 24.....Q.
2E80CCCC C3C1F2F4 2E80CC14 1ED82728 C0010100 00000000 00000600 00059000 002A
002AA000 00306EC4 C6C8E2D4 D8D7C840 40404040 C1D76DE3 C3C1F2F4 2E80CCCC 0005
End of RUNCHAIN 6 Control Blocks processed

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

RUNCHAIN コマンドを終了するには、EXIT (PF3) と入力します。

## オブジェクト・コードの逆アセンブル

DISASM コマンド ([DISASM ページ 107](#)を参照) を使用すると、アドレスを指定してオブジェクト・コードを逆アセンブルできます。DISASM コマンドは、任意の対話式レポートのコマンド行から DISASM と入力するか、DISASM を PF キーに割り当てることで呼び出せます。呼び出すと、次のようなポップアップ・パネルが表示されます。

図 153. Storage Disassemble 画面の例

```

File View Services Help
-----
Storage Disassemble
-----
WARNING Before using this function you must be aware of and respect the
intellectual property rights of others. You are not authorized to use this
function to disassemble, copy or create assembly listings or disassembled
Assembler Language source code in violation of any contractual or other
legal obligation. You are authorized to use this function only for object
code for which you have verified you have the right to perform
disassembly.

Start Address . . . . . : : : _____
Origin Address (optional) : : : _____

F1=Help      F3=Exit      F12=Cancel

{Fault Analyzer maximum storage allocated: 1.71 megabytes.}

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

DISASM コマンドは、指定された開始アドレスからのコードの逆アセンブルを試行します。オプションで、起点アドレスを指定できます。この場合、逆アセンブルされた指示それぞれのオフセットが、開始アドレスではなく起点アドレスに対して計算されます。起点アドレスが指定されない場合、開始アドレスと同様のアドレスがデフォルトとして使用されます。

DISASM コマンドの例のように、Storage Disassemble 画面は次のように指定できます。

```

File View Services Help
----- Storage Disassemble -----

WARNING Before using this function you must be aware of and respect the
intellectual property rights of others. You are not authorized to use this
function to disassemble, copy or create assembly listings or disassembled
Assembler Language source code in violation of any contractual or other
legal obligation. You are authorized to use this function only for object
code for which you have verified you have the right to perform
disassembly.

Start Address . . . . . 18D00326
Origin Address (optional) . . . 18D00300

F1=Help    F3=Exit    F12=Cancel

{Fault Analyzer maximum storage allocated: 1.71 megabytes.}

*** Bottom of data.

F1=Help    F3=Exit    F5=RptFind  F6=Actions  F7=Up      F8=Down
F10=Left   F11=Right

```

Enter を押すと、次の画面が表示されます。

```

File View Services Help
-----
Disassemble command
Command ==> ----- Line 1 Col 1 80
JOBNAME: JA84Q13A SYSTEM ABEND: 0C7 SYS01 2019/12/01 13:59:00
Scroll ==> CSR

Address Offset Hex Instruction
18D00326 +26 05EF BALR R14,R15
18D00328 +28 50B0 D0C8 ST R11,200(,R13)
18D0032C +2C 41A0 D0D4 LA R10,212(,R13)
18D00330 +30 D20F 4010 306C MVC 16(16,R4),108(R3)
18D00336 +36 D216 4020 3557 MVC 32(23,R4),1367(R3)
18D0033C +3C 58E0 303C L R14,60(,R3)
18D00340 +40 50E0 4038 ST R14,56(,R4)
18D00344 +44 5890 4038 L R9,56(,R4)
18D00348 +48 4090 403E STH R9,62(,R4)
18D0034C +4C 4190 4020 LA R9,32(,R4)
18D00350 +50 5090 4010 ST R9,16(,R4)
18D00354 +54 4170 D128 LA R7,296(,R13)
18D00358 +58 5070 4018 ST R7,24(,R4)
18D0035C +5C 4190 403E LA R9,62(,R4)
F1=Help    F3=Exit    F4=Dsect   F5=RptFind  F6=Actions  F7=Up
F8=Down    F10=Left   F11=Right  F12=retrieve

```

一度逆アセンブルされた指示を示すパネルが表示されると (上記の例を参照)、PF7 および PF8 を押して順方向および逆方向にスクロールできます。

FIND/RPTFIND は、現在表示されているデータのみを検索します。

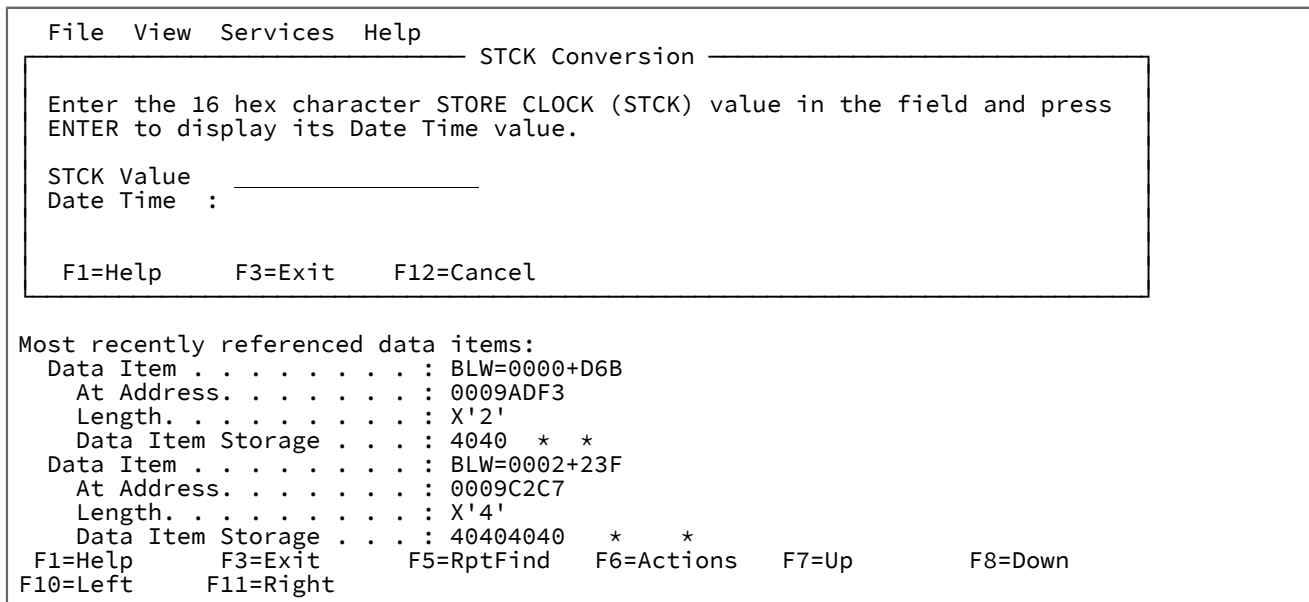
UP MAX/DOWN MAX のスクロールはこの画面ではサポートされていません。

DISASM コマンドを終了するには、EXIT と入力します (PF3)。

## STORE CLOCK 値の変換

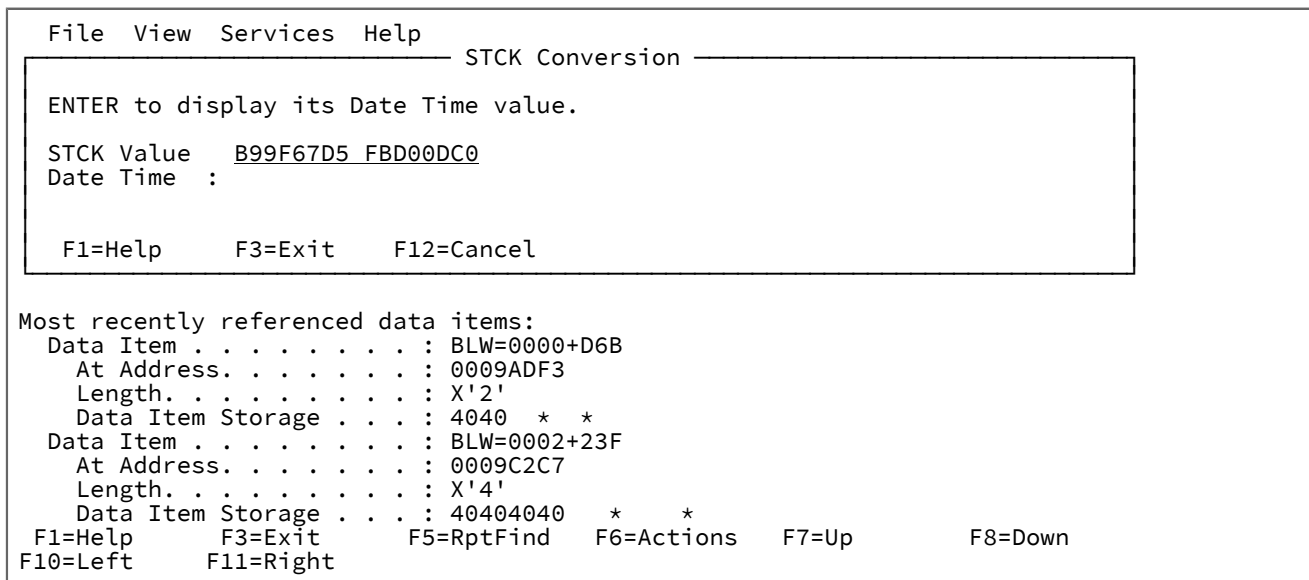
STCK コマンド (STCK ページ 117を参照) を使用すると、バイナリーの STORE CLOCK 値を目視で確認できる日付と時刻のフォーマットに変換できます。STCK コマンドは、任意の対話式レポートのコマンド行から STCK と入力するか、STCK を PF キーに割り当てることで呼び出せます。呼び出すと、次のようなポップアップ・パネルが表示されます。

図 154. STCK Conversion 画面の例



STCK 値は 16 進文字で入力する必要があります。組み込みブランクは無視されます。

STCK コマンドの例のように、STCK Conversion 画面は次のように指定できます。



Enter を押すと、次のように画面が更新されます。

```

File  View  Services  Help
----- STCK Conversion -----

Enter the 16 hex character STORE CLOCK (STCK) value in the field and press
ENTER to display its Date Time value.

STCK Value   D6C0AE3BC09E3146
Date Time   : 2019/09/20 11:45:23.462627

F1=Help      F3=Exit      F12=Cancel

Most recently referenced data items:
Data Item . . . . . : BLW=0000+D6B
At Address. . . . . : 0009ADF3
Length. . . . . : X'2'
Data Item Storage . . . . . : 4040 * *
Data Item . . . . . : BLW=0002+23F
At Address. . . . . : 0009C2C7
Length. . . . . : X'4'
Data Item Storage . . . . . : 40404040 * *
F1=Help      F3=Exit      F5=RptFind  F6=Actions  F7=Up      F8=Down
F10=Left     F11=Right

```

STCK コマンドを終了するには、EXIT と入力します (PF3)。

## ユーザー固有レポートのフォーマット

EXEC コマンドを使用して ([EXEC ページ 108](#)を参照)、REXX フォーマット・ユーザー出口を実行できます。このタイプの出口では、分析されたアプリケーション環境に固有のデータ域のフォーマットなど、ユーザー固有情報の画面を生成できます。このタイプの出口に関する概説については、[フォーマット・ユーザー出口 ページ 491](#)を参照してください。

非 REXX フォーマット・ユーザー出口は、EXEC コマンドから実行できません。

EXEC コマンドで出口名を指定すると、その出口が実行され、出口に指定されたすべての情報を含む画面が表示されます。一方、出口名を指定しない場合は、データ・セットの IDIEXEC 連結から使用可能なすべての REXX フォーマット・ユーザー出口のリストが、次の Formatting User Exit Selection List 画面の例のように表示されます。

図 155. Formatting User Exit Selection List 画面の例

```

File View Services Help
-----
Formatting User Exit Selection List                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR
JOBNAME: IDIVPPLI SYSTEM ABEND: 0C9 FAE1 2019/05/07 11:29:07

{The following line commands are available: S (Select), B (Browse), E (Edit).}

  Name      Comment/Arguments
  ----      -
  IDISUFM1  Sample Formatting user exit to display TCB information
  IDISUFM2  Sample Formatting user exit for CICS CWA
  IDISUFM3  Sample Formatting user exit to illustrate the use of formatting tags

F1=Help      F3=Exit      F4=Dsect      F5=RptFind    F6=Actions    F7=Up
F8=Down      F10=Left     F11=Right     F12=retrieve

```

Formatting User Exit Selection List 画面用のフォーマット・ユーザー出口として認識させるには、各 IDIEXEC データ・セットの制御メンバーに出口名を指定する必要があります。制御メンバーの名前には \$\$UFMTX を指定してください。制御メンバーの各レコードには、出口のメンバー名 (大/小文字の区別はありません) を指定できます。また、オプションでコメントを含めることができ、これらは Formatting User Exit Selection List 画面に表示されます。制御メンバーの例を、以下に示します。

図 156. \$\$UFMTX メンバーの例

```

BROWSE      FRED.EXEC($$UFMTX) - 01.01                          Line 00000000 Col 001 080
Command ==> _____ Scroll ==> CSR
***** Top of Data *****
IDISUFM1 Sample Formatting user exit to display TCB information
IDISUFM2 Sample Formatting user exit for CICS CWA
IDISUFM3 Sample Formatting user exit to illustrate the use of formatting tags
***** Bottom of Data *****

F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP nex  F10=LEFT     F11=RIGHT     F12=RETRIEVE

```

上記のコントロール・メンバーの例は、データ・セット IDI.SIDISAM1 内のメンバー IDISUFMX として、ソフトコピー・フォーマットで提供されています。

このサンプル制御メンバーを使用するには、以下を行います。

1. この例を他のデータ・セットにコピーし、\$SUFMTX に名前変更します。
2. このメンバー内で指定されている 3 つのサンプル・フォーマット・ユーザー出口を、同じデータ・セットにコピーします。
3. 現在、制御メンバーおよび出口を含んでいるデータ・セット名を DATASETS(IDIEXEC(*data-set-name*)) オプションに指定します。
4. Fault Analyzer ISPF インターフェースを呼び出し、障害項目に対して対話式再分析を行います。
5. 出口名を指定せずに EXEC コマンドを発行します。このアクションにより、[図 155 : Formatting User Exit Selection List 画面の例 ページ 238](#)のような画面が表示され、この画面からサンプル出口を実行できます。

Formatting User Exit Selection List 画面から、各出口に対して行コマンドを発行できます。

#### S

出口を実行する。

#### B

出口に対して ISPF ブラウズを入力する。

#### E

出口に対して ISPF エディットを入力する。

出口では、しばしば 1 つ以上のパラメーターの受け渡しが必要になります。これらのパラメーターの引き渡しは、Formatting User Exit Selection List 画面で出口名の右にある「Comments/Arguments」フィールドを (必要であれば) クリアし、上書き入力することによって行うことができます。上書きされると、フィールドの色が変わり、出口のパラメーターとしてデータが使用されることを示します。フィールドをクリアして Enter を押すと、代わりにオリジナルのコメントが再表示されます。

## コンパイラー・リストまたはサイド・ファイルのプロンプト

COBOL、PL/I、C/C++、またはアセンブラー・プログラムに適合したコンパイラー・リストまたはサイド・ファイルが検出されなかった場合、[図 157 : Compiler Listing Not Found 画面の例 ページ 240](#) に示す例のようなプロンプト画面が表示されます。サイド・ファイルが、もともと HFS にあった場合は、C/C++ に対してはプロンプトは出されません。

サイド・ファイルという用語は、以下のいずれかを表します。

- IBM® Application Delivery Foundation for z/OS (ADFz) LANGX サイド・ファイル
- TEST(NONE,SYM,SEPARATE) コンパイラー・オプションを使用して生成された COBOL SYSDEBUG サイド・ファイル
- TEST(STMT,SYM,NOHOOK,SEPARATE) コンパイラー・オプションを使用して生成された Enterprise PL/I サイド・ファイル

図 157. Compiler Listing Not Found 画面の例

```

Compiler Listing Not Found                               Line 1 Col 1 76
Command ==> ----- Scroll ==> CSR

Module SYS13029.T142716.RA000.IDIVPCOB.GOSET.H01(IDISCBL1) containing
COBOL program IDISCBL1 entry point IDISCBL1 compiled date 2019/01/29 time
14:27:30 does not have a matching listing/side-file.
This is the point of failure program. ❶
Select one of the following options and press Enter:
  1. (F3) Continue without compiler listing or side file for IDISCBL1
  2. Specify compiler listing or side file to use for IDISCBL1
  3. Retry search for compiler listing or side file for IDISCBL1
  4. Do not prompt again for any missing listing or side file
  5. Only prompt for the point-of-failure program listing or side file
  6. Add IDISCBL1 to your side file search exclude list

The trace of the listing/side-file search follows.

Rejected - NWILKES.IVPCB.LISTINGS(IDISCBL1)
Failed -
        OPEN error, member not found.

Rejected - DA.LISTING.COBOLE(IDISCBL1)

```

このプロンプトでは、以下の選択項目が提示されます。

#### 1.(F3) Continue without compiler listing or side file for *program-name*

コンパイラー・リストまたはサイド・ファイルを提供できない場合、このオプションを選択して、プログラムのソース・コード情報なしに処理を続行してください。または、EXIT (PF3) または CANCEL (PF12) コマンドを入力します。

#### 2.Specify compiler listing or side file to use for *program-name*

このオプションはポップアップ・パネルを表示します。このパネルでは、[図 158 : Specify Compiler Listing or Side File 画面の例 ページ 241](#) に示す例のように、現行プログラムに使用されるコンパイラー・リストまたはサイド・ファイルのデータ・セット名またはメンバー名 (PDS または PDSE データ・セットの場合) を指定でき、また Enterprise COBOL バージョン 5 の場合は DWARF デバッグ情報を含むプログラム・オブジェクトを指定できます。



図 158. Specify Compiler Listing or Side File 画面の例

```

Specify Compiler Listing or Side File                               Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR

Specify the data set and member name containing the compiler listing or
side file and press Enter.
Data Set Name . . . : 'TDEV003.LISTING.PLI'
Member . . . . . : IDISCBL1

Alternatively, place cursor on choice and press Enter to use previously
specified compiler listing or side file data set name.
==> 'TDEV003.LISTING.PLI'
==> 'TDEV003.@SDSF'
==> 'PMR.P03527.B370.C000.IDILANG'
==> 'TDEV003.JCLLIB'
==> 'TDEV003.$$TEMP$$LISTING'
==>
==>
==>
==>
==>

*** Bottom of data.

```

データ・セット名は、単一引用符で囲まれる場合を除いて、現行の TSO 接頭部を先行させる ISPF 規則に従って指定されます。

指定された最後の 10 データ・セット名が、アプリケーション ID の ISPF プロファイルに保管され、画面の初期設定に使用されます。データ・セット名フィールドの初期化には、最後に使用されたデータ・セット名が使用されますが、必要なデータ・セット名の上にカーソルを置いて Enter キーを押すことで、リストされている任意のデータ・セット名を選択できます。

メンバー名は、デフォルトで、リストまたはサイド・ファイルが必要なプログラム名に解釈されます。ご使用のリストまたはサイド・ファイルの実際のメンバー名が、プログラム名と異なる場合、このフィールドを変更する必要があります。

順次データ・セットが指定された場合、メンバー名は無視されます。

必要なデータ・セット名およびメンバー名を指定または選択したら、Enter キーを押します。

Fault Analyzer は、指定されたコンパイラ・リストまたはサイド・ファイルが適切に一致しないと判断した場合、[図 159: リスト/サイド・ファイルの不一致画面の例](#) ページ 242 に示す例のような別のプロンプトを表示します。

図 159. リスト/サイド・ファイルの不一致画面の例

```

                                                                    Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR
Listing/Side File . . . . . : LJBERRY.SYSDEBUG.COBOL(COBSEP)

Compile Date/Time:
  Load Module . . . . . : 2019/01/30 12:05:11
  Listing/Side File . . . . : 2019/01/30 12:05:58

Program COBSEP has a COBOL SYSDEBUG file signature or checksum mismatch.
The number of DATA DIVISION STATEMENTS is 11 in the side file, 10 in the
load module. The number of PROCEDURE DIVISION STATEMENTS is 15 in the side
file, 13 in the load module.

NOTE: If the compile mismatch is significant, and the file is accepted,
      then some information presented might not correctly reflect the
      conditions at the time of the fault.


Press ENTER to continue with this side file, or F3/F12 to cancel.

*** Bottom of data.
    F1=Help      F3=Exit      F5=RptFind  F7=Up      F8=Down      F12=Cancel

```

Enter

を押して、表示された一致しないコンパイラ・リストまたはサイド・ファイルを受け入れると、正しくない情報(正しくない値を持つデータ・フィールドや、正しくないソース行またはステートメントなど)が表示されるおそれがあります。

使用を試行することさえできない程度までサイド・ファイルの検証が失敗した場合、

[159: リスト/サイド・ファイルの不一致画面の例](#) ページ 242 のプロンプトに Enter

キーで応答すると、代わりにもう一度 [157: Compiler Listing Not Found 画面の例](#) ページ 240 の画面が表示され、別のサイド・ファイルを指定することができます。

### 3. Retry search for compiler listing or side file for *program-name*

このオプションを選択すると、Fault Analyzer

は、標準の検索パスを使用してコンパイラ・リストまたはサイド・ファイルの検索を繰り返します。このオプションを選択できるのは、例えば、分割画面 ISPF

セッションを介して現行プログラムを再コンパイルし、例えば、Fault Analyzer データ・セット連結でコンパイラ・リストまたはサイド・ファイルを IDILCOB に提供した後です。

この反復検索は 1 回のみ実行されます。リストまたはサイド・ファイルが検出されなかった場合であっても、同じプログラムに対して 2 回プロンプトがユーザーに出されることはありません。

### 4. Do not prompt again for any missing listing or side file

このオプションを選択した場合、Fault Analyzer は、現行の対話式再分析セッション中のコンパイラ・リストの欠落や他のプログラムのサイド・ファイルに関するプロンプトを、以後表示しません。

### 5. Only prompt for the point-of-failure program listing or side file

このオプションを選択した場合、Fault Analyzer は、現行の対話式再分析セッション中のコンパイラ・リストの欠落や他のプログラムのサイド・ファイル(障害点イベントに属すると判断された場合)に関するプロンプト

トを、以後表示しません。初期プロンプトが既に障害点プログラムに属す場合、(図 157 : Compiler Listing Not Found 画面の例 ページ 240 の ❶ のように) 画面にはこのことを示すメッセージが追加されます。

#### 6. Add program-name to your side file search exclude list

このオプションを選択すると、図 160 : Exclude Program from Side File Search 画面の例 ページ 243 に示す例のように、画面が表示され、サイド・ファイル検索除外リストに現行プログラム名を追加できるようになります。

図 160. Exclude Program from Side File Search 画面の例

```

Exclude Program from Side File Search                               Line 1 Col 1 76
Command ==> ----- Scroll ==> CSR

Press PF3 to continue without updates.

Press Enter to add IDISCBL1 to your side file search exclude list below.
Optionally, edit the name using wildcards '*' and/or 'before pressing
Enter.
Program Name. . . . : IDISCBL1

Current list of excluded programs ( Edit ):
(Empty)

*** Bottom of data.
```

更新を適用せずに続行するには、PF3

を押します。このプログラムのコンパイラ・リストまたはサイド・ファイルを指定するようにプロンプトが再度出されることがあります。

サイド・ファイル検索除外リストにプログラム名を追加するには、Enter

を押します。オプションとして、名前をより汎用性のあるものにするために、最初にワイルドカード文字\* (ゼロ個以上の文字) または % (単一の必須文字) を使用して名前を編集します。

現行のサイド・ファイル検索除外リストが表示されます。「Edit」ポイント・アンド・シュート・フィールドにカーソルを置いて Enter キーを押せば、リストを変更できます。プログラム名は有効な PDS または PDSE メンバー名でなければなりません、プログラム名には前述のワイルドカード文字を使用できます。プログラム名を指定するときに大/小文字は区別されません。

有効なプログラム名の指定例を以下に示します。

```

*XMAI*
PAYROLL0
SELOPT%
SUBRTN*
```

プログラム名除外リストは、Interactive Reanalysis Options 画面でも編集できます。詳しくは、[対話式再分析オプション ページ 162](#)を参照してください。

オプションのリストの後に、リスト/サイド・ファイル検索のトレースがあります。

IDITRACE 情報 ページ 387に示すように、このトレースは、IDITRACE DDname を使用した場合に得られる内容と同じです。

Compiler Listing Not Found 画面 ([図 157: Compiler Listing Not Found 画面の例 ページ 240](#)) が最初に表示されたときには、その時点までの検索トレース全体が表示されます。その後は、最後に Compiler Listing Not Found 画面が表示された後に書き込まれたレコードだけがトレースに含まれます。

## プロンプトの制御

対話式再分析オプション「Prompt for missing side files」は、プロンプトを出すかどうかを決定します。このオプションの詳細については、[対話式再分析オプション ページ 162](#)を参照してください。

## 対話式再分析に使用されるデータ・セット

ISPF インターフェースを使用して対話式再分析を実行する場合、リアルタイムで使用された任意の Fault Analyzer コンパイラー・リストまたはサイド・ファイルのデータ・セットに必要な事前割り振りが実行されます。リアルタイム JCL に明示的に含まれているか、DataSets オプションまたは分析制御ユーザー出口を使用して指定されている場合、Fault Analyzer データ・セットに対して割り振りが実行されます。これらのデータ・セットは、リアルタイムで使用されたのと同じ実行環境を再作成するために、再分析で使用されます。

IDIOPTS ユーザー・オプション・ファイルまたは PARM フィールドを使用して指定された DataSets オプションにより、これらのデータ・セットは、リアルタイム実行からのデータ・セットに論理的に連結されます。

“Interactive options”画面の“[Display panel to alter allocated data sets]”オプションが「Y」に設定される場合 ([対話式再分析オプション ページ 162](#)を参照)、再分析を開始する前に、リアルタイム・データ・セット指定に変更を加えることが可能です。また、リアルタイムで使用されたにもかかわらず、再分析環境に存在しないデータ・セットや、READ アクセスが禁止されているデータ・セットはすべて、次の IDILCOB の例に表示されているように、コメントにより識別されます。

```
//* The following IDILCOB data set is unavailable:
//*      DD DISP=SHR,DSN=D01.COBOL.LISTINGS
//* The following IDILCOB data set is READ protected:
//*      DD DISP=SHR,DSN=CTEST.PROTECT.LISTINGS
```



**注:** GenerateSavedReport オプションが有効な場合、リアルタイム処理中に使用されるすべてのデータ・セットまたはパス名は、再分析の実行前に疑似 JCL 画面から削除された場合でも、自動的に含まれます。

## リフレッシュ処理

リフレッシュ処理は、対話式再分析時に更新されたユーザー情報のために障害項目を再書き込みするときに行われます。このリフレッシュ処理を発生させるユーザー情報は、以下のいずれかです。

- INFO コマンドまたは「File->Fault Entry Information」アクション・バーのプルダウン・メニュー・オプションで変更されたユーザー名、ユーザー・タイトル、またはロック・フラグ。
- 追加、削除、または変更されたダンプ・ストレージ・アドレスに対するユーザー注記。

次の状況のいずれか、または両方が発生した場合

- ユーザー情報は変更されていない
- 有効な MaxMinidumpPages オプションを超えるサイズのミニダンプが最初に障害項目に追加されている

次に、[図 161](#) に示す例のような画面が表示されます。対話式再分析レポートの終了時に、リフレッシュ処理出口プロンプトの例 [ページ 245](#)が表示されます。

図 161. リフレッシュ処理出口プロンプトの例

```

Fault Entry Refresh                               Line 1 Col 1 76
Command ==> ----- Scroll ==> CSR

User information has been added or modified.

Press Enter to refresh the current history file entry, or press PF3/PF12
to cancel.

A minidump does not currently exist for this fault entry, but one will be
saved unless suppressed using the option below.

MaxMinidumpPages Option . . . : 100
Current Minidump Pages. . . : 163
Suppress Minidump . . . . . : N (Y/N)

*** Bottom of data.

```

画面で、更新されたユーザー情報の保存とミニダンプの書き込みの両方を求められた場合、ミニダンプはユーザー情報も保存されている場合にのみ書き込まれます。

選択したオプションを使用して Fault Entry List 画面に戻るには、Enter キーを押します。

デフォルトのオプション (更新されたユーザー情報を保存するが、ミニダンプは抑制します) を使用して Fault Entry List 画面に戻るには、cancel コマンドを入力します。

END コマンドや EXIT コマンドを使用してこの画面を終了することはできません。

代わりに、バッチで再分析が行われる場合は、同等の機能の実行にユーザー出口を使用できます。この出口は事実上 End Processing ユーザー出口で、RefreshExits オプションを使用して指定されます ([RefreshExits ページ 625](#) を参照)。

## COBOL Explorer

COBOL Explorer はイベント・ソース行を使用して、選択された変数の手続きトレースバック/使用が示される分岐分析を作成します。選択された変数が変更される参照は強調表示されます。この強調表示された行には、グループまたは REDEFINES による変更が含まれます。

従来型のエディター・ベースのソース・プログラム表示の代わりに、縮小表示/展開されるソース・ビューが使用されます。このビューでは、プログラムの関連部分のみが表示されます。この方法を使用すれば、大きな COBOL プログラムをより容易に操作できます。

COBOL Explorer を開始する方法は 3 つあります。

- COBOL プログラム・イベントの「イベント詳細」画面にあるポイント・アンド・シュート・フィールドから開始する。
- 「Services」プルダウン・メニューから開始する。
- 対話式再分析時に任意の場所で CE コマンドを入力することで開始する。

CE コマンドをパラメーターなしで入力したり、「Services」プルダウン・メニューから COBOL Explorer を開始したりすると、ポップアップ・リストが表示されます。そのポップアップからプログラムを選択します。

## COBOL Explorer の例

以下は Fault Analyzer の COBOL Explorer 機能を使用する方法の例です。

「Fault Entry List」画面で「I」行コマンドを使用して COBOL プログラムにおける異常終了の障害項目を対話式に分析します。「Event Summary」を選択して、そこから COBOL プログラム・イベントを選択すると、[図 162: サンプル COBOL Explorer イベント詳細画面 ページ 247](#)に示すような「Event Details」画面が表示されます。

図 162. サンプル COBOL Explorer イベント詳細画面

```

File  View  Services  Help
-----
Event 1 of 1: Abend S0CB *** Point of Failure ***                Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR
JOBNAME: COBPERF6  SYSTEM ABEND: 0CF                FAE1      2019/07/21  16

Abend Code. . . . . : S0CB
Program-Interruption Code . : 000B (Decimal-Divide Exception)
  The divisor was zero in a signed decimal division.

The source code below was executed via the following sequence of PERFORM
statements:
Source
Line #
000105          PERFORM READ-FILE UNTIL END-OF-FILE = '1'.
000122          NOT AT END PERFORM PROCESS-VEHICLE
000133          PERFORM PROCESS-CAR
000151          PERFORM CALC-TAX.

COBOL Source Code:
Source
Line #
000175          COMPUTE BASE-AMOUNT = PRICE / CC

Data Field Declarations:
Source
Line #
000033          03 PRICE          PIC 9(6).
000049          07 CC            PIC 9(4).
000094          01 BASE-AMOUNT   PIC 9(3)V99 COMP-3.

Data Field Values:
BASE-AMOUNT = 0.00
CC          = 0  *** Cause of error ***
PRICE      = 50000

COBOL Explorer ①

The listing file used for the above was found in
MY.LISTING.COBOL(COBPERF6).

:

```

① に示されている「COBOL Explorer」ポイント・アンド・シュート・フィールドを選択すると、[図 163: サンプル COBOL Explorer \[Source Line and Data Values\] 画面](#) ページ 248 の「Source Line and Data Values」画面が表示されま

す。

図 163. サンプル COBOL Explorer 「Source Line and Data Values」 画面

```

Source Line and Data Values                                     Line 1 Col 1 76
Command ==> ----- Scroll ==> CSR

Source:
000175          COMPUTE BASE-AMOUNT = PRICE / CC

Data Field Declarations:
000033          03 PRICE          PIC 9(6).
000049          07 CC            PIC 9(4).
000094          01 BASE-AMOUNT    PIC 9(3)V99 COMP-3.

Data Field Values:
⓪ BASE-AMOUNT = 0.00
⓪ CC          = 0 *** Cause of error *** ②
⓪ PRICE      = 50000

Select data fields to use with source line to create view.

*** Bottom of data.
    
```

②にあるデータ・フィールド「CC」がエラーの原因(ゼロ除算)とみなされたため、このデータ・フィールドをさらに入念に調べることとします。これを行うには、他の2つのデータ・フィールド(「BASE-AMOUNT」および「PRICE」)をブラウザで上書き入力して選択解除し、Enterを押します。その結果、[図 164: サンプル COBOL Explorer デバッグ・ビュー画面 ページ 248](#)に示すような画面が表示されます。

図 164. サンプル COBOL Explorer デバッグ・ビュー画面

```

File View Services Help
-----
Exploring COBOL Program COBEX1                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR
JOBNAME: COBEX1      SYSTEM ABEND: 0CB                      FAE1      2019/04/14  20:40:31
+ Expand all / - Collapse all  - Comments + Level 88  - Redefines

+ 000001 IDENTIFICATION DIVISION.
+ 000003 ENVIRONMENT DIVISION.
+ 000011 FILE SECTION.
+ 000017 WORKING-STORAGE SECTION.
+ 000026 01 VEHICLE-RECORD.
000038    03 SPECIFICATION.
000042      05 ENGINE.
000049      07 CC          PIC 9(4).
+ 000081 01 HEADINGS REDEFINES VEHICLE-RECORD.
+ 000084 01 WS-REC REDEFINES VEHICLE-RECORD.
+ 000098 LINKAGE SECTION.
+ 000101 PROCEDURE DIVISION USING PARMS.
000105     PERFORM READ-FILE UNTIL END-OF-FILE = '1'.
+ 000119 READ-FILE.
000122     NOT AT END PERFORM PROCESS-VEHICLE
+ 000125 PROCESS-VEHICLE.
000126     MOVE FS-REC TO WS-REC.
000133     PERFORM PROCESS-CAR
+ 000149 PROCESS-CAR.
000151     PERFORM CALC-TAX.
+ 000173 CALC-TAX.
000175     COMPUTE BASE-AMOUNT = PRICE / CC
000178     COMPUTE BASE-AMOUNT = CC / CYLINDERS
    
```



上記デバッグ・ビュー画面は、プログラムがどのようにしてソース行に至ったのかを示す分岐分析と、選択された変数 (前の「Source Line and Data Values」画面で変数が複数選択された場合は、ここでも変数は複数です) を参照するすべてのソース行で構成されます。

このビューでは、異常終了ソース行 175 がプロシージャ CALC\_TAX にあることが示されていて、分岐分析によりそのソース行への実行パスが明らかにされています。

選択された変数のほかにも、個々の項目を含むグループ項目に対する参照や、再定義も示されています。参照が変更されるソース行はいずれも強調表示されています (多くの場合、参照対象の変数を変更する変数は特に関心を集めるものであると考えられるため、そのようなソース行は新規ビューの作成に使用されます。ビューをスクロールするには、PF10/PF11 を使用してください)。

カーソルを適切な行に置いて Enter キーを押すこと (または「設定」>「ホットスポット」で使用可能な「Fnn」および「カーソル位置で ENTER」パーソナル・コミュニケーションズ・オプションを選択した場合はダブルクリックするだけ) で、次を選択できます。

- 強調表示された任意のソース行を選択して新規ビューを作成します。
- データ部セクション (WORKING-STORAGE SECTION など) を選択し、そのセクションの「Associated Storage Areas」画面を呼び出します (この画面には、すべての変数とその値が示されます)。また、「FILE SECTION」では、ISPF または File Manager を使用して、開かれているファイルを編集/参照/表示できます。
- データ項目を選択し、そのデータ項目に対する参照をすべて表示したり、その参照が行われたセクション/パラグラフを表示したりします。グループ項目およびグループ・メンバーに関しては、集約マップも表示されます。
- プロシージャ名を選択し、そのプロシージャへの分岐をすべて表示したり、その分岐が行われたプロシージャを表示したりします。
- 実行可能ソース行を選択し、そのソース行の変数の値を表示したり、オプションで、そのソース行や選択した変数の新規ビューを作成したりします。ソース行が分岐 (例: PERFORM、GO TO) の場合は、ターゲット・プロシージャが展開されます (その後でプロシージャを縮小表示すると分岐ステートメントに戻ります)。
- MQI 呼び出し (MQOPEN/MQGET/MQPUT/MQCLOSE) を選択して、File Manager WebSphere® MQ でキューを編集/参照したりキュー・マネージャーをリストしたりします。

セクション展開/縮小表示 (画面の左にある「+」記号や「-」記号にカーソルを置いて Enter を押す) は、プログラムの探索に使用されます。セクションは、FILE SECTION や WORKING-STORAGE SECTION などに加えて、手続き部における任意のセクションやパラグラフで構成されます。

データ部レベル 01 グループ項目は最初は縮小表示されています。グループ項目を展開すれば、グループ・メンバーを表示できます。

手続き部を展開して、すべてのプロシージャを表示します。その後でプロシージャを個別に展開すれば、ソースを表示できます。

グローバル展開/縮小表示 (画面の上部にある「+」記号や「-」記号にカーソルを置いて Enter を押す) を使用すれば、ソース・プログラムをさらに絞り込むことができます。

- 「Expand all」では、すべてのセクションおよびグループ項目が展開されます。(この状態で Find コマンドを使用すると、すべてのソース・レコード内が検索されることに注意してください。)
- 「Collapse all」ではビューの初期表示に戻ります。
- 「Comments」ではコメント行が展開/縮小表示されます。
- 「Level 88」に関しては、レベル 88 項目が展開/縮小表示されます。展開時は上位項目のみが表示されます。
- 「Redefines」に関しては、再定義が展開/縮小表示されます。展開時は上位項目のみが表示されます。

<https://www.youtube.com/watch?v=ZXwsaBnfk2Q> で COBOL Explorer を紹介するビデオをご覧ください。

## 据え置きブレークポイント機能

COBOL Explorer は、IPVLANGP によく似た z/OS® Debugger 据え置きブレークポイントの設定をサポートします。詳しくは、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* を参照してください。

据え置きブレークポイントを設定するには、ADFz 共通コンポーネントをインストールする必要があります。

## 第 6 章. CICS システム異常終了ダンプ分析の実行

Fault Analyzer の対話式コンポーネントには、CICS® システム異常終了に関連する情報を分析する機能があります。これはこのコンポーネントに固有のもので、



**注:** CICS® システム・ダンプの分析に必要な TSO 領域サイズは、たいていの場合、通常の障害項目の再分析に必要なサイズよりもかなり大きくなります。

以下のトピックで概説されている手順では、Fault Analyzer の対話式コンポーネントが ISPF セッションから既に開始されていることを前提としています。

### CICS システム異常終了分析用のオプションの設定

一般的な対話式再分析オプションが、CICS® システム異常終了分析にも使用されます ([対話式再分析オプション ページ 162](#)を参照)。

### ユーザー出口

CICS® システム異常終了分析は MVS™ SVC ダンプまたは SYSMDUMP データ・セットに対する再分析として実行されますので、以下のユーザー出口だけが使用可能です。

[分析制御ユーザー出口 ページ 475](#)

[コンパイラー・リスト読み取りユーザー出口 ページ 480](#)

[メッセージおよび異常終了コード説明ユーザー出口 ページ 485](#)

[フォーマット・ユーザー出口 ページ 491](#)

### CICS ダンプ・データ・セットの選択

CICS® システム異常終了 SVC ダンプまたは SYSMDUMP データ・セットを選択するには、まず Fault Entry List 画面の File メニューから、Analyze MVS™ Dump Data Set オプションを選択します (このオプションの一般的な選択方法については、[アクション・バーのプルダウン・メニュー ページ 100](#) を参照してください)。このオプションを選択すると、[図 165: Analyze MVS Dump Data Set 画面の例 ページ 252](#) のように、Analyze MVS Dump Data Set 画面が表示されます。

図 165. Analyze MVS Dump Data Set 画面の例

```

File  View  Services  Help
-----
Analyze MVS Dump Data Set                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Enter the name of a MVS SVC or SYSMDUMP data set and press Enter to initiate a
performing analysis, issue the Exit (PF3) or Cancel (PF12) command.

Dump Data Set Name. . . . . : 'cics.dump1'

*** Bottom of data.

```

この画面で、分析される CICS® システム異常終了ダンプを含む SVC ダンプまたは SYSMDUMP データ・セットの名前を入力します。データ・セット名の指定については ISPF データ・セット名規則に従います。つまり、データ・セット名を単一引用符で囲まない場合は、現行の TSO プロファイル接頭部を先頭に付けます。指定されたデータ・セットは、存在するかどうか確認が行われてから受け入れられます。この例では、データ・セット CICS.DUMP1 が分析されています。

最後に指定されたデータ・セット名が、アプリケーション ID に応じた ISPF プロファイルに保管され、画面の初期設定に使用されます。

指定された MVS™ ダンプ・データ・セットの妥当性検査が行われると、Fault Analyzer CICS® システム異常終了分析が開始され、「Analyzing MVS dump data set. Please wait...」というメッセージ (図 166: 分析開始の認識 ページ 253) が表示されます。

図 166. 分析開始の認識

```

File  Options  View  Services  Help
-----
IBM Fault Analyzer - Fault Entry List                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Fault History File or View : 'NWILKES.DEMO.HIST'

{The following line commands are available: ? (Query), V or S (View saved
report), I (Interactive reanalysis), B (Batch reanalysis), D (Delete), H
(Duplicate history), C (Copy fault entry), M (Move fault entry), X (XMIT fault
entry).}

  Fault ID Job/Tran User ID  Sys/Job  Abend Date      Time
  ---
  F00323 IDIVPCOB  NWILKES  MVS2    S0C7  2019/12/21 13:02:25
  F00445 ALLANT01  JACKIED  MVS8    S0C7  2019/12/19 03:29:57
  F00442 ALLANT01  ALLANT   MVS8    S0C7  2019/09/10 22:20:10
  F00349 CS05      CICSUSER CSCB0050 ASRA   2019/08/23 07:47:23
  F00348 CS04      CICSUSER CSCB0040 ASRA   2019/08/23 07:46:36
  F00345 CS01      CICSUSER CSCB0010 AEIL   2019/08/23 07:43:35
  F00050 PSTRANDR  PSTRAND  STPLEX4B S0C4   2019/08/02 17:03:18
  F00035 CICS53     n/a      MVS2    n/a    2019/04/05 14:49:11
  F00034 CI
  F00294 DB
  Analyzing MVS dump data set. Please wait...
  F1=Help      F8=Down      F10=Left    F11=Right   F12=MatchALL  F7=Up

```

## 分析対象のアドレス・スペースの選択

マルチアドレス・スペースを含むダンプを分析する場合、[図 167: CICS システム・ダンプ分析のアドレス・スペースの選択](#) ページ 253 の例に示すように選択リストが表示されます。このリストの目的のアドレス・スペースに「S」を指定し、Enter キーを押すと選択できます。CICS@ がアクティブであると検出されたアドレス・スペースには、ジョブ・タイプ「CICS」がマークされます。

図 167. CICS システム・ダンプ分析のアドレス・スペースの選択

```

File  Options  View  Services  Help
-----
Address Space Selection
-----
I C Command ==> ----- Row 1 to 2 of 2 80
  Please use S to select the ASID to analyze.  R
  F ASID Jobname Job Type
  { _ X'004F' CICS5ANS CICS
  r _ X'008A' IDISS
  ***** Bottom of data *****

  F1=Help  F3=Exit  F7=Up  F8=Down  F12=Cancel

  SW00009 ICCB0010 NWILKES  MVS2    S0CB  2019/11/26 14:29:18
  SW00008 ICCB0010 NWILKES  MVS2    S0CB  2019/11/25 14:07:15
  SW00007 ICCB0010 NWILKES  MVS2    S0CB  2019/11/25 09:29:18
  SW00006 ICCB0010 NWILKES  MVS2    S0CB  2019/11/21 10:51:16
  SW00005 ICCB0010 NWILKES  MVS2    S0CB  2019/11/21 10:46:00
  SW00004 ICCB0010 NWILKES  MVS2    S0CB  2019/11/21 10:35:29
  SW00003 ICCB0010 NWILKES  MVS2    S0CB  2019/11/21 10:19:27
  SW00002 ICCB0010 NWILKES  MVS2    S0CB  2019/11/21 08:46:35
  SW00001 ICCB0010 NWILKES  MVS2    S0CB  2019/11/21 08:33:53
  F1=Help  F3=Exit  F4=MatchCSR  F5=RptFind  F6=Actions  F7=Up
  F8=Down  F10=Left  F11=Right   F12=MatchALL

```

アドレス・スペースを 1 つのみ含むダンプの場合、Address Space Selection 画面は表示されません。

## CICS システム異常終了対話式レポートの表示

CICS® システム異常終了分析が完了すると、CICS® システム異常終了対話式レポート (図 168: CICS System Abend Interactive Reanalysis Report 画面の例 ページ 254) が表示されます。

図 168. CICS System Abend Interactive Reanalysis Report 画面の例

```

File  View  Services  Help
-----
CICS System Abend Interactive Reanalysis Report                Line 1 Col 1 80
Command ==>          Scroll ==> CSR
CICS DUMP: SYSTEM=CICSDI  CODE=SM0002  ID=          D381  2019/08/13 09:55:03

Select one of the following options and press Enter to access further fault
information:
  1. Synopsis
  2. Abend Job Information
  3. CICS System Information
  4. Options in Effect

DFHSM0002 CICSDI A severe error (code X'030E') has occurred in module DFHSMGF.

Severity 3 Observations

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

CICS® システム異常終了の初期の対話式レポート画面は、他の障害タイプの画面とは異なります。内容が論理的に次のセクションに分割されます。

### 使用可能なオプション

画面の一番上にオプションがあります。このオプションを選択するには、オプション番号にカーソルを置き、Enter キーを押します。タブ・キーを使用して、カーソルを目的のオプションに置くことができます。個々のオプションについては、下記のセクションで説明します。

### ダンプの理由

分析されたダンプが取得された理由が入手可能な場合は、この理由がオプションのリストの後に続きます。

### 分析の観測

ダンプの理由の後に、分析時に行われた観測 (ある場合) が続きます。これらの観測は、重大度に基づいて次の 3 つのカテゴリにおおまかに分類されます。

#### 重大度

##### 説明

1

障害の原因と考えられる問題

2

障害の理由である可能性がある問題

10 個以上の重大度 2 の観測が使用可能な場合は、代わりに、別の画面を表示するポイント・アンド・シュート・リンクが表示されます。

### 3

通常、障害の理由と見なされず、単に該当項目である問題

5 個以上の重大度 3 の観測が使用可能な場合は、代わりに、別の画面を表示するポイント・アンド・シュート・リンクが表示されます。

## ファースト・パス・ナビゲーション

CICS® システム・ダンプ分析画面でのナビゲーションを容易にするには、任意のコマンド行にファースト・パス・コマンドを入力します。CICS® System Information 画面で強調表示されているフィールドはすべて、有効なファースト・パス・コマンドです (例を [図 171 : CICS System Information 画面の例 ページ 257](#) に示します)。

ファースト・パス・コマンドの接頭部に感嘆符 (!) を付加して入力すると、CICS® System Abend Interactive Reanalysis Report 画面に戻ってファースト・パス・コマンドを入力するのと同等になります。つまり、ISPF で jump コマンドを使用すると類似しています。

## オプション 1: Synopsis

「CICS® システム異常終了対話式再分析レポート」画面からオプション 1 を選択すると、[図 169 : CICS システム異常終了 Synopsis 画面の例 ページ 255](#)に表示されている例のように、「CICS® システム異常終了 Synopsis」画面が表示されます。

図 169. CICS システム異常終了 Synopsis 画面の例

```

File View Services Help
-----
Synopsis                                     Line 1 Col 1 80
Command ==>                               Scroll ==> CSR
CICS DUMP: SYSTEM=CICSDI  CODE=SM0002  ID=      D381      2019/08/13 09:55:03

Fault Information:

CICS Product Level . . . . . : V5 R3 M0
Dump ID. . . . . : 4/0007
Dump Code. . . . . : SM0002
Date/Time. . . . . : 2019/08/13 09:55:03 (Local)
Message. . . . . : DFHSM0002 CICSDI A severe error (code X'030E')
                  has occurred in module DFHSMGF.
Symptoms . . . . . : PIDS/565501800 LVLS/530 MS/DFHSM0002
                  RIDS/DFHSMGF PTFS/UQ52744 PRCS/0000030E
Title. . . . . : n/a
Caller Address . . . . . : n/a

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

この画面は、分析されたダンプについての詳細を表示します。

## オプション 2: 異常終了ジョブ情報

「CICS® システム異常終了対話式再分析レポート」画面からオプション 2 を選択すると、[図 170: Abend Job Information 画面の例](#) ページ 256 に表示されている例のように、「異常終了ジョブ情報」画面が表示されます。

図 170. Abend Job Information 画面の例

```

File View Services Help
-----
Abend Job Information                               Line 1 Col 1 80
Command ==>                                       Scroll ==> CSR
SYSTEM=C72E1FA1 CODE=CHAN ID=1/0002 FAE1 2019/10/10 08:42:25

IBM Fault Analyzer Abend Job information:

Dump Date (Local) . . . . . : 2019/10/10
Dump Time (Local) . . . . . : 08:42:25
Dump Date (GMT) . . . . . : 2019/10/10
Dump Time (GMT) . . . . . : 00:42:25
System Name . . . . . : FAE1
Subsystem Info. . . . . : CICS V7 R2 M0 (TS 5.5)
Job Name. . . . . : AS720F1
Job Step Name . . . . . : AS720F1
Exec Program Name . . . . . : DFHSIP
Requested Region Size . . . . . : 0M
User id . . . . . : COSMICK

Execution Environment:

Operating System. . . . . : z/OS      V02R03M00
Data Facility Product . . . . . : DFSMS z/OS V2R3M0
CPU Model . . . . . : 2965
F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

この画面には、異常終了が発生したときに存在していた環境についての情報が表示されます。

## オプション 3: CICS System Information

「CICS® システム異常終了対話式再分析レポート」画面からオプション 3 を選択すると、[図 171: CICS System Information 画面の例](#) ページ 257 に表示されている例のように、「CICS® システム情報」画面が表示されます。



図 171. CICS System Information 画面の例

```

File View Services Help
-----
CICS System Information                               Line 1 Col 1 80
Command ==> ----- Scroll ==> CSR

Select one of the following options and press Enter:

  1.  CICS Task Summary
  2.  Error History
  3.  Storage Usage by Task

AI  - AutoInstall Manager           AP  - Application Domain
BR  - Bridge Information            CC  - Catalog Domains
CQ  - Console Queue Component      CSA - Common System Area
DB2 - DB2 Information              DD  - Directory Domain
DH  - Document Handler Domain      DLI - DL/I Information
DM  - Domain Manager              DP  - Debug Profile Domain
DS  - Dispatcher Domain           DU  - Dump Domain
EJ  - Enterprise Java Domain      FC  - File Control
IC  - Interval Control            IS  - ISC/IP Domain
KE  - Kernel Domain              LD  - Loader Domain
LG  - Log Manager Domain          LM  - Lock Manager Domain
ME  - Message Domain             MN  - Monitoring Domain
MRO - Multiregion Option          NQ  - Enqueue Domain
OT  - Object Transaction Domain   PA  - Parameter Domain
PG  - Program Manager Domain     PI  - Pipeline Manager Domain
PR  - Partner Resource Manager    PT  - Partner Domain
RM  - Recovery Manager Domain     RS  - Region Status Domain
RZ  - Request Stream Domain      SIT - System Initialization Table
SJ  - SJ (JVM) Domain            SM  - Storage Manager Domain
SO  - Sockets Domain             SSA - Static Storage Areas
ST  - Statistics Domain          TCP - Terminal Control Definitions
TD  - Transient Data Domain      TI  - Timer Domain
TMP - Table Manager              TR  - Trace Domain
TS  - Temporary Storage Domain   US  - User Domain
UEH - Global User Exit Details   WB  - Web Domain
XM  - Transaction Manager Domain  XS  - Security Domain

LCK - Lock Owner/Waiter Information
TRC - CICS Trace
NMT - MVS Name/Token Pairs

*** Bottom of data.

```

この画面には、特定の CICS® システム情報を選択するためのオプションが表示されます。

## テーブル表示のソートと突き合わせ

テーブル見出しがタブ・キーで移動可能であり、反転強調表示モードで表示される場合は必ず、この強調表示はテーブル列属性が変更可能であることを意味しています。見出しにカーソルを置いて Enter を押すと、列属性画面が表示されます。この画面では、列データを昇順または降順にソートでき、また指定された MATCH 基準を満たすテーブル行のみ表示できます。

MATCH 属性では大/小文字は区別されません。またワイルドカードを使用できます。サポートされているワイルドカード文字には、アスタリスク (\*) (0 個以上の文字を表す) と % 記号 (単一の必須文字を表す) があります。指定されたストリングに一致するテーブル・データ行のみが表示されます。

すべての属性設定は累積され、ある列に特定のソート順を適用すると、既にソートされているテーブル・データに対して、異なる列のソートが実行されます。また、一致するものがない突き合わせ基準のために、テーブル・データが表示から削除された場合、このデータを復元する唯一の方法はリセットを実行することです。

リセットは、リセット・ポイント・アンド・シュート・フィールドにカーソルを置き、Enter キーを押すことによって実行できます。または、コマンド行に RESET コマンドを入力することにより、指定された表示のすべてのテーブルを同時にリセットすることができます。

属性が変更された列は、青色ではなく、青緑色で表示されます。

最新の属性設定は、列見出しが選択されるたびに表示されます。

## オプション 4: 有効になっているオプション

「CICS® システム異常終了対話式再分析レポート」画面からオプション 4 を選択すると、[図 172: Options in Effect 画面の例](#) ページ 258 に表示されている例のように、Options in Effect 画面が表示されます。

図 172. Options in Effect 画面の例

```

File View Services Help
-----
Options in Effect                                     Line 1 Col 1 80
Command ==>                                         Scroll ==> CSR
CICS DUMP: SYSTEM=CICSDI  CODE=SM0002  ID=      D381  2019/08/13 09:55:03

Fault Analyzer Options in Effect:

{These are the options that were used to generate the current interactive
reanalysis report. To change any options, first return to the Fault Entry
List display and select "Interactive Reanalysis Options" from the "Options"
action-bar pull-down menu; then perform interactive reanalysis again.}

DumpDSN(CICS.DUMP1)
NoErrorHandler
Language(ENU)
NoLocale
NoPermitLangx

Data Sets:

{The following Fault Analyzer data set or path names were either
preallocated, specified via DataSets options, or provided as defaults.}

DDname  Data Set or Path Name
IDIADATA PROD.SYSADATA
IDIDOC  IDI.SIDIDOC1
IDIDSECT IDI.DSECTS
IDIEEXEC IBMUSER.EXEC
IDIHIST  IDI.HIST
IDILC    PROD.LISTING.C
IDILCOB  PROD.LISTING.COBOL
IDILPLI  PROD.LISTING.PLI
IDIMAPS  IDI.SIDIMAPS
IDIVSENU IDI.IDIHVENU

Exits:

{The following user exits were specified via Exits options.}

Type      Name      Type Invoked
NOTIFY    NOTIFY    REXX (Not applicable)

```

この画面には、この再分析レポートに有効な Fault Analyzer オプションについての情報が表示されます。

## ヒストリー・ファイル項目の作成

CICS® System Abend Interactive Reanalysis Report 画面を終了すると、[図 173 : Create History File Entry 画面の例](#) ページ 259 に示されているように、ヒストリー・ファイル項目を作成するためのオプションが表示されます。

図 173. Create History File Entry 画面の例

```

File View Services Help
Create History File Entry

To create a history file entry for the analyzed MVS dump data set, specify
a history file data set name and press Enter.

History file DSN . . 'NWILKES.DEMO.HIST'

Minidump pages . . : 29
Suppress minidump . . N (Y/N)

F1=Help      F3=Exit      F12=Cancel

DFHSM0002 CICSDI A severe error (code X'030E') has occurred in module DFHSMGF.

Severity 3 Observations

*** Bottom of data.

F1=Help      F3=Exit      F5=RptFind   F6=Actions   F7=Up        F8=Down
F10=Left     F11=Right

```

ヒストリー・ファイル項目を作成するには、以下のようになります。

1. 必要に応じて、指定されているヒストリー・ファイル名を変更します (ISPF データ・セット名指定規則に従います)。
2. オプションで「[Suppress Minidump]」オプションを「Y」に変更して、指示されたページ数を持つミニダンプが作成されてはならないことを示します。
3. Enter キーを押します。

「History file DSN」フィールドは、デフォルトで、「Fault Entry List」画面で選択された現行ヒストリー・ファイル名を使用して初期設定されます。ビューが使用中である場合は、ビュー・メンバー内に指定されている最初のヒストリー・ファイル名が使用されます。ヒストリー・ファイル項目を作成しない場合は、F3 または F12 を押してください。

ヒストリー・ファイル項目が作成された場合、メッセージ [ID10003I ページ 696](#) が発行され、割り当てられた障害 ID について通知されます。障害項目が作成されたヒストリー・ファイルは現行ヒストリー・ファイルとして自動的に選択されます。MATCH FAULT\_ID *fault\_id* コマンドが発行され、新規に作成された障害項目のみが表示されます。このコマンドが発行されるのは、新規に作成された障害項目が、日付順に配列された Fault Entry List 画面の先頭には表示されずに見つけづらいことがあるためです。ヒストリー・ファイル内の障害項目をすべて再表示するには、MATCH ALL コマンド (通常は PF12 にマップされている) を入力します。

CICS® システム異常終了の障害ヒストリー項目を作成し、ミニダンプを作成することで、今後の障害再分析のパフォーマンスが向上します。

## 第7章. CICS 補助トレース・データ・セットのフォーマット設定

### CICS 補助トレース・データ・セットの選択

CICS® 補助トレース・データ・セットを選択するには、まず Fault Entry List 画面の File メニューから、Format CICS Auxiliary Trace Data Set オプションを選択します (このオプションの一般的な選択については、[アクション・バーのプルダウン・メニュー ページ 100](#)を参照してください)。このオプションを選択すると、[図 174: Format CICS Auxiliary Trace Data Set 画面の例 ページ 260](#) に示すような Format CICS Auxiliary Trace Data Set 画面が開きます。

図 174. Format CICS Auxiliary Trace Data Set 画面の例

```
File View Services Help
-----
Analyze MVS Dump Data Set                               Line 1 Col 1 80
Command ==>> ----- Scroll ==>> CSR

Enter the name of a CICS auxiliary trace data set and press Enter to initiate
formatting. To return from this display without formatting trace, issue the
Exit (PF3) or Cancel (PF12) command.

Trace Data Set Name . . . . : 'cics.trace1'

*** Bottom of data.
```

この画面で、フォーマット設定する CICS® 補助トレースが入っているデータ・セットの名前を入力します。データ・セット名の指定については ISPF データ・セット名規則に従います。つまり、データ・セット名を単一引用符で囲まない場合は、現行の TSO プロファイル接頭部を先頭に付けます。指定されたデータ・セットは、存在するかどうか確認が行われてから受け入れられます。この例では、データ・セット CICS.TRACE1 がフォーマット設定されます。

最後に指定されたデータ・セット名が、アプリケーション ID に応じた ISPF プロファイルに保管され、画面の初期設定に使用されます。

指定されたトレース・データ・セット名の妥当性検査が完了すると、Fault Analyzer CICS 補助トレース・フォーマット設定が開始され、「Processing auxiliary trace data set. Please wait... (補助トレース・データ・セットの処理中です。お待ちください。)」というメッセージが表示されます ([図 175: トレース処理開始の認識 ページ 261](#))。

図 175. トレース処理開始の認識

```

File  Options  View  Services  Help
-----
Format CICS Auxiliary Data Set                               Line 1 Col 1 80
Command ==> _____ Scroll ==> CSR

Enter the name of a CICS auxiliary trace data set and press Enter to initiate
formatting. To return from this display without formatting trace, issue the
Exit (PF3) or Cancel (PF12) command.

Trace Data Set Name . . . . : 'CICS.TRACE1'
*** Bottom of data.

Processing auxiliary trace data set. Please wait...

```

## CICS トレース選択パラメーターの指定

初期トレース処理が終了すると、[CICS Trace Formatting ページ 186](#)に示された例のような「CICS® トレース選択パラメーター」画面が表示されます。この画面で、表示する内部トレース項目および必要なフォーマット設定のレベルを指定できます。

## 第 8 章. Java 分析の実行

Fault Analyzer の対話型コンポーネントに固有なフィーチャーとして、Java™に関連した情報を提供する機能があります。Java™ は MVS™ 上の WebSphere® Liberty、CICS®、または Unix System Services 下で実行できます。通常、環境は Java™ が呼び出すレガシー・プログラムです。これらのトピックでは、次の方法について説明します。

- Java™ 分析用のオプションの設定
- 分析目的での Java™ ダンプ・データ・セットまたは既存の Java™ 障害項目の選択
- 対話式レポートでの Java™ の結果情報の表示

以下に概説されている手順では、Fault Analyzer の対話式コンポーネントが ISPF セッションから既に開始されていることを想定しています。



**注:** Fault Analyzer サポートされるアプリケーション環境 ページ 27 に示されている JVM のみ、Java™ 分析との互換性があります。

### Java™ 分析用のオプションの設定

一般的な対話式再分析オプションが、Java™ 分析にも使用されます ([対話式再分析オプション ページ 162](#) を参照)。

IDIJVM サブシステム開始 JCL に追加されたオプションの IDIS DD ステートメントによって、Java™ 分析を実行するためにデフォルト JVM を使用できます (詳しくは、[IDIS Java のサブシステム要件 ページ 325](#)を参照)。

### Java ダンプ・データ・セットの選択

分析対象の Java™ ダンプ・データ・セットの指定は、最初に **Fault Entry List** 画面から **File > Analyze MVS Dump Data Set** を選択することによって行われます。

このオプションにより、Analyze MVS™ Dump Data Set 画面が表示され、この画面で MVS™ ダンプ・データ・セット名を入力できます。[CICS ダンプ・データ・セットの選択 ページ 251](#) に、この画面の例が表示されています。

分析中のダンプで Java™ アクティビティが検出された場合、非同期 DTFJ 処理が実行されている間に、ヒストリー・ファイルの障害項目を早期に作成する必要があります。そのため、[図 176 : Create Java Fault Entry 画面の例 ページ 263](#) に示すような **Create Java Fault Entry** 画面が表示されます。

図 176. Create Java Fault Entry 画面の例

```

Create Java Fault Entry                               Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR

To create a fault entry for this Java dump, specify a history file data
set name and press Enter, or press PF3/PF12 to cancel the fault entry
creation and continue the analysis with incomplete Java information.

History file DSN. . . . . : 'NWILKES.HIST'
*** Bottom of data.

```

IDIS サブシステムによって管理されており、かつ UPDATE アクセス権限以上の権限が自分にあるヒストリー・ファイルを指定します。Enter キーを押して障害項目を作成し、**Fault Entry List** 画面に戻ります。MATCH が自動的に実行され、作成されたばかりの障害項目のみが表示されます。



**注:** Fault Analyzer Java™ DTFJ 処理は、MVS™ ポスト・ダンプ出口 IDIXTSEL がインストールされ (詳細については [MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#) を参照)、IDIS サブシステムが始動されている (詳細については [Fault Analyzer IDIS サブシステムの使用 ページ 319](#) を参照) ことを必要とします。

DTFJ\_Status 列が表示されるように **Fault Entry List** で選択している場合は、Enter キーを押し続けることによって値を更新し、DTFJ\_Status に **Finished** と表示されるのを待つことができます。この時点までに、非同期 DTFJ 処理によって障害項目に情報が追加されます。このため、1 行コマンドを使用する後続の再分析では、使用可能な Java 情報をすべて含む分析レポート一式が表示されるようになります。

**Create Java Fault Entry** 画面で PF3 または PF12 を押すと、早期障害項目作成がスキップされますが、分析は続行されます。この場合、ダンプ分析は Java™ アクティビティーが検出されなかったかのようにして実行され、分析が終了した時点で、代わりに、ユーザーに障害項目を作成するように要求するプロンプトが出されます。

## Java 障害項目の再分析

再分析レポートでは、Java™ ダンプの分析から作成された障害項目、または Java™ 異常終了のリアルタイム分析から作成された障害項目のみに、Java™ 情報が含まれています。



**注:** Java™ アプリケーションからの Fault Analyzer のリアルタイム起動については、[リアルタイム分析 ページ 33](#) で説明しています。

対話式再分析を実行するには、ヒストリー・ファイルの障害項目に対して行コマンド「I」を使用します。

障害項目の非同期 DTFJ 処理がまだ完了していない場合は、[図 177 : Confirm Java Fault Entry Reanalysis 画面の例 ページ 264](#)の例に示すような Confirm Java™ Fault Entry Reanalysis 画面が表示されます。

図 177. Confirm Java Fault Entry Reanalysis 画面の例

```

Confirm Java Fault Entry Reanalysis                               Line 1 Col 1 76
Command ==> _____ Scroll ==> CSR_

Java DTFJ processing has not yet completed for this fault entry.

Press Enter to continue reanalysis with incomplete Java information, or
press PF3/PF12 to cancel.

*** Bottom of data.

```

Enter キーを押すと、再分析が続行されますが、Java™ 情報は不完全です。

代わりに PF3 または PF12 を押すと、再分析は取り消されます。その後、再分析を実行した時点で、非同期の Java™ DTFJ 処理が完了していることが分かると考えられるため、このプロンプトを表示せずに分析を続行します。

障害項目の非同期 DTFJ 処理が妥当な時間内に完了しなかった場合は、Confirm Java™ Fault Entry Reanalysis 画面の代わりに、メッセージ [IDI0177E ページ 736](#) が表示されます。



**注:** DTFJ 処理が完了しない理由には、以下のいずれかが考えられます。

- MVS™ ポスト・ダンプ出口 IDIXTSEL がインストールされていない (詳細については [MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#) を参照)。
- IDIS サブシステムが始動されていない (詳細については [Fault Analyzer IDIS サブシステムの使用 ページ 319](#) を参照)。

## 対話式レポートへの Java 情報の表示

Java™ の分析が完了すると、[図 178 : Java の Interactive Reanalysis Report 画面の例 ページ 265](#) に示すように、通常の Interactive Reanalysis Report 画面が表示されます。



図 178. Java の Interactive Reanalysis Report 画面の例

```

File View Services Help
-----
Interactive Reanalysis Report                               Line 1 Col 1 80
Command ==>                                               Scroll ==> CSR
JOBNAME: BPXBATC3  SYSTEM ABEND: 0CB                      FAE3      2010/03/16  19:25:50
User Title: Java

Fault Summary:
Module /u/rturner/kenichiExample/j2c2cob/libMyDllLib.so, program MYCOB1,
offset X'3CE': Abend S0CB (Decimal-Divide Exception).

Select one of the following options to access further fault information:
 1. Synopsis
 2. Event Summary
 3. Java Information
 4. Storage Areas
 5. Messages
 6. Language Environment Heap Analysis
 7. Abend Job Information
 8. Fault Analyzer Options

{Fault Analyzer maximum storage allocated: 9.12 megabytes.}

*** Bottom of data.

```

Interactive Reanalysis Report 画面から Event Summary オプションを選択すると、Java™ イベントと Java 以外のイベントの両方が含まれています。図 179 : Java Event Summary 画面の例 ページ 266 に例を示します。

図 179. Java Event Summary 画面の例

```

File View Services Help
-----
Event Summary
Command ==>
Full Application only - JOBNAME: BPXBATC3 SYSTEM ABEND: 0CB FAE3 2010/03/16
Top of data
Scroll ==> CSR

{The following events are presented in chronological order.}

Event      Fail  Module  Program  EP
#  Type    Point Name     Name     Name     Event Location (*)  Loaded
1  Call    BPXINLPA n/a      n/a      n/a      M+49626             LPA
2  Call    n/a      n/a      n/a      n/a      n/a                 Not de
3  Call    CEEBINIT n/a      CEEOPCMM E+8F8           LPA
4  >>> XPLink CEEPLPKA n/a      n/a      n/a      M+1BBB62           LPA
5  Call    java     JavaMain JavaMain P+34CC E+19EC           /apc/j
6  Java    n/a      n/a      Java2C2CobolExample.main L#10             Not de
7  Call    libj9prt24.so
      j9sig_protect
      j9sig_protect
      P+12DA E+5B2             /apc/j
8  Call    libj9vm24.so
      gpCheckCallin
      signalProtectAndRunGlue
      P+29FC E+14             /apc/j
9  Call    libj9vm24.so
      gpCheckCallin
      gpProtectedRunCallInMethod
      P+2C1A E+2A             /apc/j
10 Call    n/a      n/a      RUNCALLINMETHOD
      n/a
      E+1026             Not de
11 <<< XPLink CEEPLPKA n/a      CEEVRONU E+1026           LPA
12 Call    libMyDllLib.so
      *Java_Ja Java_Java2C2CobolExample_callCobol
      P+3F8 E+90             /u/rtu
13 Call    libMyDllLib.so
      *Java_Ja doSomething1
      P+340 E+90             /u/rtu
14 Call    libMyDllLib.so
      *Java_Ja doSomething2
      P+288 E+90             /u/rtu
15 Call    libMyDllLib.so
      *Java_Ja doSomething3
      P+142 E+92             /u/rtu
16 Abend S0CB ***** libMyDllLib.so
      MYCOB1 MYCOB1 P+3CE E+3CE           /u/rtu

NOTE: Program names prefixed '*' are pseudo CSECT names created to help
      determine in which compile unit an entry point belongs.

(*) One or more of the following abbreviations might appear in the "Event
      Location" column:

      F#n Source file number (refer to detailed event information for file
          identification)
      L#n Source file line number
      S#n Listing file statement number (refer to detailed event information
          for file identification)
      M+x Offset from start of load module
      P+x Offset from start of program
      E+x Offset from start of entry point
  
```

上の例から Java™ イベント #6 を選択すると、図 180: Java Event Details 画面の例 ページ 267 の例に示すような Java™ Event Details 画面が表示されます。

図 180. Java Event Details 画面の例

```

File View Services Help
-----
Event Summary
Command ==> _____ Top of data
                                Scroll ==> CSR

Previous Event Details

This event occurred in Class Java2C2CobolExample Method main.

Java source from /u/rturner/kenichiExample/j2c2cob/Java2C2CobolExample.java:
Source
Line #
-5  }
-4
-3  public static void main(String args[]) {
-2      System.out.println("Hello World from Java program!");
-1      Java2C2CobolExample e = new Java2C2CobolExample();
000010      e.callCobol();
+1  }
+2  }
+3
+4

The class static variable information is not available.
The object instance variable information is not available.

Next Event Details

*** Bottom of data.

```

Java™ に固有の情報は、Interactive Reanalysis Report 画面から「Java™ Information」オプションを選択することで検索できます。

上記の結果から「Java™ Information」ポイント・アンド・シュート・フィールドを選択すると、[図 181 : Java Information 画面の例 \(1/2\) ページ 268](#) に例が表示されているような Java™ Information 画面が表示されます。

## 図 181. Java Information 画面の例 (1/2)

```

File View Services Help
-----
Java Information                                     Line 1 Col 1 80
Command ==>                                         Scroll ==> CSR
JOBNAME: BPXBATC2  SYSTEM ABEND: 0CB                FAE2      2009/10/02  06:53:54

Java Version. . . . . : JRE 1.8.0 z/OS s390-31 (build 8.0.5.36 -
                        pmz3180sr5fp36-20190510_01(SR5 FP36))

Java environment variables
- _envvar
-----
IBM_JAVA_COMMAND_LINE=java Java2C2CobolExample
LIBPATH=/apc/java800-UK35911/usr/lpp/java/J8.0/lib/s390/j9vm:/apc/java800-UK
1/usr/lpp/java/J8.0/./lib/s390:/apc/java800-UK35911/usr/lpp/java/J8.0/lib/s
31-UK18621/usr/lpp/java/J8.0/bin:/apc/java800-UK18621/usr/lpp/java/J8.0/bin/
/j9vm:/apc/java800-UK35911/usr/lpp/java/J8.0/lib/s390
_BPX_SHAREAS=NO
JAVA_HOME=/apc/java800-UK18621/usr/lpp/java/J8.0
HOME=/u/rturner
LOGNAME=RTURNER
SHELL=/bin/sh
_IDIZZDBG_ZFS=1
_CEE_RUNOPTS=POS(ON),TERMTHDACT(UAIMM),TRAP(ON,NOSPIE)
_CREATE_LAYOUT=Y
JAVA_DUMP_OPTS=ONANYSIGNAL(ALL)
TZ=UTC0
CLASSPATH=./u/rturner/kenichiExample/j2c2cob
_BPXK_MDUMP=./RTURNER.JAVA.MDUMP01
_EDC_PTHREAD_YIELD=-2
STEPLIB=RTURNER.A0.LOAD
PATH=/apc/java800-UK18621/usr/lpp/java/J8.0/bin/./apc/java800-UK35911/usr/
_=/apc/java800-UK35911/usr/lpp/java/J8.0/bin/java

```

図 182. Java Information 画面の例 (2/2)

```

Java VM init args
-----
- args

arg=-Xjcl:jclscar_24
arg=-Dcom.ibm.oti.vm.bootstrap.library.path=/apc/java800-UK35911/usr/lpp/jav
arg=-Dsun.boot.library.path=/apc/java800-UK35911/usr/lpp/java/J8.0/lib/s390
arg=-Djava.library.path=/apc/java800-UK35911/usr/lpp/java/J8.0/lib/s390:/apc
c/java800-UK35911/usr/lpp/java/J8.0/lib/s390:/apc/java800-UK35911/usr/lpp/ja
a/J8.0/lib/s390:::/u/rturner/kenichiExample/j2c2cob:/apc/java800-UK18621/u
va/J8.0/bin/j9vm:/apc/java800-UK35911/usr/lpp/java/J8.0/lib/s390/j9vm:/apc/j
arg=-Djava.home=/apc/java800-UK35911/usr/lpp/java/J8.0
arg=-Djava.ext.dirs=/apc/java800-UK35911/usr/lpp/java/J8.0/lib/ext
arg=-Duser.dir=/u/rturner
arg=_j2se_j9=71168
arg=-Xdump
arg=-Djava.class.path=./u/rturner/kenichiExample/j2c2cob
arg=-Dsun.java.command=Java2C2CobolExample
arg=-Dsun.java.launcher=SUN_STANDARD
arg=_port_library

Java threads with traceback information

Call trace for thread: main

Method                               Location
Java2C2CobolExample.callCobol        Native Method
Java2C2CobolExample.main              Java2C2CobolExample.java:10

Call trace for thread: Signal Dispatcher

Method                               Location
com.ibm.misc.SignalDispatcher.waitForSignal Native Method
com.ibm.misc.SignalDispatcher.run      SignalDispatcher.java:66

```

## Fault Analyzer イベント・リストの Java イベント置換

異常終了に Java™ イベントが含まれるとき、Fault Analyzer は JVM のシステム・ダンプを要求します。Java™ システム・ダンプは、Diagnostic Tool Framework for Java (DTFJ) API を使用して Fault Analyzer により分析されます。

DTFJ 処理は非同期に発生し、最初のリアルタイム異常終了分析は遅延しません。つまり、リアルタイム Fault Analyzer レポートには内部 JVM イベントだけが含まれ、Java™ イベントは含まれないということです。

DTFJ 処理が完了すると、追加の Java™ 情報が障害項目に追加され、障害項目の再分析が実行されるときに、より簡潔で有用な Java™ イベント情報が代わりに表示されるようになります。

例えば、リアルタイム・イベント概要には次のものが表示される可能性があります。

Event #	Type	Fail Point	Module Name	Program Name	EP Name	Event Location (*)	Description
1	Call		JVMLDM71	CEER00TA	n/a	P+D4	BOOTSTRAP MODULE FOR LE; From CTEST.AUTHLOAD
2	Call		CEEPLPKA	n/a	CEEBBEXT	E+1D2	BOOTSTRAP MODULE FOR Language Environment; From LPA
3	Call		CELHV003	n/a	EDCZHINV	E+B4	CRTL Main invocation event XPLINK; From CEE.SCEERUN2
4	>>> XPLink		CEEPLPKA	n/a	CEEVROND	E+127E	Run on down stack swap; From LPA

5	Call	JVMLDM71	JzosVM#C	main	E+A6	From CTEST.AUTHLOAD
6	Call	JVMLDM71	JzosVM#C	JzosVM::run(int,char**)	E+34A	From CTEST.AUTHLOAD
7	Call	JVMLDM71	JzosVM#C	JzosVM::invokeMain()	E+50A	From CTEST.AUTHLOAD
8	Call	JVMLDM71	JzosVM#C	JNIEnv::CallStaticVoidMethod(_jclass*,_jmethodID*,...)	E+30	From CTEST.AUTHLOAD
9	Call	libj9vm27.so	n/a	callStaticVoidMethodV	E+4E	
						From /apc/java710/31bit/usr/lpp/java/J7.1/lib/s390/default/
10	Call	libj9vm27.so	n/a	gpCheckCallin	E+5C	
						From /apc/java710/31bit/usr/lpp/java/J7.1/lib/s390/default/
11	Call	n/a	n/a	gpProtectAndRun	n/a	From not determined
12	Call	libj9prt27.so	n/a	j9sig_protect_ceedlr	E+176	
						From /apc/java710/31bit/usr/lpp/java/J7.1/lib/s390/default/
13	Call	libj9vm27.so	n/a	signalProtectAndRunGlue	E+14	
						From /apc/java710/31bit/usr/lpp/java/J7.1/lib/s390/default/
14	Call	n/a	n/a	gpProtectedRunCallInMethod	n/a	From not determined
15	Call	n/a	n/a	RUNCALLINMETHOD	n/a	From not determined
16	<<< XPLink	CEEPLPKA	n/a	CEEVRONU	E+10CE	CEL Common Runtime; From LPA
17	Abend S0C4	*****	libHelloWorld.so	n/a	Java_HelloWorld_badArrayAccess	From /u/ctest/javatest-j2c/bin/
					E+112	

DTFJ 処理が完了すると、Java™ イベントが表示されます。

The following events are presented in chronological order.

Event #	Type	Fail Point	Module Name	Program Name	EP Name	Event Location (*)	Description
1	Call		JVMLDM71	CEER00TA	n/a	P+D4	BOOTSTRAP MODULE FOR LE; From CTEST.AUTHLOAD
2	Call		CEEPLPKA	n/a	CEEBBEXT	E+1D2	BOOTSTRAP MODULE FOR Language Environment; From LPA
3	Call		CELHV003	n/a	EDCZHINV	E+B4	CRTL Main invocation event XPLINK; From CEE.SCEERUN2
4	>>> XPLink		CEEPLPKA	n/a	CEEVROND	E+127E	Run on down stack swap; From LPA
5	Call		JVMLDM71	JzosVM#C	main	E+A6	From CTEST.AUTHLOAD
6	Java		n/a	n/a	com.ibm.j9ddr.vm27.view.dtfj.java.DTFJJavaStackFrame@300676d	L#1	From not determined
7	Call		n/a	n/a	RUNCALLINMETHOD	n/a	From not determined
8	<<< XPLink		CEEPLPKA	n/a	CEEVRONU	E+10CE	CEL Common Runtime; From LPA
9	Abend S0C4	*****	libHelloWorld.so				

```
n/a      Java_HelloWorld_badArrayAccess
                E+112      From /u/ctest/javatest-j2c/bin/
```

## Java レポートの制限の情報

Fault Analyzer ラッパー・ユーティリティにより作成された障害項目では、変数情報は表示されません。

これは Java™ 仮想マシンの例外処理の方法が原因です。

- JVM は例外のハンドラーの各スタック・フレーム (つまり、try-catch ブロック) を検索します。
- 指定したフレームにハンドラーがない場合、JVM はスタック・フレーム変数を破棄し、次のフレームを検索します。この検索および破棄は、JVM が例外のハンドラーを見つけるまで繰り返されます。

Fault Analyzer ラッパー・ユーティリティは、それ以外の場合の未処理例外に対するハンドラーとして動作します。これは、Fault Analyzer の起動時に、ユーザー・アプリケーション変数の値が存在しないことを示しています。

JVM の例外処理の方法の詳細は、「JVM の仕様」の `athrow` 命令に関するセクションを参照してください。

## Java アプリケーションのサンプル

IDI.SIDISAM1 サンプル・データ・セットで提供されている JCL サンプルは、Java™ および Fault Analyzer での言語間呼び出しの使用法を示しています。

このサンプルを実行するには、以下を行う必要があります。

- サイト固有の要件に応じてジョブ・カードを調整します。
- Fault Analyzer、Java™、および必要なコンパイラーのインストール場所を指定します。
- 300 MB 以上の領域サイズでサンプルを実行します。
- このサンプルは階層ファイル構造 (HFS) ファイルを作成するため、サンプルを実行するユーザー ID に OMVS セグメントがあることを確認してください。

このサンプルは、手で削除する必要があるいくつかの一時ファイルを作成します。

### サンプル 1 (IDISJAV1): バッチで Java を起動する Enterprise PL/I

サンプル IDI.SIDISAM1(IDISJAV1) の JCL は、Enterprise PL/I がバッチで Java™ を起動するときの Fault Analyzer の使用法を示しています。これには、PDSE データ・セット・メンバー内にある Enterprise PL/I DLL を呼び出す Java™ が含まれています。

SNAP ダンプ呼び出しの Fault Analyzer レポートで予想されるサンプル結果を以下に示します。

- イベント・サマリーには、クラスおよびメソッド情報を含む Java™ イベントのみが含まれます。
- Java™ 仮想マシン (JVM) は変数情報を取得するメカニズムを提供しないため、個々の Java™ イベントにはソース・コードは含まれますが、Java™ 変数情報は含まれません。

PL/I 異常終了の Fault Analyzer レポートで予想されるサンプル結果を以下に示します。

- 概要セクションには、PL/I プログラムからのソース・コード情報と、変数の宣言および値が含まれます。
- PL/I コードは TEST(SEP) オプションを使用してコンパイルされたため、PL/I ソース・コード情報は SYSDEBUG から取得されます。
- イベント・リストでは、PL/I イベントと Java™ イベントの両方が 1 つのビューに含まれています。Java™ では、クラス名とメソッド名は再分析中にのみ表示されます。(この情報はリアルタイム・レポートでは使用できません。)
- Java™ イベントでは、ソース・コード情報と変数値の両方が表示されます。Diagnostic Tool Framework for Java™ (DTFJ) API で変数名を使用できない場合は、代わりに変数アドレスが報告されます。

## サンプル 2 (IDISJAV2): バッチで Java を起動する Enterprise COBOL

サンプル IDI.SIDISAM1(IDISJAV2) の JCL は、Enterprise COBOL がバッチで Java を起動するときの Fault Analyzer の使用方法を示しています。これには、PDSE データ・セット・メンバー内にある Enterprise COBOL DLL を呼び出す Java が含まれています。

SNAP ダンプ呼び出しの Fault Analyzer レポートで予想されるサンプル結果を以下に示します。

- イベント・サマリーには、クラスおよびメソッド情報を含む Java イベントのみが含まれます。
- Java 仮想マシン (JVM) は変数情報を取得するメカニズムを提供しないため、個々の Java イベントにはソース・コードは含まれますが、Java 変数情報は含まれません。
- Java 情報セクションには、Java 仮想マシン (JVM) のバージョン、Java 環境変数、および JVM 初期化引数 (`initArgs`) などの情報が含まれます。

COBOL 異常終了の Fault Analyzer レポートで予想されるサンプル結果を以下に示します。

- 概要セクションには、COBOL ソース・コード情報と、関連する変数の宣言および値が表示されます。
- イベント・リストには、COBOL および Java イベントと、クラス名、メソッド名、および行番号が含まれます。
- COBOL イベントは、TEST(SOURCE) コンパイラー・オプションにより、DWARF ソース情報を使用します。

## サンプル 3 (IDISJAV3): バッチで Java を起動する 64 ビット Enterprise PL/I

サンプル IDI.SIDISAM1(IDISJAV3) の JCL は、64 ビット Enterprise PL/I がバッチで Java™ を起動するときの Fault Analyzer の使用方法を示しています。これには、PDSE データ・セット・メンバー内にある Enterprise PL/I DLL を呼び出す Java™ が含まれています。

Fault Analyzer レポートで予想されるサンプル結果は、[サンプル 1 \(IDISJAV1\): バッチで Java を起動する Enterprise PL/I ページ 271](#) の結果と基本的には同じです。

このサンプルでは、PL/I プログラムが NOTEST オプションを使用してコンパイルされているため、ソース・コード情報を表示するために LANGX サイド・ファイルが使用されます。

## サンプル 4 (IDISJAV4): IMS Java バッチ処理例

サンプル IDI.SIDISAM1(IDISJAV4) の JCL は、IMS™ Java™ バッチ処理 (JBP) 領域での Fault Analyzer の使用方法を示しています。このサンプルでは、IMS™ が Java™ プログラムを呼び出し、次にこのプログラムが COBOL およびアセンブラー・プログラムを呼び出します。



このサンプルでは、IMS™ インストール検査プログラム (IVP) のプログラム仕様ブロック (PSB) を使用しています。サンプルを実行する前に、PSB が開始されていることを確認してください。PSB が停止していると、予想される異常終了 U0123 の代わりに異常終了 0456 が発生します。

Fault Analyzer レポートで予想されるサンプル結果を以下に示します。

- このレポートには、COBOL プログラム、アセンブラー・プログラム、Java™、および IMS™ に関する情報が含まれます。
- 概要セクションには、COBOL ソースと変数の情報が表示されます。
- イベント・サマリーでは、IMS™、Java™、COBOL、およびアセンブラーの呼び出しが 1 つの概要に表示されます。
- アセンブラー・イベントでは、ソース・コード情報はアセンブラー ADATA コンパイラー・オプションから取得されます。

## サンプル 5 (IDISJAV5): バッチで Java を起動する C++ プログラム

サンプル IDI.SIDISAM1(IDISJAV5) の JCL は、31 ビットまたは 64 ビット C++ プログラムを使用してバッチで Java™ を起動するときの Fault Analyzer の使用方法を示しています。これには、PDSE データ・セット・メンバー内にある C++ DLL を呼び出す Java™ が含まれています。

31 ビットおよび 64 ビットの Snap.dump() では、Fault Analyzer レポートは Java™ 情報のみを表示します。Java™ 仮想マシン (JVM) は変数情報を取得するメカニズムを提供しないため、どちらもソース・コードは含まれますが、Java™ 変数情報は含まれません。

31 ビットおよび 64 ビット C++ 異常終了の Fault Analyzer レポートで予想されるサンプル結果を以下に示します。

- 概要セクションには、異常終了が発生した C++ ソース・コードの行が含まれます。
- イベント・サマリーでは、Java™ と C++ 両方の呼び出しが 1 つのビューに含まれています。
- C++ イベントでは、NODEBUG コンパイラー・オプションにより、ソース・コード情報にコンパイラー・リストが使用されます。
- Java™ イベントでは、ソース・コード情報がスタック・フレーム参照情報とともに表示されます。

## サンプル 6 (IDISJAV6): Fault Analyzer ラッパー・ユーティリティで Java バッチ・アプリケーションを呼び出し

サンプル IDI.SIDISAM1(IDISJAV6) の JCL は、try-catch ブロック内の main() メソッドで Java™ クラスを呼び出すための Fault Analyzer Java™ ラッパー・ユーティリティの使用法を示しています。未処理の例外が自動的にキャッチされ、Snap.dump() メソッドを使用して障害項目が作成されます。

ラッパー・ユーティリティを使用して未処理の Java™ 例外をキャッチする場合、Fault Analyzer が使用できるのは限られた Java™ 情報のみです。

- レポートの概要には、例外タイプとメッセージに関する詳細が含まれます。
- イベント・リストには、クラス名やメソッド名を含む、アクティブ・スレッドの Java™ イベントが含まれます。
- 個々の Java™ イベントでは、クラス・メソッド変数情報とスタック・フレーム参照変数情報は、Fault Analyzer で使用できないため表示されません。

詳しくは、[Java レポートの制限の情報 ページ 271](#)を参照してください。

ラッパー・ユーティリティの呼び出しの詳細については、[Fault Analyzer ラッパーの起動 ページ 51](#)を参照してください。

## Fault Analyzer ラッパー・ユーティリティからの出力例

ラッパー・ユーティリティは、呼び出されているクラスと指定されたコマンド・ライン引数を報告します。例:

```
Fault Analyzer Java Wrapper
Invoking class: com.example.JavaUnhandled
with args: '[pureJavaUnhandledException]'
When FA catches an unhandled exception, the FA wrapper displays the following
messages:
Fault Analyzer was invoked to handle a Java Throwable:
.
-----
java.lang.ClassCastException: Cannot cast class java.util.concurrent.atomic.AtomicLong to
class java.lang.Class at java.lang.Class.cast(Class.java:2614)
at com.example.JavaUnhandled.computeTheAnswer(JavaUnhandled.java:20)
at com.example.JavaUnhandled.pureJavaUnhandledException(JavaUnhandled.java:30)
at com.example.JavaUnhandled.main(JavaUnhandled.java:41)
-----
Calling Snap.dump to create a fault entry:
DA.DCAT(F39992)
FA: Passing the exception to JVM...
...
```

## 第9章. Fault Analyzer レポート

この章では、Fault Analyzer によって作成されるレポートの内容について説明します。

### 汎用レポート情報

バッチ再分析により作成される分析レポートは、リアルタイム・レポートと同様の構造を持ちます。対話式の再分析によるレポートも同じ構造ですが、レポート全体を順番に表示する必要はなく、特定のセクションを選択して表示できます。さらに、対話式レポートは、CICS® システム異常終了または Java™ 分析に関連する、リアルタイムまたはバッチの分析レポートに含まれない情報を提供します。詳しくは、[CICS システム異常終了ダンプ分析の実行 ページ 251](#)を参照してください。

Fault Analyzer レポートは、リアルタイム分析のために DDname IDIREPRT に書き込まれ、バッチ再分析のために SYSPRINT に書き込まれます。

### 最上位の異常終了コード

障害分析の結果、複数の異常終了コードが提示される場合、Fault Analyzer は常に、最初の異常終了コード (日時順) を最上位と見なします。ただし、Language Environment® CEEWUCHA ユーザー条件ハンドラーが使用される (LE ランタイム・オプション USRHDLR(CEEWUCHA) を使用して登録される) 場合、通常、最後の異常終了コードが、エラーについて最も詳細な情報を提供するコードです。

### オープン・ファイルのレコード情報

オープンしている順次ファイルのレコードは、日時順に表示されます。

入力ファイルの場合、障害の前にアプリケーション用に提供されたレコードである現行レコードとともに、「直前・現行・次」のシーケンスが使用されます。

出力ファイルの場合、「最後に書き込まれたレコード」および「現行レコードのビルド域」の説明が使用されます。

### スパン・レコードの解釈

可変のスパン・レコードが存在する可能性がある、オープン・ファイルのレコード情報を表示するときに、スパン・レコードが含まれている場合、ユーザーは、レコードの論理ビューを再アセンブルする必要があります。これは、レコード情報の基になるバッファーに、特定のスパン・レコードのセグメントの一部が入っていない可能性があるためです。スパン・レコードが含まれている場合、レコードごとのレコード情報見出しには、使用可能なデータおよび特定のスパン・レコードの構造に従って、「最初のセグメント」、「中間のセグメント」、および「最後のセグメント」が含まれます (中間のセグメントが存在しなかったり、あるいは多くの中間のセグメントが存在する場合があります)。

オープン・ファイル情報には、次の2つのリアルタイム・レポート抽出があります。

[図 183: オープン入力ファイル情報のリアルタイム・レポート抽出の例 \(パート 1/2\) ページ 276](#)は、入力ファイルのファイル情報とバッファーを示しています。

図 183. オープン入力ファイル情報のリアルタイム・レポート抽出の例 (パート 1/2)

```

Open Files

File Name . . . . . : INDD
Data Set Name . . . . . : LJBERRY.OUT80S
File Attributes . . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=SEQUENTIAL,
                           RECFM=VARIABLE BLOCKED SPANNED
Last I/O Function . . . . . : READ
Open Status . . . . . : INPUT
File Status Code. . . . . : 0

Previous Record -2. . . . . : Segment data length 6, variable record first segment
Address  Offset      Hex                               EBCDIC
-----  -
08053F32          C6C6C6F6 F6F6          *FFF666          *

Previous Record -1. . . . . : Segment data length 32, variable record intermediate segment
Address  Offset      Hex                               EBCDIC
-----  -
08053F40          40404040 40404040 40404040 40404040 *          *
Line 08053F50 same as above
    
```

図 184. オープン入力ファイル情報のリアルタイム・レポート抽出の例 (パート 2/2)

```

Previous Record . . . . . : Segment data length 2, variable record last segment
Address  Offset      Hex                               EBCDIC
-----  -
08053F68          4040          *          *

Current Record. . . . . : Record data length 20, variable record
Address  Offset      Hex                               EBCDIC
-----  -
08053F6E          C7C7C7F7 F7F74040 40404040 40404040 *GGG777          *
08053F7E          +10 40404040          *          *

Next Record . . . . . : Segment data length 2, variable record first segment
Address  Offset      Hex                               EBCDIC
-----  -
08053F86          C8C8          *HH          *

Next Record +1. . . . . : Segment data length 8, variable record last segment
Address  Offset      Hex                               EBCDIC
-----  -
08053F90          C8F8F8F8 40404040          *H888          *
NOTE: Some segments not available due to buffer wrap-around.
    
```

図 185: オープン出力ファイル情報のリアルタイム・レポート抽出の例 ページ 277 は、出力ファイルのファイル情報とバッファーを示しています。

図 185. オープン出力ファイル情報のリアルタイム・レポート抽出の例

```

Open Files

File Name . . . . . : OUTDD
Data Set Name . . . . . : LJBERRY.OUT80S
File Attributes . . . . . : ORGANIZATION=SEQUENTIAL, ACCESS MODE=SEQUENTIAL,
                           RECFM=VARIABLE BLOCKED SPANNED
Last I/O Function . . . . . : WRITE
Open Status . . . . . : OUTPUT
File Status Code. . . . . : 0

Last Record Written -2. . : Segment data length 20, variable record first segment
Address  Offset      Hex                               EBCDIC
-----  -
08053F74          C9C9C9F9 F9F94040 40404040 40404040 *III999      *
08053F84      +10  40404040                               *           *

Last Record Written -1. . : Segment data length 32, variable record intermediate segment
Address  Offset      Hex                               EBCDIC
-----  -
08053F90          40404040 40404040 40404040 40404040 *           *
      Line 08053FA0 same as above

Last Record Written . . . : Segment data length 28, variable record last segment
Address  Offset      Hex                               EBCDIC
-----  -
08053FB8          40404040 40404040 40404040 40404040 *           *
08053FC8      +10  40404040 40404040 40404040          *           *

Current Record Build Area : RDW is zero, no length assigned yet
Address  Offset      Hex                               EBCDIC
-----  -
08053E90          D1D1D1C1 C1C14040 40404040 40404040 *JJJAAA      *
08053EA0      +10  40404040 40404040 40404040 40404040 *           *
      Lines 08053EB0-08053EC0 same as above
08053ED0      +40  40404040 40404040 40404040 40406E6E *           >>*
08053EE0      +50  00000000                               *....      *

```

## COBOL で抑止されるコピーブック

大きいコピーブックが COBOL プログラムに含まれている場合、コンパイル・リストでコピーブックが展開されないようにするため、SUPPRESS オプションがよく使用されます。例:

```
COPY copy-book-name SUPPRESS
```

Fault Analyzer バッチ・レポートの作業用ストレージ・マップに、抑止されたコピーブックを (リアルタイム分析か再分析かにかかわらず) 組み込むには Detail(Long) オプションを指定する必要があります。

抑止されたコピーブック情報は、対話式再分析レポートで常に選択可能です。



**注:** 抑止されたコピーブック情報は、コンパイラー・リストから抑止された情報の再ビルドにあたっての「最良の試行」であるため、必ずしも完全でなかったり、全体的に正確でなかったりします。

## 主なレポート・セクション

主なレポート・セクションのそれぞれについて、以下に説明します。

### Prolog セクション

Prolog セクションは、レポートの先頭から Synopsis セクションの開始に至るまでのすべてにより構成されます。

Prolog セクションの先頭には、インストール済みの最新の保守とともに、Fault Analyzer のバージョン、リリース、修正レベルに関する情報が示されます。この情報に続き、Fault Analyzer の著作権文およびレポートの見出しが表示されます。

ダンプ・データ・セット、またはダンプ・データ・セットの対話式再分析で作成された障害項目のバッチ再分析を実行して、そのダンプ・データ・セットまたは障害項目に CICS® システムまたは、Java™ 環境の情報が含まれている場合は、注がここに追加されます。この注は、現行障害を分析するためのより良い方法が、適切な ISPF インターフェース再分析方式を使用することであると通知するものです。この方法が有効なのは、これらのタイプの環境用の Fault Analyzer には固有のサポートが対話式再分析レポートに用意されており、ユーザーはバッチ・レポートには含まれていない情報を参照できるためです。

### Synopsis セクション

Synopsis セクションでは、障害とその分析の要旨が示されます。次の見出しが、前に付けられます。

```
<H1> S Y N O P S I S
```

### Summary セクション

Summary セクションでは、すべてのイベント (異常終了や呼び出し項目など) のリストが日時順に示されます。次の見出しが、前に付けられます。

```
<H1> E V E N T   S U M M A R Y
```

リストの各イベント項目について、モジュール名、プログラム名、失敗する行、またはステートメント番号など、主要な情報も示されます。これらの情報は、イベントの Event Details セクションに示される情報の要約です。

リストの項目ごとに、次の内容が提供されます。

#### Event #

このセクションは、イベントに割り当てられた、発生順に昇順の固有のシーケンス番号です。イベント番号は、レポートの Event Details セクションでイベントの詳細情報を参照する場合に使用します。

#### イベント・タイプ

次のいずれかのイベント・タイプです。

**Abend *abend-code***

異常終了イベントは、システムのハードウェアまたはソフトウェアがプログラムを終了したときに発生します。これは、無効なアクションを実行しようとしたことや、何らかの理由で有効なアクションを実行できなかったことが原因です。オペレーティング・システムまたはその他のソフトウェアによって生成されたシステムまたはユーザーの異常終了コードは、異常終了に関連付けられます。

異常終了はソフトウェアによって「トラップ」され、別のタイプの異常終了に変換される可能性があるため、複数の異常終了を障害に関連付けることができます。

**AMODE64->31**

AMODE 64 から AMODE 31 への言語環境プログラムの切り替えを表すイベントです。

**BALR**

BALR イベントは、分岐およびリンク・メカニズムを介して 1 つのプログラムから別のプログラムに制御が移され、最初のプログラムのステータスに関する情報が 2 番目のプログラムによって標準の保存領域に保存されない場合に発生します。

**Branch**

分岐イベントは、Breaking Event Address Register (BEAR) ハードウェア機能がワイルド分岐を引き起こした命令に関する関連情報を提供するときに発生します。

**Call**

呼び出しイベントは、分岐およびリンク・メカニズムを介して 1 つのプログラムから別のプログラムに制御が移され、最初のプログラムのステータスに関する情報が 2 番目のプログラムによって標準の保存領域に保存された場合に発生します。リンク・イベントを使用して、呼び出しイベントはその障害の実行証跡を表示します。

Fault Analyzer は、異常終了時に取得された保管域情報から呼び出しイベントを再構成します。障害の各異常終了イベントに関連付けられる呼び出しイベントは、ない場合も、1 つ以上ある場合もあります。

**Current PRB**

現在のプログラム要求ブロック (PRB) イベントです。

**Debug Tool**

z/OS® Debugger イベントです。

**EXEC CICS**

CICS EXEC CICS ステートメント・イベントです。

**EXEC DLI**

CICS EXEC DLI ステートメント・イベントです。

## **EXEC SQL**

CICS EXEC SQL ステートメント・イベントです。

### **Execute**

実行イベントは、異常終了イベントの前に発生した EX マシン・インストラクションの実行を要約します。

## **IDISNAP**

IDISNAP イベント。

### **Interrupt**

中断要求ブロック (IRB) イベントです。

### **Java™**

Java™ イベントです。通常、Java メソッドの呼び出しです。

## **LE Condition**

Language Environment® が処理できない条件イベントです。

## **Link**

リンク・イベントは、スーパーバイザー呼び出しメカニズムを介して1つのプログラムから別のプログラムに制御が移され、最初のプログラムのステータスに関する情報が2番目のプログラムによって標準の保存領域に保存された場合に発生します。呼び出しイベントを使用して、リンク・イベントはその障害の実行証跡を表示します。

Fault Analyzer は、異常終了時に取得された保管域情報からリンク・イベントを再構成します。障害の各異常終了イベントに関連付けられるリンク・イベントは、ない場合も、1つ以上ある場合もあります。

## **ONCODE code**

PL/I 2.3 非 LE ランタイム ONCODE イベントです。ONCODE 条件が表示されます。

### **PC number**

PC イベントは、PC (プログラム呼び出し) 命令が実行されたことが原因で、制御がシステム・サービスに移されるときに発生します。

### **SVC number**

SVC イベントは、SVC 命令 (リンク以外) が実行されたことが原因で、制御がアプリケーション・プログラムからシステム・サービスに移されるときに発生します。

アプリケーション・プログラムによって呼び出されたシステム・サービス・ルーチンで異常終了が発生すると、SVC イベントが分析に表示されます。障害の各異常終了イベントに関連付けられる SVC イベントは、最大で1つです。



**Fail Point**

イベントが障害点として識別された場合に、そのことがこの列の \*\*\*\*\* によって示されます。

**Module Name**

可能な場合、イベントに関連するロード・モジュールの名前です。それ以外の場合は n/a です。

**Program Name**

可能な場合、イベントに関連するプログラムまたは CSECT の名前です。それ以外の場合は n/a です。

**EP Name**

可能な場合、イベントに関連するエン트리・ポイントの名前です。それ以外の場合は n/a です。

**Event Location**

このフィールドには、次の省略語のうち 1 つ以上が表示される場合があります。

**F#n**

ソース・ファイル番号

**L#n**

ソース・ファイルの行番号

**S#n**

リスト・ファイルのステートメント番号

**M+x**

ロード・モジュールの先頭からのオフセット

**P+x**

プログラムの先頭からのオフセット

**E+x**

エン트리・ポイントの先頭からのオフセット

**説明**

次の情報が提供されます。

- 使用可能であれば、プログラムまたはロード・モジュールの機能の要旨。
- モジュールがロードされたデータ・セットの名前、もしくは、モジュールが LPA 内で検索された場合 LPA、もしくは、モジュールが CSA 内で検索された場合 CSA です。

モジュールが LPA 内にも CSA 内にもなく、データ・セット名が決定できなかった場合、Not determined と表示されます。

## 最大呼び出し深さ

イベント・サマリーに組み込むことができるイベントの最大数は 200 です。障害に 200 を超えるイベントが含まれている場合は、最初の 100 イベントと最後の 100 イベントのみが表示され、最大呼び出し深さを超えたためにイベントが抑止されたことが示されます。

## Event Details セクション

Event Details セクションでは、各イベントの詳細情報が示されます。次の見出しが、前に付けられます。

```
<H1> E V E N T   D E T A I L S
```

Details セクションに含まれるイベントのタイプは、有効な Detail オプションに従います。また、メッセージの説明 (マニュアルを参照する必要がないように Fault Analyzer によって抽出されたもの) や、プログラムの作業用ストレージの内容など、イベントに関連する追加情報も詳細イベント・セクションに組み込まれます。該当する場合は、異常終了コードの説明やオープン・ファイル・バッファなどの情報も示されます。

レポートの詳細セクションにあるソース・コード情報には、イベント・ソース位置の前後のソース行またはステートメントが最大 5 行まで表示されます。イベント・ソース位置が、展開されたアセンブラー・マクロ内にある場合は、マクロの展開によって引き起こされるステートメントのほかに追加のソース・ステートメントがあります。

デフォルトの追加の 5 行/ステートメントを変更するには、Detail オプション ([Detail ページ 586](#)を参照) または分析制御ユーザー出口 ([分析制御ユーザー出口 ページ 475](#)を参照) を使用します。

障害点イベントに対して、一致するコンパイラー・リストまたはサイド・ファイルが提供されない場合は、障害が起こったマシン命令が表示されます。さらに、障害が起こった命令の前にある 12 個の命令、および後にある 6 個の命令も表示されます。次の例のリストは、IDIVPCOB COBOL インストール検証プログラム (IVP) に基づきますが、コンパイラー・リストまたはサイド・ファイルは提供されていません。

```
<H2> EVENT 1 OF 1: ABEND S0C7

*****
***** P O I N T   O F   F A I L U R E *****
*****

Abend Code. . . . . : S0C7
Program-Interruption Code . : 0007 (Data Exception)
                          A decimal digit or sign was invalid.

Most recently referenced data items:
Data Item . . . . . : BLW=00000+018
  At Address. . . . . : 167900A0
  Length. . . . . : X'4'
  Data Item Storage . . . : 0986888F  *.fh.*

Data Item . . . . . : BLW=00000+020
  At Address. . . . . : 167900A8
  Length. . . . . : X'4'
  Data Item Storage . . . : C1C2C3C4  *ABCD*

NOTE: Source code information could not be presented because the search for a
      compiler listing or side-file was unsuccessful for program IDISCB1.
```

```

Load Module Name. . . . . : SYS05291.T124505.RA000.IDIVPCOB.GOSET.H02(IDISCBL1)
  At Address. . . . . : 16700D88
  Load Module Length. . . . : X'1278'
  Link-Edit Date and Time . . : 2019/10/18 12:45:07

```

```

Program and Entry Point Name: IDISCBL1
  At Address. . . . . : 16700D88 (Module IDISCBL1 offset X'0')
  Program Length. . . . . : X'638'
  Program Language. . . . . : COBOL (Compiled using COBOL for OS/390 & VM V2 R2
                               M2 on 2019/10/18 at 12:45:06)

```

```

Machine Instruction . . . . : FD73D0B8D0A8 DP 184(8,R13),168(4,R13)
  At Address. . . . . : 1670115C (Program IDISCBL1 offset X'3D4')
  AMODE . . . . . : 31
  Failing Operand . . . . . : Second operand
  First Operand Address . . : 0002A0D0 (171824 bytes of storage addressable)
  First Operand Length. . . : 8
  First Operand Storage . . : 00000000 0986888C *.....fh.*
  Second Operand Address. . : 0002A0C0 (171840 bytes of storage addressable)
  Second Operand Length . . : 4
  Second Operand Storage. . : C1C2C3CF *ABC.*

```

Instructions around point of failure:

Offset	Hex	Instruction
-36	D203 D094 D098	MVC 148(4,R13),152(R13)
-30	5820 905C	L R2,92(,R9)
-2C	58F0 202C	L R15,44(,R2)
-28	4110 A0E9	LA R1,233(,R10)
-24	05EF	BALR R14,R15
-22	58B0 C014	L R11,20(,R12)
-1E	47F0 B1CC	BC 15,460(,R11)
-1A	D203 D0A8 8020	MVC 168(4,R13),32(R8) BLW=00000+020
-14	960F D0AB	OI 171(R13),15
-10	D203 D0B0 8018	MVC 176(4,R13),24(R8) BLW=00000+018
-A	960F D0B3	OI 179(R13),15
-6	F873 D0B8 D0B0	ZAP 184(8,R13),176(4,R13)
*****	FD73 D0B8 D0A8	DP 184(8,R13),168(4,R13)
+6	D201 8028 D0BA	MVC 40(2,R8),186(R13) BLW=00000+028
+C	940F 8028	NI 40(R8),15 BLW=00000+028
+10	960F 8029	OI 41(R8),15 BLW=00000+029
+14	5830 D094	L R3,148(,R13)
+18	07F3	BCR 15,R3
+1A	9120 9054	TM 84(R9),32

Program Status Word (PSW) . : 078D2000 96701162

General Purpose Registers:

```

R0: 0002A0D8 (171816 bytes of storage addressable)
R1: 16700F91 (Module IDISCBL1 program IDISCBL1 + X'209')
R2: 0001B7FC (231428 bytes of storage addressable)
R3: 16701126 (Module IDISCBL1 program IDISCBL1 + X'39E')
R4: 16700DC0 (Module IDISCBL1 program IDISCBL1 + X'38')
R5: 00016AF0 (251152 bytes of storage addressable)
R6: 00000000 (2048 bytes of storage addressable)
R7: 00000000 (2048 bytes of storage addressable)

```

```

R8: 16790088 (Module IDISCBL1 program IDISCBL1 WORKING-STORAGE SECTION
          BLW=00000 + X'0')
R9: 1678C100 (253696 bytes of storage addressable)
R10: 16700E9C (Module IDISCBL1 program IDISCBL1 + X'114')
R11: 16700FBC (Module IDISCBL1 program IDISCBL1 + X'234')
R12: 16700E84 (Module IDISCBL1 program IDISCBL1 + X'FC')
R13: 0002A018 (172008 bytes of storage addressable)
R14: 967010D2 (Module IDISCBL1 program IDISCBL1 + X'34A')
R15: 96759E60 (Module IGZCPAC + X'3C348')

```

## System-Wide Information セクション

このセクションには、例えば、特定のイベントに属しているものとして認識されないコンソール・メッセージや、トレース・データ、3270 画面バッファ内容などの CICS® システム関連情報が含まれます。次の見出しが、前に付けられます。

```
<H1> S Y S T E M - W I D E   I N F O R M A T I O N
```

特定のイベントに関連付けることができないオープン・ファイルに関する情報も、ここに示されます。このセクションに情報がない場合、この情報はレポートに含まれません。

## Abend Job Information セクション

このセクションは、Fault Analyzer のリアルタイム起動に関連する異常終了ジョブ、または再分析の場合、分析されたミニダンプや MVS™ ダンプに関する情報を提供します。次の見出しが、前に付けられます。

```
<H1> A B E N D   J O B   I N F O R M A T I O N
```

## Options セクション

このセクションは、分析時に有効となっていた Fault Analyzer オプションのリストです。次の見出しが、前に付けられます。

```
<H1> O P T I O N S
```

## Epilog セクション

Epilog セクションは、Options セクションの末尾からレポートの終わりに至るまでのすべてにより構成されます。

リアルタイム分析レポートでは、使用される呼び出し出口および分析中に割り振られた標準値以上のストレージの概算量に関する情報に続き、割り当てられた障害 ID が示されます。一方、バッチ再分析レポートでは、使用される障害 ID およびヒストリー・ファイルに関する情報が示されます。

レポートの最終行には、レポートの作成日時が示されます。

## レポート例

Fault Analyzer には、インストール検証プログラム (IVP) および IVP によって生成されたレポート例が含まれています。IVP は IDI.SIDISAM1 データ・セットのメンバーで、レポート例は IDI.SIDIDOC1 データ・セットのメンバーです。IVP およびレポート例のメンバー名については、以下の表を参照してください。

表 7. IVP およびレポート例のメンバー名

IVP	以下の IVP メンバー名: IDI.SIDISAM1	以下のレポート例メンバー名: IDI.SIDIDOC1
COBOL	IDIVPCOB	IDISRP01
PL/I	IDIVPPLI	IDISRP02
アセンブラー	IDIVPASM	IDISRP03
C DB2®	IDIVPDB2	IDISRP04
MQSeries®	この MQSeries® レポートは、MQSeries® 異常終了用に作成された障害項目の再分析を実行して作成されました。	IDISRP05
C	IDIVPC	IDISRP06

## 第 10 章. Fault Analyzer ヒストリー・ファイルにアクセスするための非 ISPF インターフェースの使用

TSO/ISPF の代わりに Eclipse プラグインを介して Fault Analyzer にアクセスできます。

### IBM Fault Analyzer プラグイン (Eclipse 用)

Fault Analyzer は IBM® Fault Analyzer プラグイン (Eclipse 用) をサポートしています。このプラグインはワークステーションをベースとしていて、FA ISPF インターフェースの代わりに使用できます。IBM Explorer for z/OS® または IBM Developer for z/OS Eclipse をベースとしたプラットフォームで使用されるこのプラグインは、FA 障害ヒストリー・ファイルおよびビューで使用できるグラフィカル・インターフェースを備えています。このプラグインでは、以下のことが可能です：

- 複数のシステムから複数の障害ヒストリー・ファイルおよびビューを処理する。
- 異常終了するプログラムのリアルタイム分析中に作成された障害項目を表示する。
- 障害項目に関連したダンプ・ストレージを表示する。
- 一致するサイド・ファイルが見つかったプログラム・ソースを表示する。

プラグインに関する情報は、プラグイン・インターフェースの「ヘルプ」メニューから参照できます。

### 以下の環境での対話式再分析の実行 CICS®

Fault Analyzer は、特殊なコンポーネントを使用して ISPF パネルを表示します。このパネルによって、ヒストリー・ファイルを表示し、対話式再分析を実行できる、CICS® トランザクションとして作動することが可能になります。CICS® 環境におけるこの機能は、TSO を使用しません。この機能は、MVS™ イメージ上で TSO ログオン機能を持たず、しかしその MVS™ イメージ上のヒストリー・ファイル情報を表示したり、分析したりする必要があるユーザーを対象としています。

CICS® トランザクションとして稼働する Fault Analyzer の機能は、([Fault Analyzer ISPF インターフェース ページ 62](#)で説明している) TSO/ISPF 環境における Fault Analyzer の機能にほぼ等しいですが、以下のような制約事項と違いがあります。

1. 障害項目のバッチ再分析はサポートされません。
2. ISPF EDIT を起動する機能はサポートされません。
  - 対話式再分析の前のオプション・データ・セットの編集。
  - 対話式再分析の前の割り振りデータ・セットの変更。
  - 使用可能な EXEC リストからのユーザー・フォーマット設定 EXEC の編集。
  - DSECT リスト画面からの DSECT の編集。
3. このコンポーネントは TSO 配下で実行されるわけではないので、データ・セット名のプレフィックス変換は実行されません。すなわち、データ・セット名を入力する場合は (例えば、Fault History File または View など)、引用符の有無にかかわらず、データ・セット名を完全修飾する必要があります。
4. Interactive Reanalysis Report の中で行われる ISPF プロファイルの変更、例えば、コマンド行のロケーションの変更などは、Fault Entry List 画面に戻っても、すぐには反映されていない場合があります。しかし、プロファイル変更は、メイン CICS® トランザクションの次の起動時には検出されます。

このインターフェースのインストールについては、[CICS 環境での対話式再分析の使用可能化 ページ 564](#)を参照してください。

## 第 II 部. Fault Analyzer インストールおよび管理

## 第 11 章. 障害分析の前のバージョンからのマイグレーション

Fault Analyzer の前のバージョンからのアップグレードに関する情報が、次のトピックで提供されます。

SMP/E インストールとそれに続く Fault Analyzer の新バージョンのアクティベーションについては、[Fault Analyzer の保守ページ 459](#)を参照してください。

次の情報は、すべてのバージョンの Fault Analyzer に適用されます。

### LPA モジュールの互換性

Fault Analyzer は、IDI.SIDIALPA データ・セット内のロード・モジュール間で下位互換性を提供します。したがって、以降のバージョンの Fault Analyzer をインストールし、CLPA で IPL を実行し、これらの LPA モジュールを LNKLST または STEPLIB において以前のバージョンの Fault Analyzer とともに使用することができます。

Fault Analyzer データ・セットが APF 許可を保持するには、STEPLIB 連結内のすべてのデータ・セットが APF 許可されている必要があります。

### シスプレックス全体でのヒストリー・ファイルの共用

複数のサポート対象レベルの Fault Analyzer が混在してインストールされているシスプレックスにおいて、ヒストリー・ファイルの共用に問題はありませぬ。

### V14.1 から V15.1 へのマイグレーション

このセクションでは、Fault Analyzer をバージョン 14.1 からマイグレーションする場合に認識しておくべきバージョン 15.1 での変更点について説明します。

- IDIXDCAP 事前ダンプ出口のインストールは、IEAVTABX CSECT 出口リストを介してサポートされなくなりました。IDIXDCAP は、IEAVTABX\_EXIT 動的出口としてインストールする必要があります。詳しくは、[MVS 変更オプション/抑止ダンプ出口のインストール IDIXDCAP ページ 334](#)を参照してください。
- XDUMP はデフォルトで有効になっています。詳しくは、[「拡張ミニダンプ・データ・セット \(XDUMP\) ページ 59」](#)を参照してください。
- 以前の非推奨のオプションが削除されました。
  - HistCols
  - InteractiveExitPromptSeconds
  - RDZClient
  - UseIDISTime
- IDIOPT1 SMP/E USERMOD (非標準 LE パラメーター・リスト分離文字の使用) は非推奨になりました。代替は現在ありません。
- IDIOPT2 SMP/E USERMOD (前の出口が RC=8 の場合に IDIXDCAP リアルタイム分析を抑止) は非推奨になりました。代わりに、IDIOPTLM 構成オプション・モジュール PREVEXIT オプション・フィールドを使用します。詳しくは、[前の出口が RC=8 の場合に IDIXDCAP リアルタイム分析を抑止 \(PREVEXIT\) ページ 343](#)を参照してください。



## V13.1 から V14.1 へのマイグレーション

このセクションでは、Fault Analyzer をバージョン 13.1 からマイグレーションする場合に認識しておく必要がある、バージョン 14.1 での変更点に関する情報を説明します。

- IDIOPTLM 構成オプション・モジュール内の XDUMPDSN オプションに、拡張ミニダンプ・データ・セットの適切な命名パターンを指定します。詳しくは、[デフォルトの拡張ミニダンプ \(XDUMP\) データ・セット名パターン \(XDUMPDSN\) の変更 ページ 342](#)を参照してください。
- 大容量の仮想ストレージにアクセスするために IDIXUFMT ロード・モジュール・フォーマット・ユーザー出口の IDIXDLOC 関数を使用している場合は、使用可能なストレージをすべて使い果たし、分析が異常終了する危険性があります。これを防ぐために、IDIXDLOC の代わりに IDIXXLOC を使用するようすべて変更します。IDIXXLOC 関数の詳細については、[IDIXXLOC – 独自のバッファリングを使用したダンプ・ストレージの検索 ページ 561](#)を参照してください。

## V12.1 から V13.1 へのマイグレーション

このセクションでは、Fault Analyzer をバージョン 12.1 からマイグレーションする場合に認識しておく必要がある、バージョン 13.1 での変更点に関する情報を説明します。

- 以下の Fault Analyzer SMP/E ++USERMOD は使用できなくなりました。
  - IDILEDS
  - IDISCNF
  - IDISRFR
  - IDISXCUM

これらは、IDIOPTLM 構成オプション・モジュールにおける同等の設定で置き換えられています。詳しくは、[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#)を参照してください。

- Fault Analyzer IDIOPTLM 構成オプション・モジュールを現在使用して IDICNFxx の代替データ・セット名を指定している場合は、ADFz Common Components IPVOPTLM 構成オプション・モジュールも必要になる場合があります。そうしないと、セキュリティー・サーバーのアクセス違反が発生する可能性があります。詳しくは、「[代替 parmlib データ・セットを IDICNFxx\(CNFDSN\) に指定 ページ 340](#)」を参照してください。
- Fault Analyzer ロケール・オプション *locale-name* を、ADFz Common Components IPVCNF00 parmlib メンバーに代替として指定できるようになりました。詳しくは、「[Locale ページ 609](#)」を参照してください。
- HistCols および InteractiveExitPromptSeconds オプションは推奨されなくなり、FAISPFopts オプションで置き換えられています。詳しくは、[FAISPFopts ページ 598](#)を参照してください。
- MVS™ ポスト・ダンプ出口 IDIXTSEL は、オプションのインストール項目ではなくなりました。これは、Java™ 障害キャプチャーなどの追加の Fault Analyzer 機能をサポートするために必要です。詳しくは、[Fault Analyzer のインストールおよびカスタマイズのチェックリスト ページ 292](#)を参照してください。
- Java™ ダンプ・キャプチャーには、IDI\_SDUMP\_ACCESS XFACILIT プロファイルへのアクセスが必要になりました。詳しくは、[Java try-catch ブロックからの Fault Analyzer の起動 ページ 47](#)を参照してください。
- CICS® オープン TCB ユーザーにとって、Fault Analyzer IDIPLT プログラムを CICS® 開始 PLT に組み込むことが必要条件になりました。詳しくは、[必要なプログラムの始動 PLT への追加 ページ 402](#)を参照してください。

- UseIDISTime は推奨オプションではなくなり、常に有効になりました。詳しくは、UseIDISTimeを参照してください。
- Fault Analyzer plug-in for Eclipse は、データ・セット IDI.SIDIDOC2 のメンバー IDIGUIP として提供されなくなりました。ただし、[IBM Fault Analyzer プラグイン \(Eclipse 用\) のインストール ページ 564](#) で説明されているように、このプラグインは Web 上で入手できます。
- IDIXFIT ユーザー出口パラメーター・リストに行われた変更は以下のとおりです。
  - 6 番目のパラメーターは以前、出口が呼び出される対象の障害項目へのポインターでした。現在では、6 番目のパラメーターは HD セグメント・データ域のアドレスであり、これには障害項目 ID だけでなく、障害項目に関連するさまざまな他の情報も含まれています。
  - 2 番目と 3 番目のパラメーターとして指定されるセキュリティー・サーバーのユーザー ID とデフォルト・グループ ID は、以前は障害項目自体から取得されることがありましたが、現在では常に出口の呼び出し元から取得されます。現在 6 番目のパラメーターが指している HD セグメント・データ域には、(使用可能であれば) 障害項目作成者のセキュリティー・サーバーのユーザー ID とデフォルト・グループ ID の情報が含まれていることに注意してください。

詳しくは、[IDIXFIT ユーザー出口の使用 ページ 369](#)を参照してください。

- ENV.VERSION の現行値が 0005 に変更されました。これは、ENV.FORMATting\_EXIT が、CICS®トランザクション障害の場合は常に「C」を指定していましたが、現在では、Fault Analyzer が XPCABND 出口によって起動された場合は「C」を、XDUREQ 出口によって起動された場合は「D」を指定するように変更されたことによります。

## V11.1 から V12.1 へのマイグレーション

このセクションでは、Fault Analyzer をバージョン 11.1 からマイグレーションする場合に認識しておく必要がある、バージョン 12.1 での変更点に関する情報を説明します。

- ADFz Common Components コンパイラー・リストまたは SYSADATA ファイルを使用するときに Fault Analyzer がソース・レベル・サポートを提供するようにするには、以下がインストールされていなければなりません:
- As BookManager® ソフトコピー・ブックは Fault Analyzer とともに出荷されなくなり、**Help > Fault Analyzer User's Guide and Reference** アクション・バー・オプションが削除されました。
- 以下のユーザー出口データ域フィールドが削除されました。
  - EPC.MINIDUMP\_PAGES (ENV.MINIDUMP\_PAGES により置き換え)
  - UFM.NUM\_FPREGS (代わりに UFM.FPREG0 から UFM.FPREG15 までを使用)
  - UFM.FPREG\_DATA\_ADDRESS (代わりに UFM.FPREG0 から UFM.FPREG15 までを使用)

## V10.1 から V11.1 へのマイグレーション

このセクションでは、Fault Analyzer をバージョン 10.1 からマイグレーションする場合に認識しておく必要がある、バージョン 11.1 での変更点に関する情報を説明します。

- Java™ 分析を使用可能にするには、IDIS subsystem始動 JCL にデータ・セット IDI.SIDIAUT2 を STEPLIB として追加する必要があります。詳しくは、[IDIS サブシステムの始動 ページ 321](#)を参照してください。
- Message and Abend Code Explanation ロード・モジュール・ユーザー出口のユーザーは、XPL.ABEND\_CODE ファイルへのオフセットが変更されたことに注意してください。

- 以前は、対話式再分析中のコンパイラー・リストまたはサイド・ファイルの欠落に関するプロンプトは、これらが障害項目の再分析で必要になる時点までに、暗黙的または明示的にコンパイラー・リストまたはサイド・ファイルのデータ・セットが Fault Analyzer に提供されていたかどうかによって左右されました。その結果、ある障害項目にはプロンプトが出され、他の障害項目には出されない場合があります。

ユーザー指定のオプションがプロンプトのコントロールに現在使用可能になったので ([対話式再分析オプション ページ 162](#) を参照)、一貫した動作を簡単に確認できます。ユーザーがこのオプションをまだ明示的に設定していない場合は、以下のいずれかのアクションによってデフォルト設定を制御できます。

- 1 つ以上のコンパイラー・リストまたはサイド・ファイルのデータ・セットを IDICNFxx parmlib メンバーに指定します (この場合、デフォルトでプロンプトが出されます)。
- これらのデータ・セットを IDICNFxx parmlib メンバーに指定しないようにします (この場合、デフォルトでプロンプトは出されません)。

## V9.1 から V10.1 へのマイグレーション

このセクションでは、Fault Analyzer をバージョン 9.1 からマイグレーションする場合に認識しておく必要がある、バージョン 10.1 での変更点に関する情報を説明します。

- IDIJ subsystem は、もう使用されていません。

## 第 12 章. Fault Analyzer のカスタマイズの準備

本章は、ご使用のインストール・システムにおける Fault Analyzer のカスタマイズ方法、およびグローバル・デフォルト・オプションの設定方法について説明します。単独のジョブまたは再分析用にオプションを設定または変更することもできます。[リアルタイム分析 ページ 33](#) および [Fault Analyzer ISPF インターフェース ページ 62](#)には、このような場合のオプションの調整方法が説明されています。

グローバル・デフォルト・オプションは、Fault Analyzer の実行方法に影響を与えます。例えば、分析するジョブの指示、レポートに含める詳細のレベル、およびコンパイラ・リストやサイド・ファイルの配置場所などのオプションがあります。

Fault Analyzer をカスタマイズするには、その前に、これをインストールする必要があります。SMP/E インストールの説明は、次のプログラム・ディレクトリ: *IBM Fault Analyzer for z/OS®*に記載されています。

インストール・システム全般のデフォルト・オプションは、parmlib メンバー IDICNFxx に含まれています。

分析プロセスの一部として、Fault Analyzer はコンパイラ・リストまたはサイド・ファイルの検出を試行します。[コンパイラ・リストまたは Fault Analyzer サイド・ファイルの提供 ページ 377](#)では、Fault Analyzer で使用できるようにするためのリストの保管方法や、サイド・ファイルの作成および保管方法について説明しています。本章はまた、IDILANGX 処理で必要とされるコンパイラ・オプションについても説明しています。

Fault Analyzer の要件として、REXX ユーザー出口が呼び出される場合、または診断トレースが IDITRACE DDname で要求される場合、REXX サポートが標準 MVS™ 検索パスから使用可能である必要があります。

以下のタスクは、Fault Analyzer が高位修飾子 IDI を使用してターゲット・ライブラリーにインストールされていることを前提としています。異なる高位修飾子を使用して Fault Analyzer がインストールされている場合は、ご使用の高位修飾子を IDI の代わりに使用してください。

### Fault Analyzer のインストールおよびカスタマイズのチェックリスト

Fault Analyzer のインストールを検証し、Fault Analyzer の使用をサイトで開始するには、以下のタスクを実行する必要があります。オプションであることが明記されていないタスクは、すべて必須です。



- 処理を開始する前にデータ・セット IDI.SIDISAM1 のすべてのメンバーをコピーし、変更をコピーのみに加えるようにしてください。
- IDICHKI ユーティリティを使用して、Fault Analyzer のインストール状況を確認したり、サービス情報を照会したりできます。[ステップ 3: サービス・レベルを検証する \(オプション\) ページ 460](#)を参照してください。

#### 1. Fault Analyzer モジュールの LINKLIST および LPA を介した有効化

詳しくは、[Fault Analyzer モジュールの有効化 ページ 307](#)を参照してください。

## 2. ヒストリー・ファイルの割り振り

最終的に複数のヒストリー・ファイルが自分のサイトで使用される可能性があります。Fault Analyzer のインストールを検証するにはヒストリー・ファイルは 1 つあれば十分です。

障害ヒストリー・ファイルの名前に制限はありませんが、Fault Analyzer が検索するデフォルトの名前は、IDI.HIST です。これとは異なる名前を使用する場合、IDICNFxx parmlib メンバーが使用され、DataSets オプションを介してその名前が提供されます。インストール・プロセスの後半で、IDICNFxx parmlib メンバーと、そのメンバーに含まれる可能性のあるオプションを検討します。

初期ヒストリー・ファイルの推奨サイズは 100 シリンダーです。

ヒストリー・ファイルの一般情報については、[ヒストリー・ファイルのセットアップ ページ 345](#)に記載されています。ここには、PDS または PDSE フォーマット選択時の考慮事項や、データ・セット割り振り用サンプル・ジョブの使用法も記載されています。

## 3. IDICNFxx parmlib メンバーの作成

詳しくは、[サイトのデフォルト・オプションの設定および変更 ページ 372](#)を参照してください。

次のことを確認してください。

```
DataSets (IDIHIST (dsn))
```

オプションが組み込まれていること (ステップ 2 ページ 293 でヒストリー・ファイルを IDI.HIST 以外の名前で割り振った場合)。

同様に、Fault Analyzer 以外の上位修飾子を使用して IDI をインストールした場合、DataSets オプションにより、必要なすべての Fault Analyzer データ・セット名が指定されます。

## 4. メッセージおよび異常終了コード説明リポジトリの定義および初期化

詳しくは、[メッセージおよび異常終了コード説明リポジトリのセットアップ ページ 309](#)を参照してください。

## 5. MVS™ 変更オプション/抑止ダンプ出口 IDIXDCAP のインストール

詳しくは、[MVS 変更オプション/抑止ダンプ出口のインストール IDIXDCAP ページ 334](#)を参照してください。

この出口の機能については、[Fault Analyzer を呼び出すための出口 ページ 299](#)に記載されています。

このステップが完了したら、ご使用のサイトで Fault Analyzer を効率的に使用できます。この段階で異常終了の分析とヒストリー・ファイルの項目の作成が開始される場合があります。

## 6. Language Environment® 異常終了出口の有効化 IDIXCEE

詳しくは、[言語環境プログラム異常終了出口 \(IDIXCEE または IDIXCE64\) の有効化 ページ 334](#)を参照してください。

ご使用のサイトでこの出口の適用度を判別するには、[Fault Analyzer を呼び出すための出口 ページ 299](#)および以下の呼び出しに必要となる [Language Environment オプション Fault Analyzer ページ 304](#)を参照してください。

この出口がインストールされていない場合、LE 対応プログラムでの異常終了が取り込まれるのは、IDIXDCAP 出口がインストールされており、いずれかの UA\* 値 (UATRACE または UADUMP など) を指定した LE TERMTHDACT オプションが有効である場合に限られます。

#### 7. SVC ダンプ登録出口 IDIXTSEL のインストール

詳しくは、[MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#)を参照してください。

#### 8. CICS® 環境のカスタマイズ

このステップは、CICS® を使用している場合にのみ適用できます。

詳しくは、[CICS 環境のカスタマイズ ページ 400](#)を参照してください。

#### 9. DB2® 環境のカスタマイズ

このステップは、DB2® を使用している場合にのみ適用できます。

詳しくは、[DB2 環境のカスタマイズ ページ 422](#)を参照してください。

[Fault Analyzer DB2 パフォーマンスの向上 ページ 422](#)で説明されている DB2® テーブル索引を作成することをお勧めします。これは、DB2® カタログ情報へのアクセス時には、Fault Analyzer の重大なパフォーマンス低下が生じる可能性があるためです。

#### 10. IMS™ 環境のカスタマイズ

このステップは、IMS™ を使用している場合にのみ適用できます。

詳しくは、[IMS 環境のカスタマイズ ページ 424](#)を参照してください。

#### 11. ISPF のためのカスタマイズ

詳しくは、[ISPF 環境の変更 ページ 328](#)を参照してください。

#### 12. Fault Analyzer IDIS サブシステムの始動

詳しくは、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照してください。

#### 13. IFAPRDxx parmlib メンバー内に項目を追加

詳しくは、[Fault Analyzer の IFAPRDxx parmlib メンバーへの登録 ページ 317](#)を参照してください。

### オプションのインストール・ステップ

#### 1. Fault Analyzer プログラムの BPX セキュリティー・サーバー・プログラム制御プロファイルの追加

このステップは、ご使用のインストール済み環境に対してプログラム制御がアクティブになっている場合にのみ必要です。

詳しくは、[Fault Analyzer プログラムへのプログラム制御アクセスの定義 ページ 308](#)を参照してください。

## 2. USERMOD IDISPLI または IDISPLIA のインストールによる PL/I V2R3 アプリケーションからの Fault Analyzer の暗黙呼び出しの有効化

詳しくは、[PL/I V2R3 アプリケーションからの Fault Analyzer 暗黙呼び出しの有効化\(++IDISPLI/++IDISPLIA\) ページ 336](#)を参照してください。

## 3. IDIOPTLM 構成オプション・モジュールを使用して、デフォルト・リカバリー障害記録 IEATDUMP データ・セット名を変更

詳しくは、[デフォルト・リカバリー障害記録 IEATDUMP データ・セット名 \(RFRDSN\) の変更 ページ 341](#)を参照してください。

## 4. XFACILIT リソース・クラスの定義によるリカバリー障害記録データ・セットの管理

詳しくは、[リカバリー障害記録データ・セット・アクセスの管理 ページ 310](#)を参照してください。

## 5. IDIOPTLM 構成オプション・モジュールを使用して、デフォルトの拡張ミニダンプ (XDUMP) データ・セット名パターンを変更

詳しくは、「[デフォルトの拡張ミニダンプ \(XDUMP\) データ・セット名パターン \(XDUMPDSN\) の変更 ページ 342](#)」を参照してください。

## 6. XDUMP データ・セット管理のための XFACILIT リソース・クラスの定義

詳しくは、「[XDUMP データ・セット・アクセスの管理 ページ 313](#)」を参照してください。

## 7. 日本語言語サポートのカスタマイズ

このステップは、Fault Analyzer の日本語機能がインストールされている場合にのみ必要です。

詳しくは、[Fault Analyzer 日本語フィーチャーのカスタマイズ ページ 425](#)を参照してください。

## 8. Fault Analyzer ヒストリー・ファイルにアクセスするためのオプションの非 ISPF インターフェースのインストール

詳しくは、[Fault Analyzer ヒストリー・ファイルにアクセスするための非 ISPF インターフェースのインストール ページ 563](#)を参照してください。

## 9. ISPF インターフェースを介した設定の変更のためのヒストリー・ファイル管理者権限の付与

詳しくは、[ヒストリー・ファイル設定の変更の制限 ページ 308](#)を参照してください。

## 10. ADFz 製品ファミリーで使用するプログラムを準備するために必要となる手順については、「*Quick start guide for compiling and assembling programs for use with the ADFz family of products*」の章『IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide』を確認してください。

## 11. SLIP トラップを設定して、選択した Fault Analyzer エラー・メッセージに関する文書を取り込む

詳しくは、[Fault Analyzer SLIP トラップの設定 ページ 306](#)を参照してください。

ユーザー出口を使用した Fault Analyzer のカスタマイズ [ページ 464](#)に記載されているように、ユーザー出口を使用して、その他のカスタマイズも必要に応じて実行できます。ただし、Fault Analyzer を実行するためにユーザー出口は不要です。

インストール検査

**1. アセンブラー IVP の実行**

詳しくは、[アセンブラーによる Fault Analyzer の使用の検証 ページ 427](#)を参照してください。

**2. COBOL IVP の実行**

このステップは、ご使用のサイトに COBOL がインストールされている場合にのみ実行してください。

詳しくは、[COBOL による Fault Analyzer の使用の検証 ページ 428](#)を参照してください。

**3. PL/I IVP の実行**

このステップは、ご使用のサイトに PL/I がインストールされている場合にのみ実行してください。

詳しくは、[PL/I による Fault Analyzer の使用の検証 ページ 429](#)を参照してください。

**4. IDIXCEE Language Environment® 出口 IVP の実行**

詳しくは、[IDIXCEE Language Environment 出口の有効化の検証 ページ 431](#)を参照してください。

**5. CICS® IVP の実行**

このステップは、ご使用のサイトに CICS® がインストールされている場合にのみ実行します。

詳しくは、[CICS での Fault Analyzer のカスタマイズの検証 ページ 432](#)を参照してください。

**6. DB2® IVP の実行**

このステップは、ご使用のサイトに DB2® がインストールされている場合にのみ実行します。

C と COBOL の両方の IVP が提供されます。詳しくは、[DB2 による Fault Analyzer の使用の検証 ページ 433](#)を参照してください。

**7. ISPF IVP の実行**

詳しくは、[ISPF による Fault Analyzer の使用の検証 ページ 438](#)を参照してください。

## インストール完了後のライブラリー名

Fault Analyzer の SMP/E APPLY 完了後は、以下のデータ・セットが存在するはずです。

データ・セット名	内容
IDI.SIDIALPA	LPA から使用可能にすべきロード・モジュール。
IDI.SIDIAUTH	LINKLIST から使用可能にすべき許可済みロード・モジュール。
IDI.SIDIDOC1	メッセージおよび異常終了コード説明の指定変更ファイル、レポート例、および Fault Analyzer User's Guide and Reference PDF。
IDI.SIDIDOC2	メッセージおよび異常終了コード説明リポジトリ入力データ。  <a href="#">メッセージおよび異常終了コード説明リポジトリのセットアップ ページ 309</a> で説明されているように、IDI.SIDIDOC2 データ・セットは IDISVENU



データ・セット名	内容
	ジョブにのみ入力されます。IDISVENU ジョブはメッセージと異常終了コード・リポジトリの割り当てと初期化を行います。
IDI.SIDIEEXEC	REXX EXEC。
IDI.SIDIMAPS	制御ブロック・マップ。
IDI.SIDIMLIB	ISPF メッセージ・メンバー。
IDI.SIDIPLIB	ISPF パネル。
IDI.SIDISAM1	ソフトコピーの例およびインストール・ジョブ。
IDI.SIDISLIB	ISPF スケルトン。
IDI.SIDITLIB	ISPF 表。

Fault Analyzer の日本語フィーチャーがインストールされている場合は、以下の追加データ・セットが存在するはずですが、

データ・セット名	内容
IDI.SIDIDJPN	日本語メッセージおよび異常終了コード説明の指定変更ファイル、レポート例、および Fault Analyzer User's Guide and Reference PDF。
IDI.SIDIMJPN	日本語 ISPF メッセージ・メンバー。
IDI.SIDIPJPN	日本語 ISPF パネル。
IDI.SIDISJPN	日本語 ISPF スケルトン。
IDI.SIDITJPN	日本語 ISPF 表。
IDI.SIDIXJPN	日本語ソフトコピーの例およびインストール・ジョブ。

## 推奨されるストレージ

異常終了後のリアルタイム実行では、異常終了した領域内で追加のストレージを必要とします。分析は、ストレージ内のデータについて実行されます。

Language Environment® と Fault Analyzer がいずれも LPA から使用不可能であることを前提とした場合、使用可能な**最小領域サイズ**の要件は次のとおりです。

- 実行環境に関係なく、最小 440 KB の境界以下 (24 ビット) のストレージ。
- CICS® トランザクションに対しては、最小 35 MB の境界以上 (31 ビット) のストレージ。
- CICS トランザクション以外に対しては、最小 33 MB の境界以上 (31 ビット) のストレージ。

分析する障害のタイプと、この障害が発生した環境によっては、さらにストレージが必要になる場合があります。

CICS では、CICS DSA 管理のストレージではなく、MVS GETMAIN 管理のストレージがストレージ要件とされています。そのため、境界以下の MVS GETMAIN 管理のストレージを増やすには、境界以下の CICS DSA 管理のストレージを減らす必要がある場合があります (境界以上のストレージについても同様です)。

Fault Analyzer が使用する実際のストレージについての情報は、リアルタイム分析レポートの終わりで参照できます。ただし、レポートに示されるストレージの量は、Fault Analyzer が実行した明示的な割り振りのみを示し、Language Environment® ヒープ、ロード・モジュール用に使用されるスタック・ストレージまたはストレージなどは含まれません。

ミニダンプまたは関連する MVS ダンプ・データ・セットが処理されている異常終了後の状態では、参照されたダンプ・ページのスペースを割り振った結果として、ストレージ要件の限界がリアルタイム実行のストレージ要件の限界よりも大きくなります。この増加は通常、500KB 未満です。

対話式再分析の場合、このストレージは TSO 領域内に存在する必要があります。

境界以上の使用可能な最小領域サイズは、必要なモジュールのサイズにより削減できます。これらのモジュールは、LPA から入手できるか (この場合ロードする必要はない)、既にロードされています (異常終了したプログラムが LE を使用する場合など)。

LPA に LE が含まれる場合は約 8 MB 節約でき、LPA に と Fault Analyzer が含まれる場合は約 13 MB 節約できます。その結果、標準的な非 CICS プログラムのストレージ要件は約 12 MB まで削減されます。

境界以下 (24 ビット) の必要なサイズが使用できない場合は、メッセージ [IDI0086E ページ 714](#) が発行され、処理が終了します。

境界以上 (31 ビット) の必要なサイズが使用できない場合は、メッセージ [IDI0055E ページ 707](#) が発行され、処理が終了します。LPA にモジュールを追加するために、メッセージ・テキストに含まれているコマンドを実行した場合、使用可能なストレージに関する情報を提供するメッセージ [IDI0087I ページ 714](#) も発行されることがあります。メッセージに含まれる可能性があるモジュール名は、Fault Analyzer モジュール IDIDA および IPVLANGX です。これらのモジュールを LPA に配置し、約 14 MB の境界以上 (31 ビット) のストレージを節約するには、次のオペレーター・コマンドを発行します。

```
SETPROG LPA,ADD,MOD=(IDIDA,IPVLANGX),DSN=LNKLST
```

上記の SETPROG コマンドを COMMNDxx parmlib メンバーに追加して、IPL 全体で保持することができます。

IPVLANGX の場合、代わりに IPV.SIPVLPA1 データ・セットを LPA.LSTxx parmlib メンバーに追加します (IDIDA が IPV.SIPVLPA1 にないため、SETPROG 経由で追加する必要があります)。



**注:**

- Fault Analyzer モジュールが LPA にロードされる場合、Fault Analyzer 保守の適用後に必ずステップ [ステップ 3: サービス・レベルを検証する \(オプション\) ページ 460](#) を実行してください。保守のインストール後に上記のステップを実行しないと、Fault Analyzer LPA モジュールが更新されません。すべての Fault Analyzer モジュールが LPA にあるとは限らないため、古いコードと新しいコードで不一致が生じ、これにより未定義の動作が発生する可能性があります。



- MVS では、LPALIST 内のデータ・セットが、マスター・カタログ内に存在するか、またはそのデータ・セットが配置されているボリューム通し番号を使用して指定される必要があります。CLPA を使用して IPL を実行し、LPALSTxx の変更をインプリメントしてください。
- IPV.SIPVLP1 および IPVLANGX は ADFz Common Components のインストールの一部としてインストールされます。

JCL 変更がすべてのジョブに対して実用的でない場合は、より大きい領域サイズを指定する一般的な方法として、MVS IEFUSI 出口を使用できます。IEFUSI のサンプル出口は、IDI.SIDISAM1 データ・セット内のメンバー IDISUSI として提供されます。出口は、すべてのジョブの領域サイズを 16MB 増加させます。出口のインストール方法の詳細については、サンプルの中のコメントを参照してください。

## Fault Analyzer を呼び出すための出口

Fault Analyzer には、それを異常終了のリアルタイム分析、または SVC ダンプ登録のために起動する多くの出口が用意されています。適用可能などの異常終了状態でも Fault Analyzer が呼び出されるようにするには、すべてをインストールする必要があります。

CICS® は固有のトランザクション・ディスパッチング・メカニズムを持っているために、CICS® の呼び出し出口は固有です。CICS 以外の実行環境は一般的に「バッチ」と呼ばれ、例えば CICS® なども含めて IMS™ 以外のすべてを意味します。

## 非 CICS トランザクション異常終了の場合の呼び出し

以下の出口は、CICS® トランザクション異常終了を除く異常終了 (例えば、バッチや IMS™) の発生時にリアルタイム分析のために Fault Analyzer を呼び出します。

### MVS™ IEAVTABX 変更オプション/抑止ダンプ出口 IDIXDCAP

- この出口は、Language Environment® ベースと非 Language Environment ベースのバッチ・アプリケーション・プログラムで使用できます。IDIXDCAP は、ジョブ・ステップで SYSMDUMP、SYSUDUMP、または SYSABEND DDname が割り振られているかどうかにかかわらず、IEAVTABX\_EXIT 動的出口 (z/OS 2.2 以降が必要) としてインストールされ、すべての異常終了に対して呼び出されます。

ACTION=NODUMP が指定された SLIP が使用される場合 (大部分の MVS™ システムで SLIP TRAP が一致する場合、Sx22 異常終了となる CANCEL コマンドなど)、IEAVTABX\_EXIT を使用しての Fault Analyzer の呼び出しが防止されない可能性があります。Fault Analyzer の呼び出しを防止するには、ご使用のシステムに該当する異常終了コードのリストを含めた Exclude オプションの指定が IDICNFxx PARMLIB 構成メンバーに含まれるようにします。標準の異常終了コードのサンプル・リストは、データ・セット IDI.SIDISAM1 のメンバー IDICNF00 として指定されたサンプル IDICNFxx parmlib メンバーに含まれています。

- ミニダンプが書き込まれた場合、この出口を使用して取り込まれた障害の再分析を実行できます。
- LE が使用可能な異常終了の場合は、LE がシステム・ダンプを呼び出してこの出口をアクティブにするように、LE オプションで UATRACE、UADUMP、UAONLY、または UA IMM サブオプションを指定する TERMTHDACT を使用

する必要があります。他のすべての TERMTHDACT サブオプション設定で、IEAVTABX 出口呼び出しがスキップされ、代わりに CEEEXTAN 出口 (下に説明) が呼び出されます。

- この出口では、異常終了するジョブに関連した WTO コンソール・メッセージをマスター・トレース・テーブルから抽出し、これらのメッセージを分析レポートに含めることができます。

この出口のインストールについては、[MVS 変更オプション/抑止ダンプ出口のインストール IDIXDCAP ページ 334](#)を参照してください。

## バッチ LE 異常終了 CEEEXTAN CSECT 出口 IDIXCEE および IDIXCE64

- この出口は、Language Environment® ベースのバッチ・アプリケーション・プログラムのみで有効です。
- この出口を呼び出すために JCL SYSMDUMP DD ステートメントを割り振る必要はありません。
- ミニダンプが書き込まれた場合、この出口を使用して取り込まれた障害の再分析を実行できます。
- インスタンス固有の LE メッセージの挿入を取得して、分析レポートに組み込めるようにします。
- この出口では、異常終了するジョブに関連した WTO コンソール・メッセージをマスター・トレース・テーブルから抽出し、これらのメッセージを分析レポートに含めることができます。
- **AMODE64 の場合:** MVS IEAVTABX 変更オプション/抑止ダンプ出口が言語環境ベースの AMODE64 バッチ・アプリケーション・プログラムで呼び出されることはありません。このため、言語環境では IDIXCE64 出口が必要になります。
- **AMODE31 の場合:** LE オプション TERMTHDACT が UATRACE、UADUMP、UAONLY、または UA IMM サブオプションとともに使用される場合、LE 異常終了出口ではなく、MVS™ IEAVTABX 変更オプション/抑止ダンプ出口が呼び出されます。

AMODE 31 プログラムと AMODE 64 プログラムの両方を実行している場合、IDIXCEE 出口と IDIXCE64 出口の両方をインストールする必要があります。

この出口のインストールについては、[言語環境プログラム異常終了出口 \(IDIXCEE または IDIXCE64\) の有効化 ページ 334](#)を参照してください。

## MVS IDIXDCAP とバッチ LE IDIXCEE 出口の両方がインストールされている

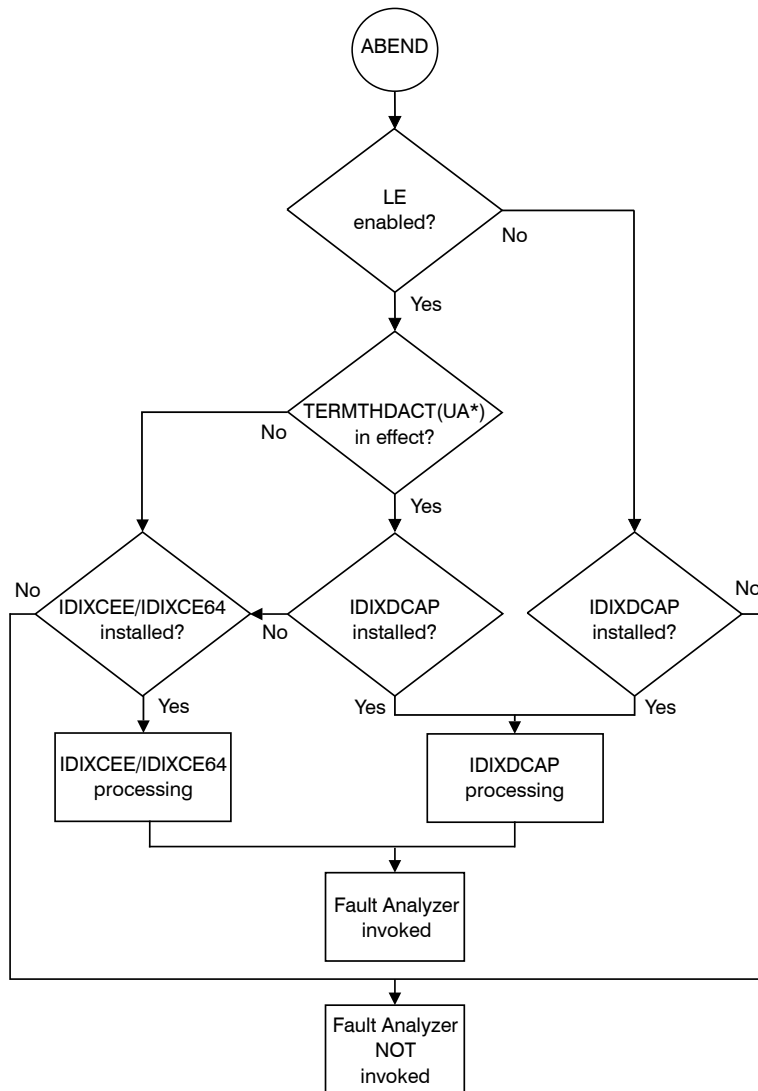
バッチ Language Environment® 異常終了出口 IDIXCEE と MVS™ 変更オプション/抑止ダンプ出口 IDIXDCAP の両方がインストールされていて、以下のいずれかの LE オプションが有効な場合は、LE 出口ではなく IDIXDCAP 出口が異常終了をインターセプトします。

- TERMTHDACT(UATRACE)
- TERMTHDACT(UADUMP)
- TERMTHDACT(UAONLY)
- TERMTHDACT(UA IMM)

## 出口使用の概要

図 186 : Fault Analyzer 非 CICS (バッチ) 呼び出し出口使用の概要 ページ 302 は、実行環境および有効なオプションに応じて、Fault Analyzer の呼び出しに使用される出口を示しています。

図 186. Fault Analyzer 非 CICS (バッチ) 呼び出し出口使用の概要



## CICS トランザクション異常終了の場合の呼び出し

以下の出口はすべて、CICS® トランザクション異常終了発生時にリアルタイム分析のために Fault Analyzer を呼び出します。

### CICS XPCABND および XDUREQ グローバル・ユーザー出口または IDIXCX53

特性:

- この出口は、CICS® トランザクション異常終了分析用に Fault Analyzer を起動するために提供されています。

Language Environment® ・ベースのアプリケーションでの U1xxx タイプまたは U4xxx タイプの異常終了を除き、すべてのトランザクション異常終了を、この出口を使用して取り込むことができます。これらのトランザクション異常終了のタイプは、CICS LE 異常終了 CEECXTAN CSECT 出口 IDIXCCEE (以下を参照) をインストールしても処理できます。

- この出口を呼び出すために JCL SYSMDUMP DD ステートメントを割り振る必要はありません。
- ミニダンプが書き込まれた場合、この出口を使用して取り込まれた障害の再分析を実行できます。

この出口のインストールについては、[CICS 環境のカスタマイズ ページ 400](#)を参照してください。

### CICS LE 異常終了 CEECXTAN CSECT 出口 IDIXCCEE

特性:

- この出口は、Language Environment® ・ベースの CICS® アプリケーション・プログラムのみで有効です。
- この出口を呼び出すために JCL SYSMDUMP DD ステートメントを割り振る必要はありません。
- ミニダンプが書き込まれた場合、この出口を使用して取り込まれた障害の再分析を実行できます。
- LE オプション TERMTHDACT は、この出口の呼び出しに影響を与えません。

この出口のインストールについては、[Language Environment for CICS を呼び出すための構成 Fault Analyzer ページ 401](#)を参照してください。

## SVC ダンプ登録

Fault Analyzer には、ヒストリー・ファイルへの SVC ダンプ登録用の出口が用意されています。

### MVS™ ポスト・ダンプ IEAVTSEL CSECT 出口 IDIXTSEL

特性:

- この出口は、SVC ダンプが DUMPSRV アドレス・スペースにより書き込まれるたびに呼び出されます。
- この出口を使用するには、Fault Analyzer IDIS サブシステムがアクティブである必要があります。この詳細については、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照してください。

- 分析は実行されませんが、ダンプ登録障害項目が作成されます。この障害項目が最初に再分析される際、レポートとミニダンプが追加されます。
- この出口は、基本的に CICS® システム・ダンプおよびリカバリー障害記録 SDUMP を記録することを目的にしています。

この出口のインストールについては、[MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#)を参照してください。

## 以下の呼び出しに必要なとなる Language Environment オプション Fault Analyzer

以下に説明するように、リアルタイム異常終了を取り込むための特定の言語環境プログラム (LE) オプションの必要性は、実行環境によって異なります。

### 非 CICS 異常終了に必要な LE オプション

CEEEXTAN LE 異常終了出口 (IDIXCEE) がインストールされている場合、異常終了時に Fault Analyzer for z/OS® を起動するために必要な特定の LE オプションはありません。ただし、MVS™ IEAVTABX 変更オプション/抑止ダンプ出口 (IDIXDCAP) が、LE 異常終了を取得するために使用される場合は、TERMTHDACT(UADUMP) などの、LE が SYSABEND、SYSUDUMP、または SYSMDUMP を使用できるようにするためのオプションが必要です。このオプションにより、MVS™ ダンプが書き込まれるときに、IEAVTABX 出口による制御が可能になります。



**注:** CEEEXTAN と IEAVTABX の両方の出口がインストールされている場合、Fault Analyzer for z/OS® は処理を解決して、障害の分析を 1 つのみ実行します。

### Java アプリケーション異常終了の取り込みに必要な LE オプション

Java™ アプリケーションの異常終了を Fault Analyzer で取り込むためには、例えば TERMTHDACT(UAIMM) など、UA\* サブオプションの 1 つを指定した LE TERMTHDACT オプションが有効でなければなりません。

次のコマンド

```
oedit .profile
```

を OMVS セッションから発行して、ユーザー・プロファイルを編集し、以下を指定することでこのオプションを設定できます。

```
export _CEE_RUNOPTS="TERMTHDACT(UAIMM)"
```

### CICS 異常終了に必要な LE オプション

CICS® 異常終了時に適用される必須の LE オプションはありません。

CICS® トレースの考慮事項については、[LE による CICS トレースの循環を回避 ページ 417](#)も参照してください。



## 類似したサード・パーティー製品を使用した Fault Analyzer の実行

一般に、Fault Analyzer は、Language Environment® バッチ・ジョブを除き、類似した他のサード・パーティー製品とともに問題なく動作します。Fault Analyzer は、類似した他のサード・パーティー製品と同じように、Language Environment® CEEEXTAN ファシリティー (IDIXCEE) を使用します。CEEEXTAN リストに複数の出口を指定する場合、最初の出口には Fault Analyzer を指定する必要があります。

Fault Analyzer による LE ジョブの分析も、IEAVTABX MVS™ オプション変更/ダンプ抑止出口 (IDIXDCAP) を使用して行うことができます。異常終了に関する MVS™ ダンプを LE に要求させるために、UATRACE、UADUMP、UAONLY、または UAIMM サブオプションを指定した TERMTHTACT オプションが LE オプションに含まれるようにしてください。このようにサード・パーティー製品は CEEEXTAN 出口を使用でき、Fault Analyzer は、IEAVTABX MVS™ 変更オプション/抑止ダンプ出口から実行できます。Fault Analyzer IDIXDCAP 出口は、IDIXCEE 出口とまったく同じ方法で LE 異常終了を分析します。これは、これらの出口が実際には分析を行わず、単に Fault Analyzer の起動方法を提供するだけであるためです。CICS® を含めて、すべての Fault Analyzer 出口に対して実行されるのは、同じ Fault Analyzer 分析エンジンです。

非 LE バッチの場合、Fault Analyzer と類似サード・パーティー製品との間に既知の競合はありません。ただし、起動用に MVS™ ダンプが取得されることに依存する可能性がある類似サード・パーティー製品が影響を受けないようにするために、IDICNFxx parmlib メンバーの中で RETAINDUMP(ALL) オプションを指定することが推奨されています。Fault Analyzer がインストールされている唯一の異常終了分析製品の場合、RETAINDUMP(ALL) オプションを削除することができます。

Fault Analyzer 異常終了前 (XPCABND) 出口は、CICS トランザクション・ダンプ前グローバル・ユーザー出口 (XDUREQ) に影響を与えます。XPCABND 出口から呼び出されると、Fault Analyzer はデフォルトでトランザクション・ダンプを抑制します。トランザクション・ダンプが抑制されている場合、CICS は XDUREQ 出口プログラムを呼び出しません。XDUREQ 出口を呼び出すために CICS が必要な場合は、Fault Analyzer RetainCICSdump(ALL) オプションを使用してください。

## MVS ダンプ・データ・セットのサイズ

Fault Analyzer がインストールされている場合、取得される MVS™ システム・ダンプのサイズが大きくなることが予想されます。ダンプ・データ・セットの割り振りサイズ・パラメーターの検討が必要な場合があります。

## アプリケーション処理のエラー条件

異常終了の表示を完全に抑止し、また完了時に異常終了処理を続行できるようにする、アプリケーション・エラー・ハンドラーを作成することができます。

一般的に、異常終了処理は、エラー・ハンドラーがタスクを完了した後も続行できます。しかし、アプリケーション・プログラムのエラー・ハンドラーによって通常の異常終了が発生しない場合に、このアプリケーションに Fault Analyzer を呼び出すには、アプリケーションのエラー処理を使用不可にしたり、IDISNAP に呼び出しを追加したりすることが必要になる場合があります (詳しくは、[プログラム SNAP インターフェース \(IDISNAP\) の使用 ページ 39](#)を参照してください)。

Fault Analyzer を呼び出さないアプリケーションのエラー処理ルーチンの例としては、PL/I の「ON ERROR」ブロックがあります。これは、「S」オプションを使用するか STOP を発行して、PLIDUMP を呼び出します。

「S」オプションを使用している場合でも、PLIDUMP を呼び出すときに常に Fault Analyzer を起動するために使用できる PL/I USERMOD があります。詳細については、[PL/I PLIDUMP \(++\)IDISPDM\) からの Fault Analyzer の常時起動 ページ 336](#)を参照してください。

## Fault Analyzer SLIP トラップの設定

Fault Analyzer がエラー・メッセージを発行する特定のシチュエーションについての資料と同様に、MVS™ ダンプは IBM® サポートによって頻繁に必要になります。

Fault Analyzer は多くの場合、以下の 2 つの目的で使用される Recovery Fault Recording (RFR) ダンプを自動的に書き込みます。

- RFR 障害エントリーの再分析を可能にして、実質的に、エラーが発生したことをユーザーが分かるようにする。
- エラー (IBM 障害であると疑われる場合) に関する情報を、Fault Analyzer サポートに提供する。

このような RFR ダンプが書き込まれる (Fault Analyzer IDIS サブシステムの始動と IDIXTSEL 出口のインストールが必要で、詳しくは、[リカバリー障害記録 ページ 57](#)を参照) 場合、一般に、下で「\*」によってマークされたメッセージ ID 以外のメッセージ ID に SLIP トラップを設定する必要はありません。

ある特定の Fault Analyzer メッセージが発行される場合に SVC ダンプを取り込む SLIP トラップのサンプルは、次のとおりです。

```
SL SET, ID=xxxx, MSGID=zzzzzzzz, ACTION=SVCD, END
```

xxxx は固有の SLIP トラップ ID (例: F047) で、zzzzzzzz はメッセージ ID です。

SLIP コマンドの完全な構文と、考慮しなければならない場合がある追加のパラメーター (MATCHLIM など) については、「MVS™ システム・コマンド」を参照してください。

以下のメッセージ ID の SLIP トラップを考慮する必要があります。

[IDI0047S ページ 705](#)

[IDI0092S ページ 715](#)

[IDI0105S ページ 718](#)

[IDI0123S ページ 722 \\*](#)

[IDI0144E ページ 727 \\*](#)

[IDI0168E ページ 734 \\*](#)

\* これらのメッセージ ID は、Recovery Fault Recording (RFR) 処理には関係ないため、これらについて、MVS™ ダンプが Fault Analyzer によって自動的に書き込まれることはありません。

# 第 13 章. Fault Analyzer の稼働環境のカスタマイズ

この章では、Fault Analyzer の実行に必要なオペレーティング環境のカスタマイズについて説明します。

## Fault Analyzer モジュールの有効化

Fault Analyzer モジュールを有効にするには、Fault Analyzer の SMP/E インストール後に次の手順を実行する必要があります。

### 1. IDI.SIDIAUTH を許可して LINKLIST に追加する

APF 許可を必要とする Fault Analyzer モジュールは、ターゲット・ライブラリー IDI.SIDIAUTH に配置されます。SYS1.PARMLIB 内の IEAAPFxx メンバーまたは PROGxx メンバー (システムで使用可能な場合) に IDI.SIDIAUTH を追加することにより、その APF 許可を行う必要があります。IDI.SIDIAUTH も LINKLIST 内に存在する必要があります。SYS1.PARMLIB 内の LNKLISTxx メンバーまたは PROGxx メンバーを使用して IDI.SIDIAUTH を連結 LINKLIST に追加してください。



**注:** MVS では、LINKLIST 内のデータ・セットは、マスター・カタログに登録しておくか、またはそのデータ・セットがあるボリュームの通し番号を使用して指定する必要があります。

APF 許可ライブラリーからロードしない限り、IDI.SIDIAUTH 内のロード・モジュール IDIDA、IDIPDDIR および IDIUTIL は正しく実行されません。

### 2. IPV.SIPVMODA を LINKLIST へ追加する

データ・セット IPV.SIPVMODA は ADFz Common Components インストールの一部として作成されます。コンパイラー・リストまたは SYSADATA ファイルを使用するときに Fault Analyzer がソース・レベルのサポートを提供するには、これを LINKLIST に追加する必要があります。



**注:** MVS™ では、LINKLIST 内のデータ・セットは、マスター・カタログに登録しておくか、またはそのデータ・セットがあるボリュームの通し番号を使用して指定する必要があります。

ADFz Common Components はデータ・セット IPV.SIPVMOD1 も提供しますが、これは Fault Analyzer では不要です。

### 3. IDI.SIDIALPA を LPALIST に追加する

LPA にロードされなければならない Fault Analyzer モジュールは、ターゲット・ライブラリー IDI.SIDIALPA にあります。SYS1.PARMLIB 内の LPALSTxx メンバーを使用して、連結された LPALIST に IDI.SIDIALPA を追加してください。



**注:** MVS では、LPALIST 内のデータ・セットが、マスター・カタログ内に存在するか、またはそのデータ・セットが配置されているボリューム通し番号を使用して指定される必要があります。

LPALSTxx の変更は、CLPA を使用して IPL を実行するか (ステップ 5 ページ 308)、IPL をスケジュールできない場合は次の手順を実行することによって実装する必要があります。

- a. 次のオペレーター・コマンドを発行します。

```
SETPROG LPA,ADD,MOD=(IDICSVCR, IDICSRBL),DSN=IDI.SIDIALPA
```

- b. 次の EXEC JCL ステートメントを含むバッチ・ジョブを送信して、Fault Analyzer SVC 109 ESR コード 53 をシステムに動的にインストールします。

```
// EXEC PGM=IDICZSVC
```

CLPA を使用した IPL (ステップ 5 ページ 308)、または SETPROG と IDICZSVC を使用した上記の動的更新を介して IDI.SIDIALPA モジュールが LPA に追加されない場合、異常終了 S16D などのエラーが発生する可能性があります。

#### 4. Fault Analyzer と IPVLANGX を LPA に追加する

これはオプションですが、Fault Analyzer リアルタイム分析のために異常終了した領域に追加のストレージを提供するために推奨されるステップです。

詳しくは、[推奨されるストレージ ページ 297](#)を参照してください。

#### 5. CLPA で IPL を実行する

上記の手順のいずれかが動的に正常に実装されなかった場合は、CLPA でシステムを IPL する必要があります。

## Fault Analyzer プログラムへのプログラム制御アクセスの定義

ご使用のシステムのセキュリティー・サーバー・プログラム制御がアクティブ状態である場合 (例えば、z/OS Unix System Services (BPX) サーバー要件のために)、制御対象プログラムとなっているすべての Fault Analyzer プログラムを識別する PROGRAM クラス・プロファイルは、次のコマンドで定義できます。

```
RDEFINE PROGRAM IDI* ADDMEM('IDI.SIDIAUTH'//NOPADCHK) UACC(READ)
```

(詳しくは、セキュリティー・サーバーの資料を参照してください。)

Fault Analyzer プログラムを制御対象として定義しないと、z/OS Unix System Services サーバー領域で異常終了が発生した場合に、CSV042I、ICH420I、ICH422I、BPX014I などのメッセージが発行されることがあります。

## ヒストリー・ファイル設定の変更の制限

デフォルトでは、ヒストリー・ファイルに対して UPDATE アクセス権限を持つユーザーはすべて、ヒストリー・ファイル接頭部か、あるいは障害項目の最小数または最大数を変更することができます。

以下のいずれかの方法を使用できます。

- Fault Analyzer ISPF インターフェースのアクション・バー・オプション **File > Change Fault History File Settings**。詳しくは、[障害履歴・ファイル設定の変更 ページ 94](#)を参照してください。
- SetFaultPrefix、SetMaxFaultEntries、および SetMinFaultEntries バッチ・ユーティリティー機能。詳しくは、[IDIUTIL 制御ステートメント ページ 442](#)を参照してください。

所定の履歴・ファイルの設定の変更を、上記方法のいずれかを使用して制限するために、セキュリティ管理者は、履歴・ファイルの IDL\_ADMIN XFACILIT プロファイルを定義することができ、必要に応じて、その履歴・ファイルに対してアクセス権限を付与することができます。

図 187. Syntax

```
▶▶ IDL_ADMIN. history-file-dsn ◀◀
```

ここで、*history-file-dsn* は、履歴・ファイルの完全修飾データ・セット名です。

IDL\_ADMIN XFACILIT プロファイルが定義された後で履歴・ファイル設定を変更するには、ユーザーは、以下のアクセス許可を両方とも持っている必要があります。

- IDL\_ADMIN XFACILIT プロファイルに対する UPDATE (またはそれ以上の) アクセス権限
- 通常のセキュリティ・サーバー・データ・セット・プロファイルか、または XFACILIT を通じた、履歴・ファイルへの UPDATE (またはそれ以上の) アクセス権限 (詳細については、[履歴・ファイル障害項目のアクセスの管理 ページ 365](#)を参照)。

履歴・ファイル MY.HIST の IDL\_ADMIN XFACILIT プロファイルを定義する RACF® コマンドの例を以下に示します。この例では、グループ PAYROLL のメンバーであるユーザー用の設定を変更する Fault Analyzer 管理者権限を付与していません。

```
RDEFINE XFACILIT IDL_ADMIN.MY.HIST UACC(NONE)
PERMIT IDL_ADMIN.MY.HIST CLASS(XFACILIT) ID(PAYROLL) ACCESS(UPDATE)
```

## メッセージおよび異常終了コード説明リポジトリのセットアップ

Fault Analyzer のメッセージおよび異常終了コードの説明を使用可能にするには、VSAM クラスタが定義されて、初期化されている必要があります。この定義と初期化を行うには、IDISVENU データ・セット内のサンプル・ジョブ IDI.SIDISAM1 を実行依頼します。IDISVENU のデフォルト名以外の名前を使用する場合は、使用する名前を含む IDIVSENU サブオプションを指定して、DataSets オプションを指定する必要があります。詳しくは、[Detail ページ 586](#)を参照してください。

z/OS® と Fault Analyzer の両方で異なるバージョンを実行しているインストール済み環境では、&SYSR1 置換シンボルを使用して IDISVENU データ・セットを指定することを検討してください。例:

```
DataSets(IDIVSENU(IDI.IDIVSENU.&SYSR1.))
```

次に、これらの VSAM データ・セットを、対応する SYSRES ボリューム上に SYSRES VOLSER を指定して割り振ります。こうすると、IDICNFxx の IDIVSENU parmlib メンバー DataSets 指定は、どの SYSRES ボリュームから IPL が実行される場合でも常に正しくなります。

必ず、すべてのユーザーが、デフォルトで IDIDOC DDname および IDIVSENU DDname となるデータ・セット、またはこれらの DDname で指定されたデータ・セットへの READ アクセス権を持つようにしてください。また、このファイルは、IDIS サブシステムによって修正や新しい説明で自動的に更新されることがあるため、IDIS VSAM KSDS データ・セットに対する UPDATE アクセス権を IDISVENU サブシステムに付与する必要があります。

Fault Analyzer の日本語フィーチャーについては、IDI.SIDIXJPN データ・セット内のサンプル・ジョブ IDISVJPN を使用して追加のリポジトリをセットアップする必要があります。IDI.IDIVSJPN のデフォルト名以外の名前を使用する場合は、DataSets オプションの IDIVSJPN サブオプションを使用して、使用する名前を指定する必要があります。詳しくは、[Detail ページ 586](#)を参照してください。

必ず、すべてのユーザーが、デフォルトで IDIVSJPN DDname となるデータ・セット、またはそれらの DDname で指定されたデータ・セットへの READ アクセス権を持つようにしてください。また、このファイルは、IDIS サブシステムによって修正や新しい説明で自動的に更新されることがあるため、IDIVSJPN VSAM KSDS データ・セットに対する UPDATE アクセス権を IDIS サブシステムに付与する必要があります。

## リカバリー障害記録データ・セット・アクセスの管理

Fault Analyzer は、仮想ストレージの不足や Fault Analyzer の異常終了などの例外条件のためにアプリケーションの異常終了を記録できなかった場合、異常終了状態を MVS SDUMP (SVC ダンプ) または MVS IEATDUMP (トランザクション・ダンプすなわち TDUMP) で取り込もうとします。このプロセスでは、リカバリー障害記録のヒストリー・ファイル障害項目にリンクされた、別のダンプ・データ・セットが作成されます。このリカバリー・プロセスでは、取り込みの際に Fault Analyzer が検出した例外条件があるにもかかわらず、Fault Analyzer は、通常、初期アプリケーション異常終了の対話式再分析を行うことができます。リンクされた SDUMP または TDUMP は、取り込みに例外がなければ通常は障害項目の「ミニダンプ」セクション内に記録されたと考えられるストレージ・データを提供します。

通常のデータ・セットへのアクセスに加えて、Fault Analyzer はオプションとして、XFACILIT セキュリティー・プロファイルを使用して RFR tdump データ・セットを制御できるようにします (詳細については、[TDUMP RFR データ・セットに対する XFACILIT リソース・クラスの使用 ページ 312](#)を参照)。

Fault Analyzer RFR SDUMP データ・セットは、関連する FACILIT プロファイルを介してアクセスが許可された場合にのみ作成されます (詳細については、[SDUMP RFR データ・セットに対する XFACILIT リソース・クラスの使用 ページ 311](#)を参照)。

Fault Analyzer は、優先ダンプ・タイプとして SDUMP を使用しようとしています。異常終了したユーザー ID に対して必要な SDUMP アクセス許可を使用できない場合のみ、TDUMP アクセス許可が検査されます。SDUMP で使用されるダンプ・プロセスは、TDUMP プロセスよりも高速です。

ここで説明する Fault Analyzer XFACILIT プロセスをその RFR ダンプの SDUMP 制御または TDUMP 制御として使用している場合、実際の SDUMP または TDUMP は、それがリンクされている障害項目を分析または削除しない限り、通常のエンド・ユーザーからは読み取りや削除を行うことができません。例えば、「給与計算」アプリケーションが、一般ユーザーには読み取り権限がない独自のヒストリー・ファイルを持っている可能性がある場合、この XFACILIT プロセスは、一般ユーザーが障害項目にアクセスできないため、「給与計算」のすべての RFR SDUMP または TDUMP について一般ユーザーの使用が禁止されることを意味しています。

RFR SDUMP または TDUMP の XFACILIT アクセス要件は、以下に説明するように異なります。

## SDUMP リカバリー障害記録データ・セット

SDUMP が要求された場合、DUMPSRV によって決定される命名規則で、DUMPSRV アドレス・スペースによって生成されます。通常、システム・プログラマーを除くシステム上の大部分のユーザーは、UACC(NONE) によって制限され、これらの SDUMP に対しのみ許可されます。SDUMP を要求できるようになるためには、Fault Analyzer は、その内部 SVC プロセスから取得する許可状態を使用しなければなりません。

Fault Analyzer は、異常終了が発生したユーザー ID に対して、以下の XFACILIT リソース・クラス・セットアップを使用して XFACILIT アクセスが認可されている場合にのみ、リカバリー障害記録プロセスで SDUMP を使用します。

## SDUMP RFR データ・セットに対する XFACILIT リソース・クラスの使用

Fault Analyzer でその RFR プロセスにおいて SDUMP を使用するには、IDI\_SDUMP\_ACCESS という名前で XFACILIT クラス・プロファイルを設定アップし、RFR 例外に対して SDUMP が必要なユーザー ID またはグループに ALTER アクセス権を提供します。ユーザーの障害項目の作成で例外が発生した場合、以下の定義により、Fault Analyzer は CICS® グループ内のすべてのユーザーに対して SDUMP を作成できます。

```
RDEF XFACILIT IDI_SDUMP_ACCESS UACC(NONE)
PERMIT IDI_SDUMP_ACCESS CLASS(XFACILIT) ID(CICS) ACCESS(ALTER)
```

ALTER アクセス権は XFACILIT IDI\_SDUMP\_ACCESS プロファイルに対するものであり、実際の SDUMP データ・セットに対するものではありません。Fault Analyzer は、許可状態を使用して RFR SDUMP へのアクセスを許可します。IDI\_SDUMP\_ACCESS プロファイルはスイッチとして機能し、Fault Analyzer はそのスイッチを調べて、そのユーザー ID に対して SDUMP を作成するかどうかをチェックできます。

偶然に、障害項目の作成で RFR ダンプを必要とする例外が発生した場合には、ユーザーが XFACILIT Fault Analyzer プロファイルに対する ALTER アクセス権を持っている場合にのみ、IDI\_SDUMP\_ACCESS は SDUMP を作成して障害項目にリンクします。

問題分析を行っているユーザーが、障害項目に対する読み取りアクセス権および削除アクセス権を持っており、その障害項目に SDUMP がリンクされている場合 (障害項目はリカバリー障害記録例外によって作成されている)、Fault Analyzer は、障害項目の拡張として、その SDUMP に対する等価のアクセス権を提供します。障害項目を暗黙的に削除すると、リンクされたすべての SDUMP が削除されます。

SDUMP の取り込みは通常は TDUMP の取り込みよりもはるかに高速であるため、少なくとも CICS® などのパフォーマンスが重要なシステムには、上記のアクセス権を認可することによって、RFR SDUMP を使用する権限を与えることをお勧めします。

## TDUMP リカバリー障害記録データ・セット

このセクションでは、例外条件 RFR TDUMP の作成、読み取り、および削除に必要な権限について説明します。TDUMP リクエスターは、TDUMP データ・セット名を指名できるため、このセクションでは、命名規則プロセスについても説明しません。



**注:** ユーザー ID 高位修飾子の古い RFR TDUMP 命名規則は、ユーザー ID TDUMP 高位修飾子で頻繁に発生するセキュリティとデータ・セット削除の問題のため、もう使用されていません。

TDUMP リカバリー障害記録データ・セットに関する通常のデータ・セット・プロファイルのセキュリティの問題を克服するために、Fault Analyzer は、以下に説明するように、XFACILIT リソース・クラスの使用をサポートします。XFACILIT リソース・クラスを使用するとともに、UACC(NONE) を TDUMP リカバリー障害記録データ・セットの一般データ・セット・プロファイルのアクセス・レベルとして使用して、セキュリティの問題の可能性を防止することをお勧めします。UACC(NONE) と XFACILIT のセットアップの代わりに、すべてのユーザーに対して RFR TDUMP データ・セット・プロファイルへの ALTER アクセス権が認可され、作成が許可されている場合には、セキュリティの問題が発生する場合があります。

すべてのエンド・ユーザーが同様のアクセス権を持っている状態のシステムの場合、XFACILIT IDIRFR\_TDUMP\_HLQ をセットアップしないように選択し、代わりに TDUMP データ・セット・プロファイルに対する ALTER アクセス権をすべてのユーザーに与えると、引き続き RFR TDUMP が取られます。この環境では、システム上のヒストリー・ファイルに対してすべてのユーザーが同じアクセス権限を持つ可能性があります。しかし、一部のユーザーが必ずしもすべてのヒストリー・ファイルに対する読み取りアクセス権を持つとは限らない場合、データ・セット・プロファイルに対して IDIRFR\_TDUMP\_HLQ と UACC(NONE) を検討して、リンクされたすべての RFR TDUMP に保護を与える必要があります。

## TDUMP RFR データ・セットに対する XFACILIT リソース・クラスの使用

Fault Analyzer TDUMP リカバリー障害記録データ・セットに対して XFACILIT リソース・クラスをセットアップするには、高位修飾子を最初に判別する必要があります。

TDUMP リカバリー障害記録データ・セットのデフォルトのデータ・セット名パターンは、

```
IDIRFRHQ.IDIRFR.&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.
```

このデータ・セット名パターンを変更するには、[「デフォルト・リカバリー障害記録 IEATDUMP データ・セット名 \(RFRDSN\) の変更 ページ 341」](#) を参照してください。



**注:** このセクションでは単一の修飾子を使用するものとしていますが、アクセス制御のセットアップでは1つ以上の修飾子を使用できます。

使用する高位修飾子が指定されると、次の名前が XFACILIT クラス・プロファイルがセットアップされます。

```
IDIRFR_TDUMP_HLQ.hlq.**
```

*hlq* はリカバリー障害記録データ・セットの高位修飾子です。

例えば、TDUMP データ・セット名パターンが次に設定されている場合、

```
IDIRFRHQ.TDUMP.&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.
```

IDIOPTLM RFRDSN オプションを使用して、XFACILIT クラス・プロファイルを追加します

```
IDIRFR_TDUMP_HLQ.IDIRFRHQ.TDUMP.**
```



高位修飾子に、例えば TDUMP&SYSCNONE. などのシンボル名が含まれていると、予期されたシンボル置換値に応じて、複数のプロファイルを設定アップする必要がある場合があります。

XFACILIT プロファイル (または、シンボル置換により複数のプロファイル) が定義されている場合は、関係するユーザーに対して該当するレベル ALTER または NONE を指定します。XFACILIT クラスに対するユーザーのアクセス・レベルが ALTER の場合、高位修飾子がシンボル置換後に XFACILIT プロファイル名 *hlq* 値と一致するデータ・セットに対して、ユーザーは、Fault Analyzer を通して TDUMP 作成機能を暗黙的に所持します。

XFACILIT プロファイルへの汎用アクセス権限 ALTER では、リカバリー障害記録データ・セットを保護する通常のデータ・セット・プロファイルを指定変更しません。Fault Analyzer を使用してアクションを実行するときに、リンクされた TDUMP データ・セットに対して必要なアクセス権限を許可するのみです。このアクションには、再分析時のデータ・セットの読み取りや関連障害項目の削除時のデータ・セットの削除などがあります。

偶然に、障害項目の作成で RFR ダンプを必要とする例外が発生する場合があります。この場合、Fault Analyzer は、以下のいずれかの条件下でのみ TDUMP を作成して障害項目にリンクさせます。

- 適切な XFACILIT IDIRFR\_TDUMP\_HLQ プロファイルに対する ALTER アクセス権限を持っている。
- TDUMP データ・セット・プロファイルに対する ALTER アクセス権限を持っている。

Fault Analyzer は、以下の 2 つの条件に該当する場合、障害項目に対する拡張として、TDUMP に対する同等のアクセス権限を提供します。

- 問題分析を行っていて、障害項目に対する読み取りアクセス権限または削除アクセス権限を持っている。
- 障害項目に TDUMP がリンクされている (障害項目はリカバリー障害記録例外によって作成)。

障害項目を暗黙的に削除すると、リンクされたすべての TDUMP が削除されます。

## RFR TDUMP XFACILIT の例

以下は、Fault Analyzer リカバリー障害記録データ・セットを管理するための、XFACILIT クラスのセットアップ方法の例です。この例は必要に応じて、インストール済み環境で変更または拡張できます。

1. IDIRFR\_TDUMP\_HLQ.IDIRFRHQ.\*\* という名前の XFACILIT プロファイルを定義して、ALTER 汎用アクセス権限をこのプロファイルに付与します。

```
RDEFINE XFACILIT IDIRFR_TDUMP_HLQ.IDIRFRHQ.** UACC(ALTER)
```

2. NONE の汎用アクセス権限を持つ IDIRFRHQ.\* の総称データ・セット・プロファイルを定義します。

```
ADDSD 'IDIRFRHQ.**' UACC(NONE)
```

## XDUMP データ・セット・アクセスの管理

通常のデータ・セット・アクセスに加えて、Fault Analyzer はオプションで XFACILIT セキュリティ・プロファイルを使用して XDUMP データ・セットを制御できるようにします。

ここで説明する Fault Analyzer XFACILIT プロセスを XDUMP データ・セットの制御方式として使用している場合、実際の XDUMP データ・セットは、それがリンクされている障害項目を分析または削除しない限り、通常のエンド・ユーザーから

は読み取りや削除を行うことができません。例えば、“給与計算”アプリケーションが、一般ユーザーには読み取り権限がない独自のヒストリー・ファイルを持っている場合、この XFACILIT プロセスは、一般ユーザーが障害項目にアクセスできないため“給与計算”のすべての XDUMP データ・セットの使用が制限されることを意味します。

機密漏れの可能性を防止するには、UACC(NONE) を XDUMP データ・セットの一般データ・セット・プロファイルのアクセス・レベルとして使用します。UACC(NONE) と、続いて XFACILIT をセットアップする代わりに、作成を許可するために XDUMP データ・セット・プロファイルへの ALTER アクセス権がすべてのユーザーに付与されている場合には、機密漏れの問題が発生することがあります。特定のシステム上で、すべてのエンド・ユーザーが似たようなアクセス権を持っている場合、XFACILIT アクセス権をセットアップしないことを選択した場合にも XDUMP データ・セットが引き続き作成され、代わりにすべてのユーザーに XDUMP データ・セット・プロファイルに対する ALTER アクセス権が付与されます。この環境では、システム上のヒストリー・ファイルに対してすべてのユーザーが同じアクセス権を持つ可能性があります。ただし、一部のユーザーが必ずしもすべてのヒストリー・ファイルに対する READ アクセス権を持つとは限らない場合、リンクされているすべての XDUMP に保護を拡張するために、データ・セット・プロファイルに対して UACC(NONE) を指定した XFACILIT プロファイルを使用することを考慮する必要があります。

## XDUMP データ・セットに対する XFACILIT リソース・クラスの使用

XFACILIT クラス・プロファイルを `IDIXDUMP_HLQ.hlq.**` という名前を使用してセットアップします。*hlq* を、IDIOPTLM 構成オプション・モジュール内の XDUMPDSN オプションを使用して指定した、データ・セット名前パターンの 1 つ以上の修飾子で置き換えます。(詳しくは [デフォルトの拡張ミニダンプ \(XDUMP\) データ・セット名パターン \(XDUMPDSN\) の変更 ページ 342](#) を参照してください。)

高位修飾子に、例えば `XDUMP&SYSCLONE` などのシンボル名が含まれていると、予期されたシンボル置換値に応じて、複数のプロファイルを設定アップする必要がある場合があります。

XFACILIT プロファイル (または、シンボル置換により複数のプロファイル) が定義されている場合は、関係するユーザーに対して該当するレベル (ALTER または NONE) を指定します。XFACILIT クラスに対するユーザーのアクセス・レベルが ALTER の場合、高位修飾子がシンボル置換後に XFACILIT プロファイル名 *hlq* 値と一致するデータ・セットに対して、ユーザーは、Fault Analyzer を通して XDUMP 作成機能を暗黙的に所持します。

XFACILIT プロファイルへの汎用アクセス権限 ALTER では、XDUMP データ・セットを保護する通常のデータ・セット・プロファイルを指定変更しません。Fault Analyzer を使用してアクションを実行するときに、リンクされた XDUMP データ・セットに対して必要なアクセス権限を許可するのみです。このアクションには、再分析時のデータ・セットの読み取りや関連障害項目の削除時のデータ・セットの削除などがあります。

Fault Analyzer は、次のいずれかの条件下のみで、XDUMP を作成して障害項目にリンクします。

- 適切な XFACILIT `IDIXDUMP_HLQ` プロファイルに対する ALTER アクセス権を持っている。
- XDUMP データ・セット・プロファイルに対する ALTER アクセス権を持っている。

Fault Analyzer は、ユーザーが問題分析を実行しており、障害項目に対する READ アクセス権または DELETE アクセス権を持っている場合、XDUMP データ・セットに対して、関連する障害項目と同等のアクセス権を提供します。

障害項目を暗黙的に削除すると、関連するすべての XDUMP データ・セットも削除されます。

## XDUMP XFACILIT の例

以下は、Fault Analyzer XDUMP データ・セットを管理するための、XFACILIT クラスの推奨セットアップ例です。この例は必要に応じて変更したり、それを基に拡張したりできます。

この例では、IDIOPTLM 構成オプション・モジュールの XDUMPDSN オプションが、以下の値で指定されていると想定しています。

```
'IDIHLQ.XDUMP.&&SYSNAME..D&&YYMMDD..T&&HHMMSS..S&&SEQ.'
```

1. XFACILIT プロファイルを定義して、ALTER 汎用アクセス権限をこのプロファイルに付与します。

```
RDEFINE XFACILIT IDIXDUMP_HLQ.IDIHLQ.XDUMP.** UACC(ALTER)
```

2. NONE 汎用アクセス権限を持つ同じデータ・セットの総称データ・セット・プロファイルを定義します。

```
ADDS 'IDIHLQ.XDUMP.**' UACC(NONE)
```

## コピーされた SDUMP データ・セット・アクセスの管理

関連する密結合 SDUMP データ・セットを含む障害項目を移動またはコピーする場合、Fault Analyzer は元の SDUMP データ・セットのコピーを作成し、コピーされた障害項目およびコピーされた SDUMP データ・セットを共にリンクします。

Fault Analyzer では、XFACILIT セキュリティー・プロファイルを使用して、コピーされた SDUMP データ・セットを制御できます。

このトピックで説明するプロセスが、コピーされた SDUMP データ・セットを制御するのに使用される場合、実際のコピーされた SDUMP データ・セットは、それがリンクされている障害項目の分析または削除を通して以外、通常のエンド・ユーザーからは読み取りや削除を行うことができません。例えば、“給与計算”アプリケーションが、一般ユーザーには読み取り権限がない独自のヒストリー・ファイルを持っている場合、この XFACILIT プロセスは、一般ユーザーが障害項目にアクセスできないため、“給与計算”のすべてのコピーされた SDUMP データ・セットについて一般ユーザーの使用が禁止されることを意味しています。

機密漏れの可能性を防止するには、UACC(NONE) をコピーされた SDUMP データ・セットの一般データ・セット・プロファイルのアクセス・レベルとして使用します。UACC(NONE) と、続いて XFACILIT をセットアップする代わりに、作成を許可するためにコピーされた SDUMP データ・セット・プロファイルへの ALTER アクセス権がすべてのユーザーに付与されている場合には、機密漏れの問題が発生することがあります。特定のシステム上で、すべてのエンド・ユーザーが似たようなアクセス権を持っている場合、XFACILIT アクセス権をセットアップしないことを選択した場合にもコピーされた SDUMP データ・セットが引き続き作成され、代わりにすべてのユーザーにコピーされた SDUMP データ・セット・プロファイルに対する ALTER アクセス権が付与されます。この環境では、システム上のヒストリー・ファイルに対してすべてのユーザーが同じアクセス権限を持つ可能性があります。ただし、一部のユーザーが必ずしもすべてのヒストリー・ファイルに対する READ アクセス権を持つとは限らない場合、障害項目にリンクされているすべてのコピーされた SDUMP データ・セットに保護を拡張するために、データ・セット・プロファイルに対して UACC(NONE) を指定した XFACILIT プロファイルを使用することを考慮してください。

## コピーされた SDUMP データ・セットに対する XFACILIT リソース・クラスの使用

XFACILIT クラス・プロファイルを `IDISDUMP_HLQ.hlq.**` という名前を使用してセットアップします。*hlq* を、IDIOPTLM 構成オプション・モジュール内の SDUMPDSN オプションを使用して指定した、データ・セット名前パターンの 1 つ以上の修飾子で置き換えます。(詳しくは、[コピーされた SDUMP データ・セット名パターンの指定 \(SDUMPDSN\) ページ 342](#)を参照してください。)

高位修飾子に、例えば `SDUMP&SYSCLONE` などのシンボル名が含まれていると、予期されたシンボル置換値に応じて、複数のプロファイルを設定アップする必要がある場合があります。

XFACILIT プロファイル (または、シンボル置換により複数のプロファイル) が定義されている場合は、関係するユーザーに対して該当するレベル (ALTER または NONE) を指定します。XFACILIT クラスに対する ALTER アクセス権のあるユーザーは、高位修飾子がシンボル置換後に XFACILIT プロファイル名 *hlq* 値と一致するコピーされた SDUMP データ・セットに対して、Fault Analyzer を通して作成機能を暗黙的に所持します。

XFACILIT プロファイルへの汎用 ALTER アクセス権では、コピーされた SDUMP データ・セットを保護する通常データ・セット・プロファイルをオーバーライドしません。Fault Analyzer を使用してアクションを実行するときに、障害項目にリンクされているコピーされた SDUMP データ・セットに対して必要なアクセス権を許可するのみです。このアクションには以下のようなものがあります。

- 再分析時のデータ・セットの読み取り
- 関連障害項目の削除時のデータ・セットの削除

Fault Analyzer は以下の場合に、コピーされた SDUMP を作成し、障害項目にリンクしようとします。

- C または M 行コマンドのいずれかを使用して、障害項目を「障害エントリー・リスト」画面またはバッチ IDIUTIL IMPORT 制御ステートメントからコピーする場合
- 障害項目が既に元の SDUMP データ・セットと関連付けられている、または以前に SDUMP データ・セットに関連付けられている場合

コピーした SDUMP データ・セットを作成およびリンクするには、適切な XFACILIT `IDISDUMP_HLQ` プロファイルか、またはコピーした SDUMP データ・セット・プロファイルに対して ALTER アクセス権が付与されている必要があります。

Fault Analyzer は、ユーザーが問題分析を実行しており、障害項目に対する READ アクセス権または DELETE アクセス権を持っている場合、コピーした SDUMP データ・セットに対して、関連する障害項目と同等のアクセス権を提供します。

障害項目を暗黙的に削除すると、障害項目にリンクされている関連するすべてのコピーされた SDUMP データ・セットも削除されます。

### 例

#### XFACILIT の例: コピーされた SDUMP データ・セット

以下は、Fault Analyzer によりコピーされた SDUMP データ・セットを管理するための、XFACILIT クラスの推奨セットアップ例です。この例は必要に応じて変更したり、それを基に拡張したりできます。

この例では、IDIOPTLM 構成オプション・モジュールの SDUMPDSN オプションが、以下の値で指定されていると想定しています。

```
' IDIHLQ.SDUMP.&&SYSNAME..D&&YYMMDD..T&&HHMMSS..S&&SEQ.'
```

1. XFACILIT プロファイルを定義して、ALTER 汎用アクセス権限をこのプロファイルに付与します。

```
RDEFINE XFACILIT IDISDUMP_HLQ.IDIHLQ.SDUMP.** UACC(ALTER)
```

2. NONE 汎用アクセス権限を持つ同じデータ・セットの総称データ・セット・プロファイルを定義します。

```
ADDSD 'IDIHLQ.SDUMP.**' UACC(NONE)
```

## Fault Analyzer の IFAPRDxx parmlib メンバーへの登録

Fault Analyzer を製品コード 5755-A01 IBM Application Delivery Foundation for z/OS の一部として購入した場合、IFAPRDxx parmlib メンバーに次のような項目を含めます。

```
PRODUCT OWNER('IBM CORP')
  NAME('IBM APP DLIV FND')
  ID(5755-A01)
  VERSION(*) RELEASE(*) MOD(*)
  FEATURENAME('FAULT-ANALYZER')
  STATE(ENABLED)
```

Fault Analyzer を別個に購入した場合、IFAPRDxx parmlib の項目は必要ありません。ただし、項目を組み込みたい場合は、以下のように追加できます。

```
PRODUCT OWNER('IBM CORP')
  NAME('FAULT ANALYZER')
  ID(5755-A02)
  VERSION(*) RELEASE(*) MOD(*)
  FEATURENAME('FAULT ANALYZER')
  STATE(ENABLED)
```

VERSION、RELEASE および MOD を除くすべてのパラメーターは、示されている通り正しく指定する必要があります。構文上有効な値を VERSION、RELEASE および MOD に指定できます。

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

```
SET PROD=xx
```

IFAPRDxx parmlib メンバーに関する一般情報については、『MVS™ 初期設定およびチューニング解説書』を参照してください。

### 追加の Fault Analyzer IFAPRDxx 処理

上記の製品が IFAPRDxx に定義されていない場合は、Fault Analyzer が最初に起動されたとき、初期設定時に製品コード 5755-A02 として登録されます。

上記の製品が STATE(DISABLED) または STATE(NOTDEFINED) で定義されている場合、製品は登録用には選択されません。

Fault Analyzer が実行されないようにするには、次の IFAPRDxx 項目を使用します。

```
PRODUCT OWNER('IBM CORP')  
NAME('FAULT ANALYZER')  
ID(5755-A02)  
VERSION(*) RELEASE(*) MOD(*)  
FEATURENAME('FAULT ANALYZER')  
STATE(DISABLED)
```

製品の購入方法を変更した場合 (例えば、スタンドアロンのバージョンの使用から IBM Application Delivery Foundation for z/OS 製品バンドルにアップグレードした場合など)、新しい定義をアクティブ化する際に、z/OS® で保持されているストレージ内の表から既存の製品定義を明示的に削除する必要があります。これは、以下のシナリオに従って、動的に行います (IPL なし)。

1. IFAPRDxx で、上記のように新しい製品を定義し、古い製品を STATE(DISABLED) を指定して更新します。
2. 次のオペレーター・コマンドを使用して、更新をアクティブにします。

```
SET PROD=xx
```

3. その後、古い製品定義を IFAPRDxx から安全に削除できます。

IBM® では NAME(\*) または ID(\*) フィールドを含む IFAPRDxx 項目を定義しないことをお勧めします。意図しない製品の登録が行われる可能性があるからです。STATE(ENABLED) の NAME(\*) および ID(\*) で定義された項目で一致が検出された場合、Fault Analyzer は製品コード 5755-A02 として登録されます。

## 第 14 章. Fault Analyzer IDIS サブシステムの使用

サービスを他の方法では実行できない場合や、サービスを開始しないと分析が不完全なものになる可能性がある場合、Fault Analyzer は IDIS というサブシステムを使用します。

MVS™ を実行する IDIS イメージごとに個別の Fault Analyzer サブシステムが必要です。

Fault Analyzer は、IDIS サブシステムを使用して、以下のことを行います。

- DB2® サブシステムに接続し、アドレス・スペースが異常終了して接続が失敗する場合にカタログを読み取ります。これらの接続の失敗により、次のメッセージがアドレス・スペースの異常終了で出されます。

```
IDI0082E DB2 Call Level Interface error: SQL return code -1 for SQLAllocConnect  
to DB2 system system-id
```

- IDIXTSEL ポスト・ダンプ出口の代わりに SVC ダンプを登録します。この登録は主に CICS® システム・ダンプを対象としており、Java™ 障害の取り込みを容易にすることを目的としていますが、SLIP ダンプの取り込みも可能です。
- PDSESHARING(NORMAL) が使用される場合は、異常終了 S213-70 のシステム間共有の競合を減らすために、ヒストリー・ファイル・アクセス管理を行います。
- ヒストリー・ファイル \$\$INDEX メンバーを管理すると、より高いパフォーマンスを得ることができます (オプション)。この機能は適格履歴ファイルのデフォルトですが、[IDIS サブシステムの始動 ページ 321](#)で説明されているように、IDIS サブシステムに NOUPDINDEX PARM フィールド・オプションを指定して使用不可にすることができます。詳しくは、[ヒストリー・ファイル \\$\\$INDEX データのキャッシュ ページ 320](#)を参照してください。
- NoDup(ImageFast(*minutes*,IMS(...))) オプションを使用して指定した IMS fast duplicate fault suppression を有効化します (オプション)。この機能はデフォルトですが、[IDIS サブシステムの始動 ページ 321](#)で説明されているように、IDIS サブシステムに NOUPDINDEX または NOIMAGEFAST PARM フィールド・オプションを指定して使用不可にすることができます。NoDup(ImageFast(*minutes*,IMS(...))) オプションについて詳しくは、[NoDup ページ 611](#)を参照してください。
- 高速 Exclude オプション処理を使用可能にします (オプション)。この機能はデフォルトですが、[IDIS サブシステムの始動 ページ 321](#)で説明されているように、IDIS サブシステムに NOFASTEXCLUDE PARM フィールド・オプションを指定して使用不可にすることができます。詳しくは、[高速 Exclude オプション処理 ページ 375](#)を参照してください。
- リカバリー障害記録サポートを提供します。(リカバリー障害記録の一般情報については、[リカバリー障害記録 ページ 57](#)を参照してください。)
- 情報をキャッシングして保管することにより、Fault Analyzer ISPF インターフェース LOOKUP コマンドでメッセージおよび異常終了コードの説明を検索するときのパフォーマンスを向上させます。さらに、必要に応じて、VSAM KSDS メッセージおよび異常終了コードの説明リポジトリを自動的に更新します。必要なアクセス許可については、[メッセージおよび異常終了コード説明リポジトリのセットアップ ページ 309](#)を参照してください。
- Java 提供の Diagnostic Tooling Framework for Java™ (DTFJ) を使用して、Java™ 分析サポートを提供します。DTFJ プロセスは、PARM='JAVA' オプションが IDIS サブシステム始動 JCL で使用されているときに IDIS サブシステムから spawn される、BXPAS アドレス・スペースから実行します。

- ソース・データ・セットからメッセージと異常終了の説明の読み取りを実行し、パフォーマンス向上のために、選択した説明をキャッシュに入れます。
- マイグレーション済みの密結合ダンプ・データ・セットを、それと関連する障害項目を削除するときと一緒に削除します。[移行された密結合の関連ダンプ・データ・セットを削除するための IDIS サブシステム要件 ページ 326](#) を参照してください。

IDIS サブシステムの優先順位は、このサブシステムが呼び出される可能性のあるいずれのタスクよりも低く設定しないでください。IDIS サブシステムに高い優先順位を割り当てたとしても、IDIS サブシステムは未使用時にはリソースを消費しないため、システム・パフォーマンスに悪影響を及ぼすことはありません。

## OMVS ホーム・パス

IDIS サブシステムは、OMVS UID 値およびホーム・パスを使用してセキュリティー・システムに定義する必要があります。ホーム・パスは適切なアクセス権を有している必要があります。

## シスプレックス全体にわたるサブシステムの相互通信

シスプレックス内の複数の IDIS サブシステムは、システム間カップリング・ファシリティ(XCF)を使用して互いに連絡するので、\$\$INDEX データのキャッシングなど、サブシステム管理データの効率的な共用が可能になります。

IDIS サブシステムは、その整合性と共有方法の一部として `ISGQUERY GATHERFROM=SYSPLEX` を使用します。IBM 以外のエンキュー管理製品を使用しているお客様は、ベンダーに連絡して、この機能をサポートするために使用可能なセットアップ要件を検討してください。

それ以外の場合は、この機能を使用するために特別なアクションは必要ありません。XCF が使用不可になった場合でも、IDIS の処理ではすべての機能が引き続き実行されます。ただしこの場合、結果としてパフォーマンスが低下する可能性があります。

## ヒストリー・ファイル \$\$INDEX データのキャッシュ

Fault Analyzer IDIS サブシステム始動 JCL にある PARM パラメーターのデフォルト値は `"UPDINDEX"` です ([IDIS サブシステムの始動 ページ 321](#) を参照)。デフォルト値が使用された場合、IDIS サブシステムは、以下のすべての PDSE ヒストリー・ファイルの \$\$INDEX メンバー・アクセスを管理します。

- IDIS サブシステムが実行されている MVS™ イメージで使用されるファイル。
- IDIS サブシステムが UPDATE アクセス権を持つファイル。

\$\$INDEX メンバーの詳細については、[ヒストリー・ファイル・データ・セット内の特殊メンバー ページ 26](#) を参照してください。PDS フォーマットのヒストリー・ファイルは、エンキューおよび並列更新制限によって \$\$INDEX メンバーの効果的なキャッシュが妨げられているため、IDIS サブシステムで管理されません。


ヒストリー・ファイルから \$\$INDEX メンバーが読み取られた後、情報はアクティビティーが高い期間中、IDIS サブシステム・アドレス・スペースに保持されます。この方法は入出力を必要としないため、このデータの要求側は高速アクセスが可能になります。リアルタイム分析、対話式再分析とバッチ再分析、および Fault Analyzer ISPF インターフェースでは、キャッシュされた IDIS サブシステム・データが使用されます。



読み取りまたは書き込みでキャッシュにアクセスするたびに、ストレージ内の保存期間はリセットされます。このリセットにより、使用頻度の高いアクティブな履歴・ファイルでも、高速なキャッシュ・アクセスが維持されます。この方法は、複数の異常終了が短期間に連続して起こる可能性があり、同じ履歴・ファイルをたびたび更新する必要がある、CICS® などの環境で有効です。

IDIS サブシステムのストレージ内保存の制限時間は 5 分に設定されています。以下の場合に、\$\$INDEX メンバーは履歴・ファイルに書き戻され、IDIS サブシステムはその制御を解放します。

- 制限時間を超過した。
- 同じ履歴・ファイルの更新要求が同じシスプレックス内の別の MVS™ イメージから保留されている。


 **ヒント:** シスプレックス内のすべての IDIS サブシステムで、IDIS サブシステムの PARM='UPDINDEX' オプションまたは PARM='NOUPDINDEX' オプションの設定を同一にします。シスプレックスの履歴・ファイルの共有については、[シスプレックス全体での履歴・ファイルの共有 ページ 364](#)を参照してください。

UPDINDEX オプションを使用すると、履歴・ファイル \$\$INDEX メンバーが逐次化されるため、Fault Analyzer の実行で特定の低優先度ジョブが過度に遅延するのを回避できます。これらのジョブは、CPU に制約のある環境で、パフォーマンスの影響を受けやすいジョブ、または IMS™ や CICS® などの実行環境と、履歴・ファイルを共用するジョブです。

UPDINDEX オプションを使用する場合は、処理することが期待されているすべての履歴・ファイルに対して、通常のセキュリティ・サーバー・データ・セット・プロファイルを介して IDIS サブシステムが UPDATE アクセス権を持っていることを確認してください。XFACILIT アクセス権では不十分です。

IDIS サブシステムが UPDATE アクセス権を持たない履歴・ファイルを処理するために呼び出されたか、または何らかの理由で履歴・ファイルの更新に失敗すると、次のいずれかの手順が実行されるまで、その履歴・ファイルの \$\$INDEX メンバーを管理する試みはこれ以上実行されません。

- IDIS サブシステムは停止して再始動されます。
- 対話型 IDIS サブシステム・インターフェースを使用して、履歴・ファイルの「除外」状況をリセットします。詳しくは、[対話式 IDIS サブシステム・インターフェースの使用 ページ 147](#)を参照してください。

 **注:** 最大の Fault Analyzer パフォーマンスを得るには、DeferredReport オプションを検討してください。詳しくは、[DeferredReport ページ 584](#)を参照してください。

## IDIS サブシステムの始動

Fault Analyzer IDIS サブシステムを開始するために、以下のような単純なジョブを実行依頼できます。

```
//IDISS JOB ...
//IDISSTST EXEC PGM=IDISAMAN,TIME=NOLIMIT,REGION=region-size,PARM='options'
//IDIDOC2 DD DISP=SHR,DSN=IDI.SIDIDOC2
//* (Optional DD statements might follow, as described below)
```

各部の意味は以下のとおりです。

**REGION=region-size**

IDIS サブシステムに使用する領域サイズを指定します。

ほとんどの場合、領域サイズは 100 MB で十分です (REGION=100M)。しかし、IDIS サブシステムで非常に多くの履歴・ファイルを管理する場合、または履歴・ファイルのサイズが非常に大きい場合は、さらに大きい領域サイズを指定する必要があります。必要な領域サイズを見積もる方法については詳しくは、[IDIS サブシステム・ストレージ要件 ページ 324](#)を参照してください。

**PARM='options'**

IDIS サブシステムが一部のサブシステム機能を使用不可にするためにのみ使用する特殊オプションを指定します。サブシステム内のその他のオプション処理は、標準 IDICNFxx parmlib メンバーおよび IDIOPTS DD ステートメントを使用して行われます。[オプション ページ 567](#)の説明を参照してください。オプションの PARM フィールドの指定には、*options* に以下のいずれかの値を含めることができます。

**UPDINDEX**

IDIS サブシステムが実行されていて IDIS サブシステムが UPDATE アクセス権を持つイメージと同じ MVS™ イメージで使用される、すべての PDSE 履歴・ファイルの \$\$INDEX メンバー・アクセスを、IDIS サブシステムが管理するよう指定します。詳しくは、[履歴・ファイル \\$\\$INDEX データのキャッシュ ページ 320](#)を参照してください。

この値はデフォルトです。

**NOUPDINDEX**

異常終了ジョブですべての履歴・ファイルの更新を実行することを指定します。



**注:** NOUPDINDEX を指定すると、Fault Analyzer リカバリー障害記録 (Recovery Fault Recording - RFR) 機能が無効になります。

**IMAGEFAST**

NoDup(ImageFast(*minutes*,IMS(...))) オプションを使用して指定された IMS fast duplicate fault suppression を有効にします。詳しくは、[NoDup ページ 611](#)を参照してください。

この値はデフォルトです。

**NOIMAGEFAST**

NoDup(ImageFast(...)) の設定に関係なく、IMS fast duplicate fault suppression を使用不可にします。

**FASTEXCLUDE**

高速 Exclude オプション処理を使用可能にします。詳しくは、[高速 Exclude オプション処理 ページ 375](#)を参照してください。

この値はデフォルトです。

**NOFASTEXCLUDE**

IDIDA タスクによる通常の Exclude 処理に戻すために、高速 Exclude オプション処理を使用不可にします。

**XCFGRPSUFFIX=c**

代替の XCF メッセージ・グループを作成するため、IDIS サブシステム XCF メッセージング・グループ名 IDISXCFc の最後の文字 c を制御します。この作成は通常、DASD とヒストリー・ファイルをメイン IDISXCFM デフォルト・グループと共用しない MVS™ イメージがシスプレックス内にある場合にのみ行われ、DASD とヒストリー・ファイルを共用しないイメージに対して設定されます。各メッセージング・グループは、XCF メッセージング・グループを使用して、データ・セット名によりヒストリー・ファイルに対する更新を共有します。

**NOXCFGRPSUFFIX**

デフォルトの「M」サフィックスを使用します。この値はデフォルトです。

**JAVA**

Java™ 分析を使用可能にします。詳しくは、[IDIS Java のサブシステム要件 ページ 325](#)を参照してください。

**NOJAVA**

Java™ 分析を使用不可にします。この値はデフォルトです。

**SLIP**

SLIP ダンプの取り込みを有効にします。

**NOSLIP**

SLIP ダンプの取り込みを無効にします。この値はデフォルトです。

PARM フィールド・オプションを複数指定する場合は、1 つ以上のブランク文字で区切ってください。

あるいは、IDIS サブシステムは開始済みタスクを使用して設定できます。IDIS サブシステムは、データ・セットを SYSOUT=\* に動的に割り振るため、ジョブ入力サブシステム (JES) で実行されている必要があります。



**注:** IDIS サブシステムが S522 で異常終了しないよう、上記の例のように TIME=NOLIMIT パラメーターが指定されていることを確認してください。

Fault Analyzer によって使用されるこのサブシステムのサブシステム名は IDIS です。この名前は、IDISAMAN プログラムにより動的に定義されるため、IEFSSNxx parmlib メンバーに定義されている必要はありません。

//IDIDOC2 DD ステートメントは IDI.SIDIDOC2 SMP/E ターゲット・データ・セットの割り振りに使用されます。これには、Fault Analyzer VSAM KSDS メッセージおよび異常終了コード説明リポジトリーに対する更新が含まれていることがあります。更新が使用可能である場合に、//IDIDOC2 DD ステートメントが指定されていなかったり、IDI.SIDIDOC2 データ・セットの高位修飾子が、DataSets(IDIDOC(...)) オプションによって識別される IDI.SIDIDOC1 データ・セットの高位修飾子と

同じでなかったりするときは、メッセージ [IDI0165A ページ 733](#) が発行されます。この状態が発生した場合は、上に示したように //IDIDOC2 DD ステートメントを追加してください。

IDIS サブシステムがアクティブであるときに、LE HEAPZONES オプション設定を HEAPZONES(0,...) 以外に変更しないでください。

## IDIS サブシステム・ストレージ要件

PARM='NOUPDINDEX' を使用する場合、REGION サイズの指定に要件はありません。デフォルトの 32 MB の領域で十分です。

PARM='UPDINDEX' (この値がデフォルトです) を使用する場合は、より大きい領域サイズを使用する必要があります。メガバイト単位での領域サイズは以下のように概算できます。

```
32 + (num_hist_files * (0.5 + (avg_num_entries * 0.0005)))
```

各部の意味は以下のとおりです。

### num\_hist\_files

IDIS サブシステムで管理される PDSE ヒストリー・ファイルの総数。

### avg\_num\_entries

管理対象ヒストリー・ファイル内の障害項目の平均数。

IDIUTIL バッチ・ユーティリティーの SetMaxFaultEntries

制御ステートメントを使用してヒストリー・ファイルに障害項目の最大数が設定された場合

(詳細については [ヒストリー・ファイルの管理 \(IDIUTIL ユーティリティー\) ページ 441](#) を参照)、この数がそのヒストリー・ファイルに含まれる障害項目の数になります。

## IDIS DB2 のサブシステム要件

以下の場合、LINKLIST でアクセスできないすべての DB2® サブシステム、および Fault Analyzer で分析を行うときの対象となるすべての DB2 サブシステムに関して、DD ステートメントを追加する必要があります。

- 複数のバージョンの DB2® がインストールされている場合。
- DB2® ロード・モジュール・ライブラリーが LINKLIST にない場合。

DD ステートメントは次のようなフォーマットにします。

```
//DB2subsystem-id DD DISP=SHR,DSN=data-set-name
```

ここで、*subsystem-id* は DB2® サブシステム ID (通常 4 文字)、*data-set-name* は関連するロード・モジュール・ライブラリーを指します。データ共有グループの場合は、DB2® サブシステムがデータ共有モードで稼働しているかどうかに関係なく、グループに指定された名前がサブシステム ID として使用されます。

例えば、ID DSN1 を持つ DB2® サブシステムに、LINKLIST に含まれないロード・ライブラリー DSN1.SDSNLOAD が必要な場合は、次の JCL DD ステートメントを Fault Analyzer IDIS サブシステム・ジョブに追加します。

```
//DB2DSN1 DD DISP=SHR,DSN=DSN1.SDSNLOAD
```

この DDname は必要に応じて、IDIS によって適切な DB2® サブシステムに動的に割り当てられるため、Fault Analyzer サブシステム JCL に DSNAOINI DD ステートメントを含めないでください。

IDIS サブシステムには、DSNACLI プランに対する EXECUTE 権限、および以下の SYSIBM カタログ・テーブルに対する SELECT 権限が必要です。

```
SYSIBM.SYSDBRM
SYSIBM.SYSPACKAGE
SYSIBM.SYSPACKSTMT
SYSIBM.SYSPACKLIST
SYSIBM.SYSPLAN
SYSIBM.SYSSTMT
```

## IDIS Java のサブシステム要件

Java™ 障害項目を作成するには、少なくともデフォルトのヒストリー・ファイルに対する UPDATE アクセス権限が IDIS サブシステムに必要です。

## IDIJLIB DD ステートメントの指定

オプションの IDIJLIB DD ステートメントを Java™ 用の IDIS サブシステム JCL に以下のように含めることができます。

```
//IDIJLIB DD PATH='path'
```

IDIJLIB JCL ステートメントで、HFS 実行可能ファイルのターゲット・ディレクトリーを指定します。IDIS サブシステムは、この実行の一環としてこのディレクトリーに少数のプログラム・ファイルを書き込むため、IDIS サブシステムは、指定されたディレクトリー内のファイルの読み取り権限、書き込み権限、および実行権限を持っている必要があります。さらに、診断情報も、このパスに書き込まれる場合があります。パス名は大/小文字が区別され、パスが存在する必要があります。可能な指定例として、`PATH='/u/user-id/idi j'` があります。ここで、`user-id` は IDIS サブシステムを実行するユーザー ID です。



**注:** 指定されている場合、IDIJLIB パスは移行された密結合の関連ダンプ・データ・セットの削除にも使用されます。詳しくは、[移行された密結合の関連ダンプ・データ・セットを削除するための IDIS サブシステム要件 ページ 326](#)を参照してください。

IDIJLIB が提供されない場合、IDIS サブシステムのユーザー ID のデフォルト・パスが使用されます。このケースでは、IDIS サブシステムのユーザー ID に有効なデフォルト・パスが存在しない場合、メッセージ [IDIO155W ページ 730](#) が発行されます。

シスプレックス内のイメージごとに Fault Analyzer は、&SYSCClone 名を使用して IDIJLIB パスのサブディレクトリーを作成します。これにより、各イメージが個別のディレクトリーに書き込むようになります。

## Java ダンプ分析で使用するデフォルトの JVM の指定

Java™ ダンプ分析を実行するには、Java™ ダンプを記録した JVM と同等以上のサービス・レベルを持つ JVM が必要です。オプションの DD ステートメントでは、ダンプ Java™ パスで示された JAVA\_HOME のバージョンがシステムで見つからない場合に使用するデフォルトの JVM を指定できます。

IDIJVM および IDIJVM6 DD ステートメントを Java™ 用の IDIS サブシステム JCL に追加することで、デフォルトの JVM を指定できます。

- IDIJVM DD ステートメントでは、デフォルトの 31 ビット JVM へのパスを指定します。
- IDIJVM6 DD ステートメントでは、デフォルトの 64 ビット JVM へのパスを指定します。

```
//IDIJVM DD PATH='path'
//IDIJVM6 DD PATH='path'
```

## 移行された密結合の関連ダンプ・データ・セットを削除するための IDIS サブシステム要件

移行された密結合の関連ダンプ・データ・セットの削除は、spawn された BPX バッチ領域を介して実行されます。これは、潜在的なパフォーマンス上の問題を回避するために行われます。

IDIS サブシステムは、次の順序で決定されたパスに「ダミー」ファイルを作成し、その後削除します。

1. IDIJLIB DD ステートメントで指定されたもの (詳細については、[IDIJLIB DD ステートメントの指定 ページ 326](#)を参照)。
2. IDIS サブシステムのユーザー ID 用の OMVS セグメント HOME パス ([OMVS ホーム・パス ページ 320](#)を参照)。
3. /tmp

これらのパスのいずれかを使用できなかった場合、[メッセージ IDI0192I が発行されます ページ 739](#)。

## IDIJLIB DD ステートメントの指定

オプションの IDIJLIB DD ステートメントを IDIS サブシステム JCL に以下のように含めることができます。

```
//IDIJLIB DD PATH='path'
```

ここで、パスは Fault Analyzer が必要なダミー・ファイルを一時的に作成する場所です。



**注:** 注: パスが指定されている場合、IDIJLIB パスは Java 分析の HFS 実行可能ファイルのターゲットとしても使用されます。詳しくは、315 ページの「Java の IDIS サブシステム要件」を参照してください。

## IDIS サブシステムの停止

IDIS サブシステムはいつでも停止して再始動できます。

IDIS サブシステムが非アクティブ (停止) の場合、新しい障害 ID の作成などの基本的な Fault Analyzer プロセスだけが引き続き実行されます。SVC ダンプ登録、リカバリー障害記録 (RFR)、Java DTFJ 処理、一部の DB2® カタログ・データ・アクセス、移行された密結合のダンプ・データ・セットの削除は機能しません。IDIS サブシステムが PARM='UPDINDEX' で稼働しているときと比較して、CPU および入出力の使用量も増加します。

**!** **重要:** 決して CANCEL コマンドを使用して IDIS サブシステムを停止しないでください。

正常終了を可能にするために、MODIFY コマンドを使用して IDIS サブシステムを停止します。例: `F name,STOP` または `P name`

*name* は MODIFY コマンドの適切な ID を指します。この ID は、IDIS サブシステムの始動方法によって異なります。

IDIS サブシステムがアクティブでない場合、または MODIFY コマンドに不正確な ID が使用された場合、MVS™ は次のメッセージを発行します。

```
IEE341I XYZ                NOT ACTIVE
```

IDIS サブシステムが既にアクティブな場合に、それを始動しようとする、オペレーター・コンソールに次のメッセージが発行されます。

```
IDISAMAN The Fault Analyzer Subsystem is already active in jobname job-id
```

*jobname* は現在実行している IDIS サブシステムのジョブまたは開始済みタスク名、*job-id* は JES ジョブまたは開始済みタスク ID です。

## 第 15 章. ISPF 環境の変更

Fault Analyzer を ISPF で使用するには、適切なデータ・セットを割り振り、Fault Analyzer を起動する 1 つ以上の方法を提供しておく必要があります。このプロセスについて、以下のトピックで説明します。

### ISPF データ・セットの割り振り

以下のデータ・セットを、個々の ISPF DDname に割り振る必要があります (TSO ログオン手順、またはその他のインストール・システム固有の方法を使用します)。

DDname	データ・セット名
ISPLLIB	IDI.SIDIPLIBIPV.SIPVPENU (1)
ISPLMLIB	IDI.SIDIMLIBIPV.SIPVMENU (1)
ISPSLIB	IDI.SIDISLIB
ISPTLIB	IDI.SIDITLIBIPV.SIPVTENU (1)
SYSEXEC	IDI.SIDIEEXEC



**注:** (1) データ・セットは、ADFz Common Components のインストールの一環として作成されます。

ISPF 内から Fault Analyzer を呼び出すために使用できる REXX EXEC のサンプルは、メンバー IDISISPF としてデータ・セット IDI.SIDISAM1 に提供されます。EXEC は、ISPF LIBDEF および TSO ALTLIB サービスを使用して要求されたデータ・セットに必要な動的定義を実行します。



**注:** File Manager for z/OS® を使用してデータ・セットの編集またはブラウズを可能にするには、Fault Analyzer が起動されたときに、File Manager で必要なすべての ISPF ライブラリーも使用可能にする必要があります。TSO ログオン・プロシージャまたは起動 exec に追加すべき必要なデータ・セット名については、File Manager for z/OS® の資料を参照してください。IDISISPF サンプル exec は、必要に応じて File Manager for z/OS® データ・セットを含めるために使用できます。

Fault Analyzer へのデータ・セットの割り振りに関する問題の診断を支援するには、TSO/ISPF コマンドの ISRDDN、ISRFIND、または ISPLIBD の使用が適しています。

### Fault Analyzer IDISCMDS コマンド・テーブルを使用可能にする

Fault Analyzer は、サンプルのコマンド・テーブルをデータ・セット IDISCMDS 内のメンバー IDI.SIDITLIB として備えています。このコマンド・テーブルは、SFA コマンド、LOOKC コマンド、および LOOKUP コマンドを発行するために必要な定義を備えています。以下を参照してください。

- [SDSF からの Fault Analyzer の起動 \(SFA コマンド\) ページ 331](#)
- [カーソル選択を使用した LOOKUP コマンドの起動 \(LOOKC コマンド\) ページ 331](#)
- [LOOKUP ページ 112](#)



IDISCMDS コマンド・テーブルは、ISPF に対して定義することによって使用可能にする必要があります。その方法については、「z/OS® ISPF 計画とカスタマイズ」の『DM のカスタマイズ』の章の『コマンド・テーブルのカスタマイズ』セクションを参照してください。

## ISPF 選択パネルの更新

ISPF 選択パネルを更新して、IDIPDDIR プログラムを選択するためのオプションを組み込み、IDI アプリケーション ID を使用して ISPF ヒストリー・ファイル・インターフェースを開始します。例えば、オプション 9 を使用して Fault Analyzer を起動するには、以下のように ISPF 選択パネルの )PROC セクションを更新します。

```
)PROC
&ZSEL = TRANS (TRUNC (&ZCMD, '.'))
.
. (other selections)
.
9, 'PGM(IDIPDDIR) NEWAPPL(IDI) SCRNAME(FAULTA) PARM(HEAPZONES(0,ABEND)/)'
X,EXIT
' ',' '
*, '?' )
&ZTRAIL=.TRAIL
```

### 注記:

1. 上記の IDI というアプリケーション ID は、例としてのみ示してあります。以前のバージョンの Fault Analyzer で別の値が使用されていた場合、値を変更すると ISPF インターフェースのすべてのユーザー調整が失われるため、変更しないでください。
2. Fault Analyzer がアクティブである間は、LE HEAPZONES パラメーターが HEAPZONES(0,ABEND) に設定されていることを確認します。

プログラム IDIPDDIR を上述のように選択パネルから直接呼び出す代替方法として、代わりに IDISISPF サンプル REXX EXEC を呼び出すことを検討できます。この呼び出しにより、Fault Analyzer の ISPF ライブラリーおよび TSO ライブラリーを ISPF の起動前に割り振る必要がなくなります。

)BODY セクションも適切に更新してください。

ISPF 選択パネルから Fault Analyzer ISPF インターフェースを直接呼び出すための代替手段として、カスタマイズされたフロントエンドを使用できます。[カスタマイズ済み ISPF インターフェース・フロントエンドの例 ページ 741](#)にはサンプル・フロントエンドが示されており、これはインストール時に発生する可能性のある、特定の要件に適合するよう変更することができます。

## ISPF 3.4 の行コマンドを使用した Fault Analyzer の起動

ISPF 選択メニューのオプションを使用した、Fault Analyzer ISPF インターフェースの標準的な起動の代替方法があります。ISPF オプション 3.4 「データ・セット・リスト・ユーティリティ」パネルで、MVS™ ダンプ・データ・セット、ヒストリー・ファイル、または CICS® 補助トレース・データ・セットに対する行コマンドとして、Fault Analyzer ISPF インターフェースを直接起動するほうが便利な場合があります。この呼び出しには、選択したデータ・セットを使用して、対話

式ダンプ分析の実行、障害項目の表示、または CICS® トレース情報の表示を即時に行うことができるという利点があります。ダンプ・データ・セットの「File」プルダウン・メニューを使用して当該アクションを開始したり、履歴・ファイルのデータ・セット名を入力したりする必要はありません。

これを容易にするために、ISPF ユーザーの SYSEXEC 連結に割り振られたデータ・セットの 1 つで、(例えば、「障害分析」の FA という名前の) REXX exec が必要です。この exec 名は、ISPF オプション 3.4 のデータ・セット・リストで、ダンプ・データ・セット名、履歴・ファイル・データ・セット名、または CICS® 補助トレース・データ・セット名の横に入力できます。exec は、正しい ISPF NEWAPPL ID を使用して Fault Analyzer を呼び出し、パラメーターとしてデータ・セットの名前を渡します。この exec は、その割り振り属性からデータ・セット・タイプを判別します。

FA exec は次のとおりです。

```
Parse Arg dsn .

If dsn = '' Then
  Do
    Say 'Data set name required.'
    Exit(4)
  End

/* Determine if dsn is a history file or a MVS DUMP.          */
outl. =
x = Outtrap('OUTL.', "NOCONCAT")
Address TSO "LISTDS " || dsn
x = Outtrap(OFF)

svcdump = 0
auxtrac = 0
If outl.0 > 2 Then
  Do
    lrecl = Word(outl.3,2)
    If lrecl = 4160 Then
      svcdump = 1
    Else
      If lrecl = 4096 Then
        auxtrac = 1
      End
  End

/* Invoke the Fault Analyzer ISPF interface passing the appropriate */
/* PARM string.                                                    */
/* NOTE: The HEAPZONES(0,ABEND) parameter is only required if this */
/* is not the default Language Environment option in effect.      */
/* However, this parameter can be left as is regardless, without any */
/* adverse side effects.                                          */
Address ISPEXEC
If svcdump = 1 Then
  'SELECT PGM(IDIPDDIR) NEWAPPL(IDI) '!!',
  'PARM(HEAPZONES(0,ABEND)/DSN('dsn'))'
Else If auxtrac = 1 Then
  'SELECT PGM(IDIPDDIR) NEWAPPL(IDI) '!!',
  'PARM(HEAPZONES(0,ABEND)/AUXTRACEDSN('dsn'))'
Else
  'SELECT PGM(IDIPDDIR) NEWAPPL(IDI) '!!',
```

```
'PARM(HEAPZONES(0,ABEND)/ISPFHISTDSN('dsn'))'  
Exit
```

この例は、データ・セット IDI.SIDIEEXEC にメンバー IDISFA として組み込まれています。

IDISFA は、IDI.SIDIEEXEC が SYSEXEC 連結に含まれている限り、ISPF 3.4 データ・セットに対する行コマンドとして機能しますが、この exec を SYSEXEC 連結内の別のデータ・セットに、もっと短い名前 (FA など) でコピーするほうが便利な場合があります。

## SDSF からの Fault Analyzer の起動 (SFA コマンド)

IDISSDSF という名前の exec がデータ・セット IDI.SIDIEEXEC 内に用意されています。この exec を使用すると、ISPF の下の SDSF でジョブまたは syslog を参照しているときに、メッセージ [IDI0003I ページ 696](#) から履歴・ファイル・データ・セット名と障害 ID を抽出し、その障害の対話式再分析のために Fault Analyzer を起動できます。

IDISSDSF exec は、Fault Analyzer IDIS ISPF コマンド・テーブル IDISCMDS 内の SFA という ISPF コマンドにマップされます。このテーブルは、ISPF から使用できるようになっている必要があります。詳しくは、[Fault Analyzer IDISCMDS コマンド・テーブルを使用可能にする ページ 328](#)を参照してください。

## カーソル選択を使用した LOOKUP コマンドの起動 (LOOKC コマンド)

IDISMLKP という名前の exec がデータ・セット IDI.SIDIEEXEC 内に用意されています。この exec を使用すると、現行カーソル位置にあるワードを判別し、そのカーソル・ワードを引き渡して Fault Analyzer LOOKUP コマンドを起動することができます。

IDISMLKP exec は、Fault Analyzer IDIS ISPF コマンド・テーブル IDISCMDS 内の LOOKC という ISPF コマンドへマップされます。このテーブルは、ISPF から使用できるようになっている必要があります。詳しくは、[Fault Analyzer IDISCMDS コマンド・テーブルを使用可能にする ページ 328](#)を参照してください。

## ISPF インターフェースのデフォルトを新規ユーザーに提供

Fault Analyzer ISPF インターフェースの新規ユーザーにインストールに固有のデフォルトを提供するには、次の処理を行います。

1. Fault Analyzer ISPF インターフェースを呼び出します。
2. 新規ユーザーがこのインターフェースを初めて使用する時に参照する可能性のある値にオプションを設定します。これらの値には、以下が含まれる場合があります。
  - 「Fault Entry List」画面で選択された最初の履歴・ファイルまたはビュー。または、FAISPFOPPTS(INITHIST(...)) オプションを使用して、必要なデフォルト値を入力します。詳しくは、[InitHist ページ 604](#)を参照してください。
  - Fault Analyzer プリファレンス (「Options」プルダウン・メニューから)。
  - バッチ再分析のオプション (「Options」プルダウン・メニューから)。または、FAISPFOPPTS(BATCHOPTS(...)) オプションを使用して、必要なデフォルト値を入力します。詳しくは、[BatchOpts ページ 120](#)を参照してください。
  - 対話式再分析のオプション (「Options」プルダウン・メニューから)。
  - ビューの設定 (「View」プルダウン・メニューから)。



**注:** Fault Entry List 画面に表示される列に対するインストール・システム全般のデフォルトは、ISPF プロファイル・メンバーのデフォルトを使用して定義するのではなく、IDICNFxx parmlib メンバーの FAISPFopts(HistCols(...)) オプションを指定して定義することをお勧めします。これらの列をさらに変更する可能性のあるユーザーは、他の方法では PF4 を押してインストール済み環境全体のデフォルトにリセットできないため、この方法が必要です。

3. Fault Analyzer ISPF インターフェースを終了します。
4. *applid* PROF メンバーを、ご使用の ISPF プロファイルのデータ・セットから、すべてのユーザーがそのユーザーの ISPTLIB 連結で使用できるデータ・セットにコピーします。ここで *applid* は、ご使用のインストールで Fault Analyzer に使用されるアプリケーション ID (例えば、IDI) です。

ユーザーがプロファイル・メンバーに含まれている変数に変更を加えるとすぐに ISPPROF DDname によって識別される各ユーザーのプライベートなプロファイル・データ・セットに即時に保管されます。その後、Fault Analyzer ISPF インターフェースの使用時には、このデータ・セットから読み込まれます。

## インストール固有のバッチ再分析 JCL 制御ステートメントを提供

ユーザーがヒストリー・ファイル項目のバッチ再分析を実行する際、Fault Entry List 画面で B 行コマンドを使用すると、Fault Analyzer は、そのユーザーが Batch Reanalysis Options 画面で指定したオプションをベースに適切な JCL ストリームを生成します。そのような生成されたジョブの JCL 制御ステートメントをいつでもインストールに追加できるよう、Fault Analyzer には、オプションのユーザー提供スケルトン・メンバーが提供されています。検索できた場合、生成された JCL ストリームの JOB カードの後に、このメンバーが即時に挿入されます。スケルトン・メンバーは、IDISJCTL という名前であり、ISPSLIB 連結から使用できる必要があります。存在しない場合や検索できなかった場合は、単に使用されません。

このメンバーには、JCL 制御カードを追加してデフォルトの出力行数を超える出力を可能にするといった使用方法が考えられます。例:

```
/** Override of installation default output line limit
/**JOBPARM LINES=300
```

Batch Reanalysis Options 画面上の **Job card style** は「S」に設定されているが、**Job Card Statements** はまったく指定されていないという場合、IDISJCTL メンバーを使用し、JOB カードと任意の追加制御カードの両方を提供できます。IDISJCTL メンバーに対する代替については、[BatchOpts ページ 120](#)を参照してください。

## Fault Analyzer レポートの表示域の拡大

ISPF では、幅が 80 を超える画面サイズがサポートされています。Fault Analyzer でより幅の広い画面を使用できるようにすると、オンライン・レポートの読みやすさが大幅に改善され、左右のスクロール・コマンドを使用する必要がなくなります。

手順は以下のとおりです。

1. [画面サイズ調整の技術的詳細 ページ 753](#)で設定されている技術要件を満たしていることを確認します。
2. 使用するエミュレーター設定 (27x132 など) を変更します。
3. ISPF にログオンします。
4. ISPF オプション 0 に移動します。任意のコマンド行に `SETTINGS` を入力すると、同じ画面が表示されます。
5. 画面フォーマットを DATA または MAX に設定します。DATA と MAX の動作の違いの説明については、関連する ISPF 資料を参照してください。

## 第 16 章. USERMOD を使用した Fault Analyzer のカスタマイズ

一部の Fault Analyzer カスタマイズ・タスクは、SMP/E USERMOD を使用しないと実行できません。その他のタスクは、構成オプション・モジュールを使用して実行できます。詳しくは、[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#) を参照してください。

### 呼び出される Fault Analyzer の有効化

プログラムの異常終了時に Fault Analyzer に制御が移るようにするには、いくつかの呼び出し出口をインストールする必要があります。

非 CICS 環境に必要となる 2 つの呼び出し出口をインストールすることについて以下で説明します。CICS® 環境の呼び出し出口について、詳細は[CICS 環境のカスタマイズ ページ 400](#)を参照してください。

### MVS 変更オプション/抑止ダンプ出口のインストール IDIXDCAP

IDIXDCAP は、IPL 時に自動的に選択された PARMLIB PROGxx メンバーに次を組み込むことで、IEAVTABX\_EXIT 動的出口ルーチンとしてインストールされます。

```
EXIT ADD EXITNAME(IEAVTABX_EXIT) MODNAME(IDIXDCAP)
```

次の IPL の前に IDIXDCAP 出口をアクティブにするには、次のオペレーター・コマンドを発行します。ここで、xx は PARMLIB PROGxx メンバーの接尾部に一致します。

```
SET PROG=xx
```

### 言語環境プログラム異常終了出口 (IDIXCEE または IDIXCE64) の有効化

Fault Analyzer は、MVS™ 変更オプション/抑止ダンプ出口に対して Fault Analyzer を呼び出す追加の方法として、Language Environment® (LE) 異常終了出口 (IDIXCEE または IDIXCE64) を必要とします。この出口を有効化するには、CEE.SCEESAMP データ・セットのサンプル・メンバーを使用して、Language Environment® の CEEEXTAN CSECT に出口を追加することが必要です。サンプル・メンバーの推奨を変更した後、次のコメントを適切な行で置き換えます。

```
<<< REPLACE THESE 2 LINES WITH A COPY OF CEEEXTAN  
AND OVERRIDE AS DESIRED >>>
```

#### AMODE 31:

CEEWDEXT サンプル・メンバーのコピーで、コメントを次の行で置き換えます。

```
CEEXAHD      ,User exit header  
CEEXART  TERMxit=IDIXCEE  
CEEXAST      ,Terminate the list
```

#### AMODE 64:

CEEWQEXT サンプル・メンバーのコピーで、コメントを次の行で置き換えます。

```
CEEXAHD AM=64 ,User exit header (AMODE 64)  
CEEXART  TERMxit=IDIXCE64  
CEEXAST      ,Terminate the list
```

CEEEXTAN リストに複数の出口を指定する場合 (例えば、他の類似のサード・パーティー製品も使用されている場合)、最初の出口には Fault Analyzer を指定する必要があります。

Fault Analyzer インストールの一環として MVS™ の IPL を実行する計画がない場合、この出口は、LE が LPA に配置されているかどうかにより、それぞれ次のようにアクティブにすることができます。

#### • LE が LPA 内にある場合

APPLY ジョブの出力から、USERMOD によって更新されたデータ・セット名とロード・モジュールを判定して、次のコマンドを発行します。

```
SETPROG LPA,ADD,MOD=(module-list),DSN=data-set-name
```

LPA に配置されていない更新済みロード・モジュールが 1 つ以上ある場合は、下の「LE が LPA 内がない場合」の説明に従って処理を続けてください。



**注:** CLPA オプションを使って IPL を実行する場合を除いて、次の IPL の後で、SETPROG コマンドを再発行することを忘れないでください。

#### • LE が LPA 内がない場合

LE が LPA 内がない場合は、LINKLIST 内にあるものと見なされます。出口をアクティブにするには、次のコマンドを発行します。

```
F LLA,REFRESH
```

影響を受けるロード・モジュールのみを選択的にリフレッシュする場合は、更新されたデータ・セットおよびロード・モジュール名の SMP/E APPLY 出力を確認し、LIBRARIES パラメーターと MEMBERS パラメーターを使用して、これらを CSVLLAx parmlib メンバーに追加し、コマンド `F LLA,UPDATE=xx` を発行する必要があります。あるいは、Data Set Commander for z/OS® を使用してリフレッシュを実行します。

LE 異常終了出口のインプリメントに関する一般情報については、「*Language Environment®* カスタマイズ」を参照してください。



#### CICS に関する注記:

1. サンプル・ジョブ CEEWCEXT でインストールされる AMODE 31 LE 出口 (CEECXTAN) の CICS® バージョンは、IDIXCEE 出口と非常によく似ています。両方ともインストールする場合は、正しい名前を使用していること、および同じ出口を 2 回インストールしていないことを重ねて確認してください。
2. CICS® のこの出口の AMODE 64 バージョンは現在ありません。

## 非言語環境プログラム・ランタイムを使用するアプリケーションの操作

以下の USERMOD は、非 LE アプリケーションにのみ適用できます。

## PL/I V2R3 アプリケーションからの Fault Analyzer 暗黙呼び出しの有効化(++IDISPLI/++IDISPLIA)

PL/I バージョン 2 リリース 3 の非 LE ランタイム・ライブラリーを使用するアプリケーションで発生する異常終了に対して、暗黙的に Fault Analyzer を起動させるには、IDISPLI または IDISPLIA USERMOD を適用する必要があります。どちらの USERMOD も、PL/I ランタイム・ロード・モジュール IBMBLIIA における ONCODE 処理を変更して、Fault Analyzer プログラム SNAP インターフェース (IDISNAP) を呼び出します。ただし、IDISPLI USERMOD は IDISNAP への呼び出しの後 PL/I に戻りますが、IDISPLIA USERMOD は故意に異常終了 U3001 を発行します。これにより、この USERMOD が、ロールバックが要求される可能性のある IMS™ などの環境に対しより適したものとなります。

LE ランタイムを使用する場合、この USERMOD は必要ありません。

この USERMOD を適用するには、サンプル・ジョブ IDISPLI または IDISPLIA を編集して実行依頼します。

## PL/I PLIDUMP(++IDISPDM) からの Fault Analyzer の常時起動

PL/I アプリケーションが PLIDUMP を [S] オプション指定で呼び出すことによってエラーを処理するように作成されている場合は、Fault Analyzer を起動できません ([アプリケーション処理のエラー条件 ページ 305](#)を参照)。この理由から、通常の PLIDUMP 処理の前に常に Fault Analyzer を使用して IDISNAP を起動するように、PL/I PLIDUMP 主制御ルーチンを変更する USERMOD が用意されています。

この USERMOD を適用するには、サンプル・ジョブ IDISPDM を編集して実行依頼します。

## CA-Panexec からのロード・モジュールの取得

Fault Analyzer は、CA-Panexec によって管理されるロード・モジュールの CSECT 情報を取得するために、これを可能にする出口をインストールする機能を提供します。通常、ロード・モジュールが CA-Panexec によって管理されると、IBM® バインド・プログラムが CA-Panexec 管理のライブラリーの使用を試みるため、IEW2717S および IEW2718S 入出力エラーが表示されます。その結果、Fault Analyzer は障害点のソース行情報を提供できません。

CA-Panexec 出口の機能は、バインド・プログラムが使用できる通常のロード・モジュール・フォーマットで、ロード・モジュールを CA-Panexec 管理のデータ・セットから一時データ・セットにコピーします。

この目的で出口をインストールするために、通常ブランクの 8 バイトから成るモジュール IDIDA の IDILMODX CSECT をザップして、バインド・プログラムの INCLUDE プロセスより先に呼び出されるロード・モジュール名を指定できます。IDILMODX にあるロード・モジュールは、MVS™ LOAD マクロでアクセス可能である必要があります。

IDILMODX 名の出口は、レジスター 1 のパラメーター・リストを受け、2 つの 8 バイト・フィールドへの 2 つのポインタをアドレッシングします。1 つ目のフィールドには検査されるロード・モジュール名 (バインド・プログラムが CSECT データの抽出に使用するロード・モジュール)、2 つ目の 8 バイト・フィールドには DD 名またはブランクが使用されます。DD 名がブランクではない場合、この DD は、バインド・プログラムによってロード・モジュール名の検索に使用されるということを示します。Fault Analyzer は通常、CICS® で実行されている場合にのみ、DD 名フィールドを DFHRPL に設定します。他のほとんどの場合、DD 名はブランクになります。これは、読み取られるモジュールからのデータ・セットがバインド・プログラムによって配置されるよう、Fault Analyzer がブランクに設定していることを示します。ブランクの DD には 'X'00' が含まれます。IDILMODX 名の出口は、2 番目のパラメーターを、バインド・プログラムが読み取るモジュールの新規



DD 名に更新したり、DD 名を変更しないままにすることが許可されています。ddname が更新されると、モジュールのバインド・プログラムの INCLUDE 呼び出しに使用されます。

CA-Panexec 出口の例は、データ・セット IDI.SIDISAM1 のメンバー IDIPANEX として使用可能です。

## 第 17 章. Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ

Fault Analyzer のカスタマイズには、SMP/E USERMOD を介してのみ行える側面 (詳しくは、[USERMOD を使用した Fault Analyzer のカスタマイズ ページ 334](#)を参照) があります。その他の側面は、IDIOPTLM 構成オプション・モジュールで行えます。

IDIOPTLM 構成オプション・モジュールは、ロード・モジュール・フォーマットの以下のデータ域で構成されている必要があります。

表 8. IDIOPTLM データ域

オフセット		タイプ	長さ	名前	説明
10 進	16 進				
0	(0)	CHAR	44	LEDSN	非推奨
44	(2C)	CHAR	44	CNFDSN	代替 parmlib データ・セットを IDICNFxx(CNFD SN) に指定 <a href="#">ページ 340</a> を参照してください。
88	(58)	CHAR	100	RFRDSN	デフォルト・リカバリー障害記録 IEATDUMP データ・セット名 (RFRDSN) の変更 <a href="#">ページ 341</a> を参照してください。
188	(BC)	BIN	4	CICSNOA	CICS XDUREQ グローバル・ユーザー出口 (CICSNOA) を使用して起動された場合の Fault Analyzer のアクションの変更 <a href="#">ページ 341</a> を参照してください。
192	(C0)	CHAR	44	SSCHKDSN	代替セキュリティー・サー

表 8. IDIOPTLM データ域 (続く)

オフセット		タイプ	長さ	名前	説明
10 進	16 進				
					バー・テスト・データ・セット名の指定 (SSCHKDSN) ページ 342 を参照してください。
236	(EC)	CHAR	100	XDUMPDSN	、デフォルトの拡張ミニダンプ (XDUMP) データ・セット名パターン (XDUMPDSN) の変更 ページ 342 を参照してください。
336	(150)	CHAR	100	SDUMPDSN	コピーされた SDUMP データ・セット名パターンの指定 (SDUMPDSN) ページ 342 を参照してください。
436	(1B4)	BIN	4	NOIPVOPT	IBM Application Delivery Foundation for z/OS Common Components オプションを無視 (NOIPVOPT) ページ 343 を参照してください。
440	(1B8)	BIN	1	PREVEXIT	前の出口が RC=8 の場合に IDIXDCAP リアルタイム分析を抑制 (PREVEXIT)

表 8. IDIOPTLM データ域 (続く)

オフセット		タイプ	長さ	名前	説明
10 進	16 進				
					ページ 343を参照してください。
441	(1B9)	BIN	15		予約済み。すべてのバイナリーがゼロである必要があります。

これらの設定を指定するには、ジョブを編集して実行依頼し、IDIOPTLM 構成オプション・ロード・モジュールを作成します。サンプル・ジョブが、メンバー IDIOPTLM としてデータ・セット IDI.SIDISAM1 に提供されます。通常のオプション処理を行うときは、IDIOPTLM ロード・モジュールが読み込まれます。

構成オプション・ロード・モジュールの名前は IDIOPTLM でなければならず、IDI.SIDIAUTH または別の APF 許可ライブラリーに配置しなければなりません。また、このライブラリーを LNKLST に入れて、IDIOPTLM ロード・モジュールをすべてのジョブで検出可能にする必要があります。

## 代替 parmlib データ・セットを IDICNFxx(CNFDSN) に指定

SYS1.PARMLIB (または論理 parmlib 連結内のデータ・セットのいずれか) への汎用 READ アクセス権を提供しないインストールを実行可能にするため、Fault Analyzer では、CNFDSN 設定が IDIOPTLM 構成オプション・モジュールに用意されています。この設定によって、代替データ・セットを IDICNFxx 構成メンバーに指定できます (詳しくは、[parmlib メンバー IDICNFxx ページ 372](#)を参照してください)。

指定されたデータ・セット名は、MVS™ システム・シンボルを含み、LRECL=80 および RECFM=FB で PDS(E) として割り振られる必要があります。データ・セットには汎用 READ アクセス権が付与されている必要があります。

Fault Analyzer では、IBM ADFz Common Components IPVCNF00 parmlib メンバーが存在する必要はありませんが、IDIOPTLM 構成オプション・モジュールで NOIPVOPT 値が 1 に設定されていない限り、IPVCNF00 の処理を試行します。詳しくは、[IBM Application Delivery Foundation for z/OS Common Components オプションを無視 \(NOIPVOPT\) ページ 343](#)を参照してください。

IPVCNF00 parmlib メンバーが存在するかどうかを判別するには、Fault Analyzer が以下のいずれかに対する READ アクセス権を持っている必要があります。

- 論理 parmlib 連結
- IBM ADFz Common Components IPVOPTLM 構成オプション・モジュールで指定されたデータ・セット

そうでない場合、異常終了 S913 および ICH408I のようなメッセージ (セキュリティ・サーバーに応じて異なる) が発生します。

IPVOPTLM 構成オプション・モジュールおよび IPVCNF00 parmlib メンバーの詳細については、「*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*」を参照してください。

## デフォルト・リカバリー障害記録 IEATDUMP データ・セット名 (RFRDSN) の変更

SDUMP が使用できない場合、Fault Analyzer リカバリー障害記録機能では、異常終了するリアルタイム分析タスクで IEATDUMP データ・セットを割り振りできることが必要です。(リカバリー障害記録機能の一般情報については、[リカバリー障害記録 ページ 57](#)を参照してください。)

デフォルトの IEATDUMP データ・セット名パターンは次のとおりです。

```
IDIRFRHQ.IDIRFR.&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.
```

デフォルトのデータ・セット名パターンを変更するには、IDIOPTLM 構成オプション・モジュール内の RFRDSN 設定を使用します。

データ・セット名パターンには、割り振られた各データ・セットが固有であることを保証するシンボル値が含まれる必要があります。例:

```
CL100'IDIHLQ.TDUMP.&&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.'
```

記号の使用についての詳細は、[IDIOPTLM データ・セット名のシンボル置換 ページ 343](#)を参照してください。



**注:** すべてのユーザーがパターンを指定してデータ・セットを割り振れるようにすることが重要です。詳しくは、「[リカバリー障害記録データ・セット・アクセスの管理 ページ 310](#)」を参照してください。

リカバリー障害記録 IEATDUMP データ・セット名パターンに UTC の代わりにローカル日時が含まれていることが望ましい場合、&YYMMDD を &LYMMDD に、また &HHMMSS を &LHHMMSS に変更します。

RFR TDUMP キャプチャーを無効にするには、特殊な高位修飾子 NULLFILE を使用します。この名前を使用した場合は、メッセージ [IDI0136W ページ 726](#) が発行されます。

## CICS XDUREQ グローバル・ユーザー出口 (CICSNOA) を使用して起動された場合の Fault Analyzer のアクションの変更

IDIOPTLM 構成オプション・モジュールの CICSNOA 設定は、CICS® XDUREQ グローバル・ユーザー出口を介して起動された場合の Fault Analyzer のアクションを変更するために使用できます。これは、以下に説明するように、インストール済み環境において CICS® トランザクションの異常終了を特殊処理する必要がある場合にのみ検討すべきことです。

前の CICS® 出口プログラムが戻りコード UERCBYP (トランザクション・ダンプ抑止) を指定していた場合、Fault Analyzer はデフォルトで分析を続行します。この動作は、MVS™ 事前ダンプ出口 IDIXDCAP による非 CICS 異常終了の処理方法に等しいものです。

CICSNOA 設定を使用すると、前の XDUREQ 出口プログラムの戻りコード UERCBYP により Fault Analyzer は分析を停止します。

## 代替セキュリティー・サーバー・テスト・データ・セット名の指定 (SSCHKDSN)

IDIOPTLM 構成オプション・モジュールの SSCHKDSN 設定は、Fault Analyzer がセキュリティー・サーバーの通常応答を検査するときの代替データ・セット名を提供するために使用されます。これは、実行環境の一般検証の一部として、リアルタイム再分析の前に実行されます。SSCHKDSN オプションで指定されている代替データ・セット名がない場合、Fault Analyzer はデフォルトのデータ・セット名 SILLY99.DATA88.SET77.NAME66.CHECK55 を使用します。指定されたデータ・セット名は、存在する必要はありませんが、構文的には有効でなければなりません。

## デフォルトの拡張ミニダンプ (XDUMP) データ・セット名パターン (XDUMPDSN) の変更

XDUMP は、ミニダンプ内のページに加えてストレージのページを含む、障害項目と密に結合されたデータ・セットです。

デフォルトの XDUMP データ・セット名パターンは次のとおりです。

```
IDIXDPHQ.XDUMP.&&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.
```

拡張ミニダンプ・データ・セットの命名パターンを変更するには、IDIOPTLM 構成オプション・モジュール内の XDUMPDSN 設定を使用します。

データ・セット名パターンには、割り振られた各データ・セットが固有であることを保証するシンボル値が含まれる必要があります。例:

```
CL100'IDIHLQ.XDUMP.&&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.'
```

記号の使用についての詳細は、[IDIOPTLM データ・セット名のシンボル置換 ページ 343](#)を参照してください。

データ・セットを作成する必要があるユーザーへのアクセスを許可するには、1つ以上のデータ・セット修飾子を IDIXDUMP\_HLQ XFACILIT プロファイルと一緒に使用する必要があります。詳細については、[XDUMP データ・セットに対する XFACILIT リソース・クラスの使用 ページ 314](#)を参照してください。

XDUMPDSN オプションに指定した高位修飾子が NULLFILE の場合、Fault Analyzer は拡張ダンプ・データ・セットを割り振りません。この結果として、イベント関連ソース行またはステートメントに直接関係していないユーザー・ストレージ領域が、障害項目の再分析から欠落する可能性があり、MaxMinidumpPages オプション値が超過した場合に、DeferredReport オプションがオーバーライドされる可能性があります (特に CICS で、重大なパフォーマンスの問題を引き起こす可能性があります)。または対話式再分析のために TSO 領域に追加スペースが必要になる可能性があります。

## コピーされた SDUMP データ・セット名パターンの指定 (SDUMPDSN)

コピーされた SDUMP データ・セット名パターンを指定するには、IDIOPTLM 構成オプション・モジュール内の SDUMPDSN 設定を使用します。

データ・セット名パターンには、割り振られた各データ・セットが固有であることを保証するシンボル値が含まれる必要があります。例:

```
CL100'IDIHLQ.SDUMP.&&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ.'
```

記号の使用についての詳細は、[IDIOPTLM データ・セット名のシンボル置換 ページ 343](#)を参照してください。

データ・セットを作成する必要があるユーザーへのアクセスを許可するには、1つ以上のデータ・セット修飾子を IDISDUMP\_HLQ XFACILIT プロファイルと一緒に使用する必要があります。詳細については、[コピーされた SDUMP データ・セットに対する XFACILIT リソース・クラスの使用 ページ 316](#)を参照してください。

SDUMPDSN オプションにデータ・セット名を指定していない場合や、高位修飾子を NULLFILE にしている場合、Fault Analyzer はコピーした SDUMP データ・セットを作成しません。

## IBM Application Delivery Foundation for z/OS Common Components オプションを無視 (NOIPVOPT)

インストールにより汎用 READ アクセス権がユーザーに提供されていない場合、および次のいずれかの場合、Fault Analyzer が IPVCNF00 parmlib メンバーを処理しようとするセキュリティ・サーバーのアクセス違反が発生します。

- IBM Application Delivery Foundation for z/OS® Common Components がインストールされていません。
- IBM Application Delivery Foundation for z/OS® Common Components Parmlib メンバー IPVCNF00 が設定されていません。

FAULT Analyzer が IDIOPTLM 構成オプション・モジュールの検出または IPVCNF00 parmlib メンバーの処理を試行しないようにするには、IDIOPTLM 構成オプション・モジュールの NOIPVOPT オプションを 1 に設定します。

IPVOPTLM 構成オプション・モジュールおよび IPVCNF00 parmlib メンバーの詳細については、「*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*」を参照してください。

## 前の出口が RC=8 の場合に IDIXDCAP リアルタイム分析を抑止 (PREVEXIT)

前の事前ダンプ出口でダンプを抑止するために RC=8 が設定されている場合、IDIXDCAP 事前ダンプ呼び出し出口による Fault Analyzer リアルタイム分析を抑止するには、PREVEXIT オプション・フィールドを 1 に設定します。

PREVEXIT オプション・フィールドは、非推奨の IDISOPT2 SMP/E USERMOD を置き換えます。

## IDIOPTLM データ・セット名のシンボル置換

シンボル置換に関するこの情報は、RFRDSN、SDUMPDSN および XDUMPDSN オプションで指定したデータ・セット名に適用されます。

これらのデータ・セット名には、割り振られた各データ・セットが固有であることを保証する MVS システム・シンボルが含まれる必要があります。特に、&SEQ. (Fault Analyzer 順序番号変数) は名前に含まれている必要があります。&SEQ. 変数は 5 桁の 10 進数で置換されます。

障害項目の移動またはコピー ([ヒストリー・ファイル項目のコピー ページ 143](#)および[ヒストリー・ファイル項目の移動 ページ 145](#)を参照)、または IDIUTIL IMPORT ([IMPORT 制御ステートメント ページ 448](#)を参照) 操作の結果として割り振られたデータ・セットの場合、シンボル &JOBNAME. および &SYSNAME. は障害項目の作成時にそれぞれの値に従って解決されます。例えば、障害項目がシステム SYS1 でリアルタイム処理中に作成され、後でシステム SYS2 でコピーされた場合、&SYSNAME. シンボルは引き続き値 SYS1 で置換されます。

&SEQ.、&JOBNAME. および &SYSNAME. 以外の指定したすべてのシンボルは、ASASYMBM サービスを使用して現在のシステム値に置換されます。使用可能なシンボル名については、「z/OS MVS アセンブラー・サービス解説書」を参照してください。

最終データ・セット名(すべての記号変数が可能な限り長い値で置換される)が 44 文字を超えないようにします。44 文字を超える場合、Fault Analyzer は最終データ・セット名を 44 文字(最後の文字がピリオドの場合は 43 文字)に切り詰めます。この場合、データ・セット名が一意ではなくなり、割り振りエラーが発生する可能性があります。



## 第 18 章. ヒストリー・ファイルのセットアップ

Fault Analyzer では、1 つ以上の PDS または PDSE データ・セットを障害ヒストリー・ファイルとして割り振る必要があります。

ヒストリー・ファイルの DUMMY 指定がサポートされていますが、実際のデータ・セットを使用して、ISPF インターフェイスに付属の機能を利用できます。

### ! 重要:

- 1 つの例外を除き、ヒストリー・ファイルは空でない限り名前を変更しないでください。変更すると、関連するダンプ・データ・セットは使用できなくなり、障害項目が削除されるときに自動的に削除されなくなります。

このルールの例外は、次で概説されている手順に従う場合です: [PDSE ヒストリー・ファイルのサイズの変更 ページ 349](#)。

- 先にすべての障害項目を削除せずに、ヒストリー・ファイルが削除されると、関連する密結合ダンプ・データ・セットは孤立します。ヒストリー・ファイルの名前の変更または削除の前に、IDIUTIL DELETE(ALL) を使用してすべての障害項目を削除します。詳しくは、[DELETE 制御ステートメント ページ 445](#)を参照してください。

## 割り振られるヒストリー・ファイルのサイズの決定

割り振られるヒストリー・ファイルのサイズは、インストール環境や、障害が記録されるジョブのタイプなど、いくつかの要因により決まります。ただし、スペース所要量の初期近似値 (KB) として、次の式を使用できます。

```
kilobytes = 500 * number-of-entries
```

各部の意味は以下のとおりです。

### **number-of-entries**

ヒストリー・ファイルに保持する同時項目の最大数です。

障害の典型的な数を記録した後、実際のヒストリー・ファイル・スペースの使用率を、メンバーの合計数で割ると、平均の障害項目サイズをさらに正確に判別できます。

初期ヒストリー・ファイルの推奨サイズは 100 シリンダーです。潜在的なスペース不足状態を防止するには、2 次スペースを使用してヒストリー・ファイルを割り振ります。

## ヒストリー・ファイルへの PDS または PDSE の割り振り

1 つ以上のヒストリー・ファイルを割り振るには、サンプル・ジョブ IDISHIST を編集し、実行依頼します。


このジョブにより、2 つの PDSE データ・セット IDI.HIST および IDI.HIST.TEST がそれぞれ割り振られます。異常終了ジョブをこれらのヒストリー・ファイルに記録するすべてのユーザーは、データ・セットへの書き込みアクセス権を保有

している必要があります。このジョブに対して行わなければならない可能性のある変更について、詳細は、サンプル・ジョブ内の指示を参照してください。

 **注記:**

1. PDSE データ・セットはスペース管理の自動化、IDIS サブシステム管理 ([「履歴・ファイル \\$INDEX データのキャッシュ ページ 320」](#) 参照)、共用使用でのディレクトリー保全性を高める効果が極めて高いことにより、履歴・ファイルには PDSE データ・セットの使用をお勧めします。(PDS 管理対象履歴・ファイルも使用できます。)
2. Fault Analyzer は、0 より大きいメンバー世代が定義された PDSE バージョン 2 の履歴・ファイル・データ・セットをサポートしません。

バッチ・ジョブを実行依頼して新しい履歴・ファイルを割り振る代わりに、Fault Entry List 画面のアクション・バーの File プルダウン・メニューを選択して履歴・ファイルを割り振ることができます。詳しくは、[新しい履歴・ファイルの割り振り ページ 93](#)を参照してください。

 **重要:** 書き込まれると予想されるすべてのデータを収容できるスペースを、履歴・ファイルに割り振ります。そうでない場合、スペース不足状態が発生し、その結果、Fault Analyzer が異常終了する (例えば、システム異常終了 SE37) 可能性があります。

履歴・ファイルのスペース管理を支援するために、Fault Analyzer は、次の用途に使用できる IDIUTIL バッチ・ユーティリティーを備えています。

- 古い履歴・ファイル項目を削除する (DELETE 制御ステートメント)。
- AUTO 管理 (または自己保守) 履歴・ファイルをセットアップする (SetMinFaultEntries 制御ステートメント)。

これらの機能は、Fault Analyzer ISPF インターフェースを使用して実行することもできます。

シンプレックスでは、任意の数のシステム間で履歴・ファイルを共用できます。

#### 関連情報

[履歴・ファイルの管理 \(IDIUTIL ユーティリティー\) ページ 441](#)

[AUTO で管理される PDSE 履歴・ファイル ページ 347](#)

[障害履歴・ファイル設定の変更 ページ 94](#)

[シンプレックス全体での履歴・ファイルの共用 ページ 364](#)

## 履歴・ファイルの 2 次スペースの割り振り

特に PDSE 履歴・ファイルの場合は、2 次スペースを使用してデータ・セットを割り振ることが重要です。これに失敗すると、後でデータが失われた場合にスペース不足状態が発生する可能性があります。

通常、Fault Analyzer は [論理ヒストリー・ファイル・サイズ ページ 96](#) (ヒストリー・ファイルが最初に障害項目の最小数に達したときに確立される割り振りサイズ) 内で PDSE ヒストリー・ファイルを自動管理します。( [AUTO で管理される PDSE ヒストリー・ファイル ページ 347](#) を参照。) ただし、主に PDSE の同時更新機能のために、新しい障害項目が書き込まれるときにフリー・スペースの使用可能性を保証できない場合があります。したがって、Fault Analyzer では、予想されるスペースが使用可能でない場合に、2 次スペースを使用してヒストリー・ファイルを拡張できるようにすることが重要です。

Fault Analyzer によって使用される論理ヒストリー・ファイル・サイズ概念により、次の更新が発生すると、ヒストリー・ファイルの使用サイズは自動的に元のサイズに戻されます。また、割り振られた 2 次エクステン트는、余分なスペースを必要とする同様の状況が再度発生するまで使用されません。注意すべき重要なことは、ヒストリー・ファイルが無期限に増大し続けないことです。

IDIS サブシステムが 2 次スペースのない PDSE ヒストリー・ファイルを検出すると、メッセージ [IDI0191W ページ 738](#) を発行します。

## PDS と PDSE の間でのヒストリー・ファイルの変換

ヒストリー・ファイルを PDS データ・セットから PDSE データ・セットに、または PDSE データ・セットから PDS データ・セットに変換する場合は、Minimum Fault Entries または Maximum Fault Entries 値を設定してください。この設定を行わないと、スペース不足などの予期しない結果を引き起こす可能性があります。

変換されたヒストリー・ファイルが PDSE データ・セットの場合は、以下のいずれかを実行します。

- SETMINFAULTENTRIES 制御ステートメントを使用して IDIUTIL を実行する。
- ISPF インターフェースの **Change Fault History File Settings** ダイアログを使用して **Minimum Fault Entries** 値を設定する。

変換されたヒストリー・ファイルが PDS データ・セットの場合は、以下のいずれかを実行します。

- SETMAXFAULTENTRIES 制御ステートメントを使用して IDIUTIL を実行する。
- ISPF インターフェースの **Change Fault History File Settings** ダイアログを使用して **Maximum Fault Entries** 値を設定する。

---

### 関連資料

[SETMINFAULTENTRIES 制御ステートメント ページ 447](#)

[SETMAXFAULTENTRIES 制御ステートメント ページ 447](#)

### 関連情報

[障害ヒストリー・ファイル設定の変更 ページ 94](#)

## AUTO で管理される PDSE ヒストリー・ファイル

デフォルトでは、新規作成の PDSE ヒストリー・ファイルは AUTO で管理されます。

PDSE ヒストリー・ファイルは、IDIUTIL SetMinFaultEntries 制御ステートメントを使用して、または Fault Analyzer ISPF インターフェースを使用して管理することもできます。

ヒストリー・ファイルは、少なくとも 25 個の障害項目が作成されるまではアクティブに AUTO で管理されません。(最小障害項目数はデフォルトで 25 ですが、IDIUTIL バッチ・ユーティリティーの SetMinFaultEntries 制御ステートメントを使用したり Fault Analyzer ISPF インターフェースを使用したりすることで、この値を変更できます。)25 個の障害項目が作成されるまでは、2 次エクステントを割り振ることでヒストリー・ファイル・データ・セットのサイズを増やすことができます。

25 個の障害項目を超過すると直ちに、Fault Analyzer が、割り振られているヒストリー・ファイルの現在のサイズを記録します。このサイズは論理ヒストリー・ファイル・サイズです。割り振られているすべてのスペースが使い尽くされるまで追加の障害項目は依然としてヒストリー・ファイルに書き込むことができます。

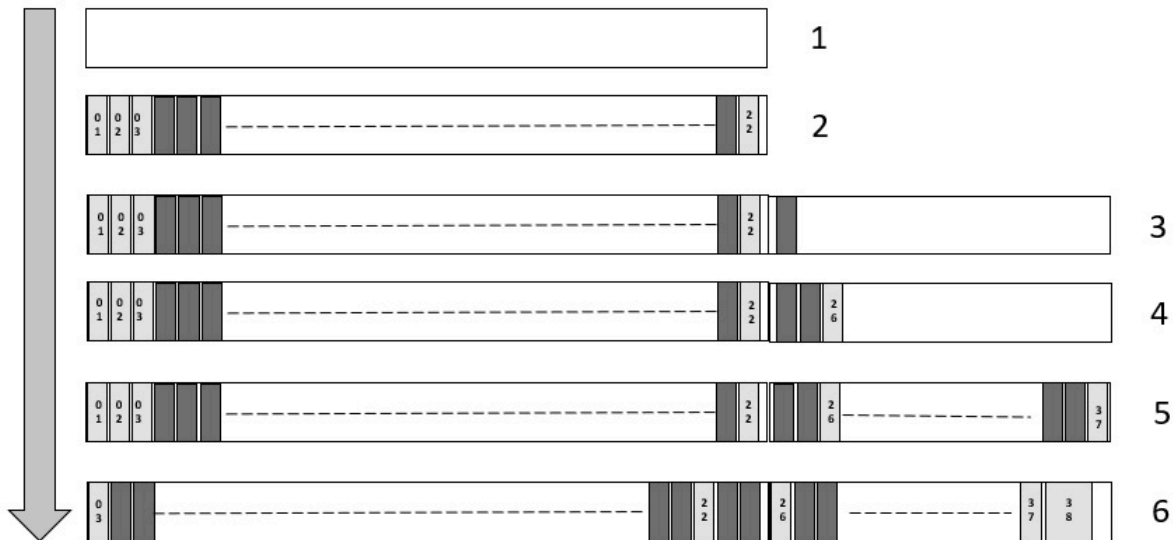
ヒストリー・ファイルにおいて使用されていないスペースが不足してくると、Fault Analyzer は、25 個を超えた適格な障害項目を古い順に削除し、新しい障害項目を書き込むために必要となるスペースを提供します。通常は、この処理により、データ・セット・エクステントを追加で割り振らなくてもよくなり、関連スペース不足状態とともにヒストリー・ファイルが無制限に肥大化し続けることはなくなります。

障害 ID の割り当てと再利用については、[障害ヒストリー・ファイル ページ 25](#)を参照してください。

## 例

図 188. AUTO で管理される PDSE ヒストリー・ファイルの例

MY.HISTORY SPACE=(20,10)



- 1 次割り振り: 1 次スペース 20 個のシリンダーと 2 次スペース 10 個のシリンダーを使用して新規ヒストリー・ファイルが割り振られます。
- 障害項目はヒストリー・ファイルに書き込まれます。最初の 20 個のシリンダーが使用されるまでには、ヒストリー・ファイルに 22 個の障害項目が含まれるとします。

3. 次の障害項目は、2次割り振りをトリガーします。23番目の項目が書き込まれるときに、最小値である25個の障害項目にはまだ達していないため、10個のシリンダーの追加エクステントが割り振られます。
4. 26番目の障害項目が書き込まれると、割り振られているヒストリー・ファイルの現在のサイズが論理ヒストリー・ファイルのサイズとして記録されます。この例では、以下ようになります。20個のシリンダー + 10個のシリンダー = 30個のシリンダー
5. 30個のシリンダーがすべて使用されるまでは、障害項目は書き込まれ続けます。30個のシリンダーがすべて使用されたときにヒストリー・ファイルに含まれる障害項目の数は37だとします。
6. 38番目の障害項目が書き込まれる前に、割り振られている30個のシリンダー内にその新規障害項目が収まるまで、障害項目が(古いもの順に)削除されます。

サイズに関係なく、残される障害項目の数が25個以下になれば、古い障害項目を削除する処理は停止します。

---

#### 関連資料

[SETMINFAULTENTRIES 制御ステートメント ページ 447](#)

#### 関連情報

[障害ヒストリー・ファイル設定の変更 ページ 94](#)

## PDSE ヒストリー・ファイルのサイズの変更

PDSE ヒストリー・ファイルの割り振りサイズは変更できます。

1. IDIS サブシステムを停止します。
2. 元のヒストリー・ファイルの名前を固有の名前に変更して、他のユーザーがヒストリー・ファイルとして使用できないようにする必要があります。例えば、「.OLD」をヒストリー・ファイル名に付け足します。
3. 元のヒストリー・ファイルとは異なる固有の名前（「.NEW」を付け足すなど）を使用して、2次エクステントを持つ新しい大きなヒストリー・ファイルを割り振ります。
4. IEBCOPY または ISPF オプション 3.3 を使用し、「.OLD」ヒストリー・ファイルの全コンテンツを「.NEW」ヒストリー・ファイルにコピーします。
5. アクション・バーから **File > Change Fault History File Settings** を使用して、「論理ヒストリー・ファイル・サイズ (ページ数)」の値を 0 に設定します。この設定について詳しくは、[障害ヒストリー・ファイル設定の変更 ページ 94](#) を参照してください。
6. 「.NEW」ヒストリー・ファイルの名前を元のヒストリー・ファイル名に変更します。
7. IDIS サブシステムを開始します。
8. ISPF オプション 3.2 または類似するものを使用し、「.OLD」ヒストリー・ファイルを削除します。

## 以下へのヒストリー・ファイル名の指定: Fault Analyzer

現行の障害ヒストリー・ファイルは、異常終了ジョブ・ステップで IDIHIST DD ステートメントを明示的にコーディングするか、または DataSets オプションの IDIHIST サブオプションを指定することにより判別できます。これらのいずれも指定されていない場合は、デフォルト名は IDI.HIST となります。

## ビューのセットアップ

このセクションでは、インストール済み環境のビューをセットアップする方法について説明します。ビューの使用に関する一般情報については、[ビューの使用 ページ 66](#)を参照してください。

ビューは基本的に、Fault Analyzer ISPF インターフェースを使用してまとめて表示される履歴・ファイルのリストです。

XFACILIT クラスを使用せずに個々の履歴・ファイル障害項目へのアクセスを提供する場合 ([履歴・ファイル障害項目のアクセスの管理 ページ 365](#)を参照)、インストール・システムで作成したいビューの種類を決めるために、特別な考慮が必要になることがあります。それらの考慮事項については、[XFACILIT リソース・クラスを使用しない場合のビューに関する考慮事項 ページ 352](#)に概要が示されています。

ビューを作成するための履歴・ファイルのデータ・セット名のリストは、ビュー定義を保持するために作成された別の PDS のメンバーに配置されます。このメンバーは通常、すべてのユーザーが読み取りアクセス権を持つ、インストール済み環境全体のデータ・セットです。これは、DataSets オプションの IDVIEWS データ・セットがポイントするものです。メンバーの名前がビュー名として使用されます。ISPF オプション・プルダウンを使用して履歴・ファイルを変更し、F4 を押してビューをリストすると、ビュー名のリストが表示されます。オプションで、これらのメンバーの最初の行の 1 列目にアスタリスクを指定してコメントにすることができます。このコメント行は、ビュー・メンバーに存在する場合、選択画面でビュー名の横に表示されます。このコメント行に示されるビューの説明は、ユーザーが選択するときに役立ちます。

**図 189: ビュー・データ・セット・メンバーの例** [ページ 350](#) では、ビュー・データ・セット・メンバーの例を示しています。

**図 189. ビュー・データ・セット・メンバーの例**

```
* Department P024 history files ①
P024.&SYSNAME..CICS.HIST ②
P024.IMS.HIST ③
* Include the installation-wide dehistory file ④
IDI.HIST ⑤
```

上記の例では、以下を示しています。

### ①

この行は、コメントです (1 列目のアスタリスク)。このコメントは、ビュー・メンバーの 1 行目にあるため、Fault Analyzer ISPF インターフェースを使用してビューのリストを表示する際のビュー記述としても使用できます。

### ②、③、および⑤

これらは、このビューを選択するときに、Fault Analyzer ISPF インターフェースを使用して同時に表示される完全修飾履歴・ファイル名です。各データ・セット名は、列 1 から始まり、1 行で指定する必要があります。データ・セット名を一重引用符で囲むのはオプションです。1 つのビューで指定できるデータ・セット名の合計数には、制限がありません。

ビュー・メンバー内で指定されるデータ・セット名に適用されるシンボル置換規則は、DataSets オプションの IDVIEWS

サブオプションで指定されるデータ・セット名に適用される規則と同じです。詳しくは、[DataSets オプションのデータ・セット名の置換シンボル ページ 580] を参照してください。

④

この行は、Fault Analyzer ISPF インターフェースでは無視されるコメントです。

PDS または PDSE である必要がある場合を除いて、ビュー・データ・セットの特定の属性は必要ありません。論理レコード長には、最長のデータ・セット名、および最長の -HistCols (デフォルトの列レイアウトの指定 ページ 351 を参照) または -Match (最初の障害項目選択基準の指定 ページ 351 を参照) の指定を含めることができるサイズが必要です。1つの例として、論理レコード長が 32,756 バイトのレコード・フォーマット VB を使用してください。ビュー・メンバーに、シーケンス番号を含めないでください。

複数のヒストリー・ファイル間で障害 ID をより簡単に参照するために、個々のヒストリー・ファイル用に最大 3 文字の障害接頭部文字 (英字) を設定する機能が使用可能です。複数のグループ間でさまざまな接頭部文字を使用すると、ユーザーが複合ビュー表示で障害を表示するときに、所有グループを認識しやすくなります。障害接頭部文字は、IDIUTIL パッチ・ユーティリティーの SETFAULTPREFIX コマンドを使用して設定または変更できます。

## デフォルトの列レイアウトの指定

ビューのレコードの 1 列目に -HistCols キーワードを使用すると、デフォルトの列レイアウトを指定できます。

**i ヒント:** ビュー記述も必要な場合、このオプションを先頭のレコードに指定することは避けてください。

-HistCols キーワードに続けて、Fault Entry List 画面の列のリストを括弧で囲んで記述する必要があります。列名リストの構文およびフォーマットは、FAISPFopts(HistCols(...)) オプションと同じです (FAISPFopts ページ 598 を参照)。Fault Entry List Column Configuration 画面から、列を望ましい形に構成した実際の Fault Entry List 画面 (Fault Entry List の列構成 ページ 71 を参照) をカット・アンド・ペーストすれば、簡単に列名リストを取得できます。

完成した -HistCols 指定は、単一のビュー・メンバー・レコードに適合する必要があります。複数レコードの連結はサポートされていません。

-HistCols の指定においては、キーワードそのものも含め大/小文字が区別されません。キーワードの後の開き括弧文字の前には、空白文字がいくつあっても、またはなくても構いません。

単一のシングル・ビュー・メンバー内で複数の -HistCols 指定が検索された場合、最後の指定のみが有効になります。

デフォルトの列レイアウトを指定しているビュー・データ・セット・メンバーの例を示します。

```
* All IMS history files
-HistCols(IMS_Pgm Jobname User_ID System Abend Date Time )
SYS1.IMS.HIST
SYS2.IMS.HIST
```

## 最初の障害項目選択基準の指定

ビューのレコードの 1 列目に -Match キーワードを使用すると、最初の障害項目選択基準を指定できます。


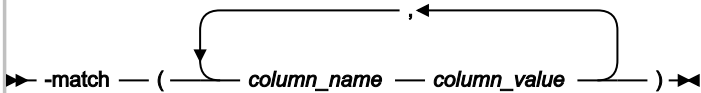
 ヒント: ビュー記述も必要な場合、このオプションを先頭のレコードに指定することは避けてください。

図 190. Syntax



`column_value` には、ワイルドカードを使用できます。サポートされているワイルドカード文字には、アスタリスク (\*) (0 個以上の文字を表す) と % 記号 (単一の必須文字を表す) があります。指定されたストリングに一致するテーブル・データ行のみが表示されます。

完成した `-Match` 指定は、単一のビュー・メンバー・レコードに適合する必要があります。複数レコードの連結はサポートされていません。

`-Match` の指定においては、キーワードそのものも含め大/小文字が区別されません。キーワードの後の開き括弧文字の前には空白文字がいくつあっても、または、なくても、構いません。

単一のシングル・ビュー・メンバー内で複数の `-Match` 指定が検索された場合、最後の指定のみが有効になります。

デフォルトの最初の障害項目リスト基準を指定しているビュー・データ・セット・メンバーの例を示します。

```
* My CICS view
-Match(CICS_Trn *)
PROD.CICS.HIST
```

## XFACILIT リソース・クラスを使用しない場合のビューに関する考慮事項

履歴・ファイル障害項目へのアクセスの制御に XFACILIT リソース・クラスを使用していないインストール済み環境の場合、最も効率的なビューのセットアップを行うには、Fault Analyzer のさまざまなユーザー・セットをすべて理解しておく必要があります。各ユーザー・グループについて、主に以下の事柄について把握しておく必要があります。

1. データ・セットに対するデータ・セット・セキュリティー・プロファイルの書き込みアクセス権。
2. 他のユーザー・グループが実行依頼したジョブの障害の検討または処理を行う必要性。
3. グループ間で障害項目を読み取り専用またはアクセス権なしにする必要性。

インストール・システムのセキュリティー管理者は、主にデータ・セット・セキュリティー要件を取り扱うため、上記の要件をある程度まで了解済みです。追加の要素は、グループ間の障害の可視性に対する要件です。

履歴・ファイル障害項目へのアクセスの制御に XFACILIT リソース・クラスを使用しない場合、履歴・ファイルのセキュリティーは、ご使用のシステムのデータ・セット・セキュリティーです。ユーザーのジョブが障害履歴・ファイルに障害を記録するには、そのジョブが使用する履歴・ファイルへの書き込み (更新) アクセス権が必要です。同じグループ内のすべてのユーザーは、そのグループ用に設定された、履歴・ファイルの障害への読み取りアクセス権および削除アクセス権を保有していることになります。

ユーザー・グループごとに個別の障害履歴・ファイルを保持し、ジョブで使用する履歴・ファイルへの書き込み (更新) アクセス権限のみを提供するのが、一般的には良い方法です。ただし、この方法だと、障害がさまざまなヒス



トリー・ファイルにある場合に、ユーザーは環境全体にわたって障害を調べることが困難になります。この問題を解決するには、ユーザーが1つの Fault Analyzer ISPF 画面で多数のヒストリー・ファイルの複合ビューを参照できるようにする「ビュー」を作成します。

## 共用 DASD のない MVS システム全体でのヒストリー・ファイルの管理

ビューを使用すると、複数のヒストリー・ファイル内の障害項目に簡単にアクセスできます。ただし、この簡易アクセスが可能なのは、ビュー内のすべてのヒストリー・ファイルが、同じ MVS™ システムからアクセス可能な場合のみです。

これらのシステムで DASD が共用される場合、これらのシステムでは単に共通ヒストリー・ファイルを使用できます。障害が発生したあるシステムから分析用の別のシステムに障害項目を送信するために、2つのソリューションが提供されます。

- 最初のソリューションは、通知ユーザー出口を使用し、手操作による介入は不要です。[自動化インプリメンテーション ページ 353](#)を参照してください。
- 2番目のソリューションは、対話式再分析中にフォーマット・ユーザー出口を使用して「オンデマンド」で障害項目を送信します。[オンデマンドの実装 ページ 361](#)を参照してください。

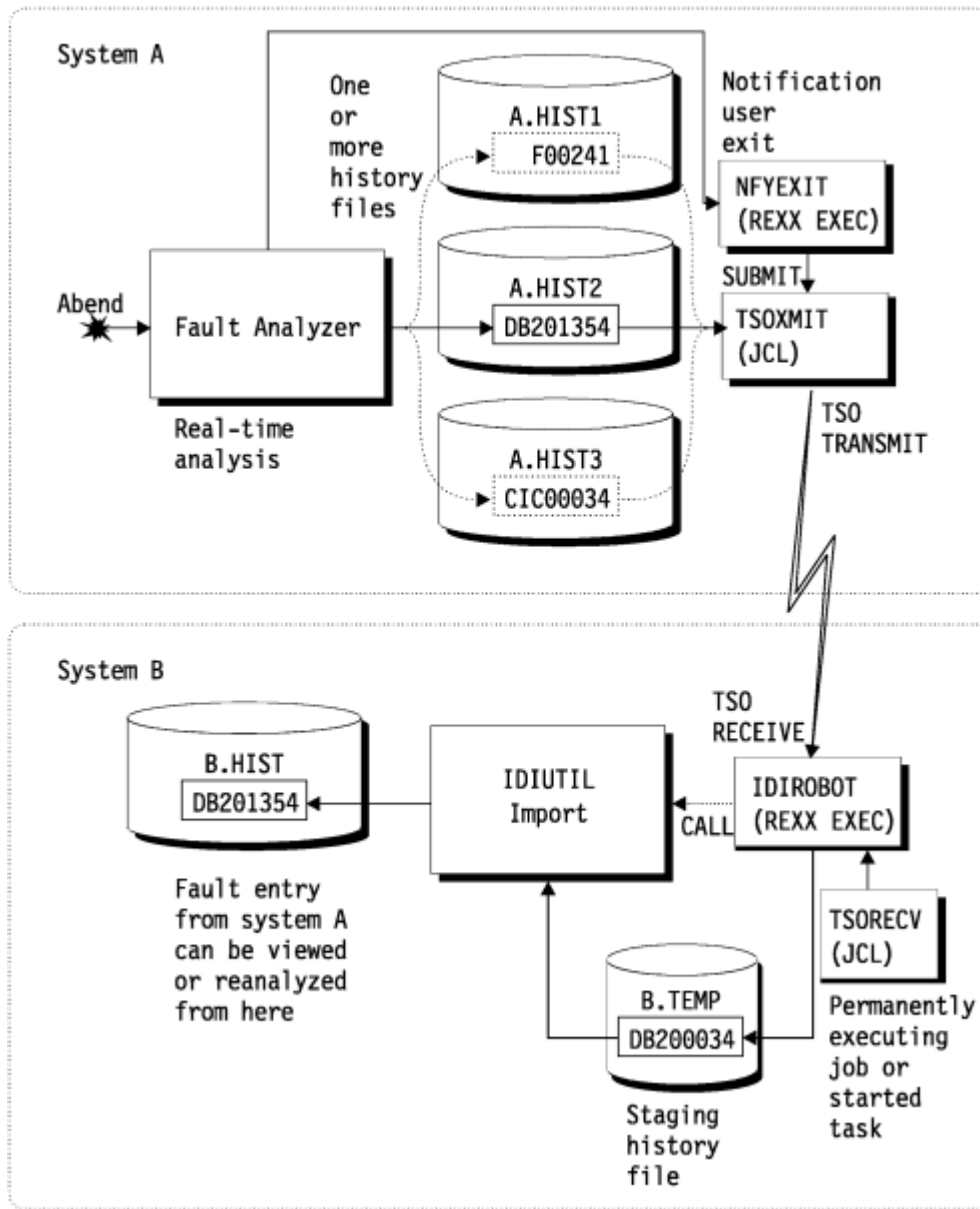
## 自動化インプリメンテーション

A および B という2つの MVS™ システムがあるとします。これらのシステムで DASD は共用されません。システム A で作成された障害項目は、Fault Analyzer ISPF インターフェースを使用して表示/再分析のためにシステム B に自動的にコピーできます。

その項目はシステム A 上の通知ユーザー出口を使用してコピーされます。システム A では TSO バッチ・ジョブが実行され、障害項目がシステム B 上の専用ユーザー ID に PDS メンバーとして送信されます。システム B では、連続して実行されるバッチ TSO ジョブが障害項目を受信してステージング・データ・セットに入れます。次に IDIUTIL バッチ・ユーティリティーが呼び出されて、障害項目がローカル・ヒストリー・ファイルにインポートされます。

 図 191 : TSO XMIT/RECEIVE による障害項目インポート ページ 354 はこの概念を示します。

図 191. TSO XMIT/RECEIVE による障害項目インポート



ユーザー出口全般については、[ユーザー出口を使用した Fault Analyzer のカスタマイズ ページ 464](#)を参照してください。具体的な通知ユーザー出口については、[通知ユーザー出口 ページ 499](#)を参照してください。IDIUTIL バッチ・ユーティリティーについては、[ヒストリー・ファイルの管理 \(IDIUTIL ユーティリティー\) ページ 441](#)を参照してください。

### NFYEXIT: サンプル通知ユーザー出口

NFYEXIT は、データ・セット IDI.SIDISAM1 内のメンバー IDISXNFY として、ソフトコピー・フォーマットで使用可能です。

## 図 192. バッチ TSO XMIT ジョブを実行依頼するサンプル通知ユーザー出口 (NFYEXIT)

```

nodeid   = 'MVS'           /* <--- verify/change */      ①
userid   = 'IDIRBOT'      /* <--- verify/change */      ②
jobcard  = '//NOTIFY JOB MSGCLASS=Z' /* <--- verify/change */      ③
/*****/  ④
/* #Optionally, add checks here for selective transmission of fault */
/* entries that only match a certain criteris. */
/* For example: */
/* If ENV.USER_ID = "FRED" then exit 0 */
/* If ENV.USER_IDIHIST = "MY.HISTFILE" the exit 0 */
/*****/
"MAKEBUF"
queue jobcard
queue '//*****'
queue '//* Export fault entry'
queue '//*****'
queue '//EXPORT EXEC PGM=IDIUTIL"
queue "//DD1 DD DISP=(,PASS),"
queue "// SPACE=(CYL,(10,100,5),RLSE),"
queue "// DCB=(DSORG=PO,RECFM=VB,LRECL=10000)"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSIN DD *"
queue " EXPORT("ENV.IDIHIST"("ENV.FAULT_ID"),DD1)"
queue "/*"
queue '//*****'
queue '//* Terse the export data set'
queue '//*****'
queue "//TERSE EXEC PGM=AMATERSE,PARM='PACK'"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSUT1 DD DISP=SHR,DSN=*.EXPORT.DD1"
queue "//SYSUT2 DD DISP=(,PASS),"
queue "// SPACE=(CYL,(10,100),RLSE)"
queue "//SYSPRINT DD SYSOUT=*"
queue '//*****'
queue '//* Perform TSO XMIT of the exported and tersed fault entry'
queue '//*****'
queue "//XMIT EXEC PGM=IKJEFT01"
queue "//DD1 DD DISP=SHR,DSN=*.TERSE.SYSUT2"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSTSIN DD *"
q_rec(" XMIT" nodeid"."userid "DDNAME(DD1) -")
q_rec(" NONOTIFY")
queue '/*'
/* 'Submit' the stacked TSO batch job */
n = queued()
"IDIALLOC DD(DD1) SYSOUT PGM(INTRDR)"
if rc = 0 then do /* allocation worked so generate output */
  address mvs "EXECIO" n "DISKW DD1 (FINIS"
  "IDIFREE DD(DD1)"
  say 'Fault entry' ENV.FAULT_ID 'sent to' nodeid'.'userid
end
else do
  "IDIWTO Allocation of INTRDR failed"
  say 'Fault entry' ENV.FAULT_ID 'job submission failure'
end
exit 0

/* Pad record with blanks to 80 bytes. */
q_rec: procedure
parse arg rec
if (length(rec) < 80) then rec = rec||copies(' ',80-length(rec))
queue rec
return 0

```

**注:****①**

「nodeid」は障害項目の送信先のターゲット・システムを指定します。

**②**

「userid」は障害項目がターゲット・システムで受信されるユーザー ID を指定します。このユーザー ID は、障害項目の受信にのみ使用してください。

**③**

ジョブ・カードがローカル標準に準拠していることを確認します。

**④**

障害が別のシステムへの送信対象として適格であるかどうかを確認するために、ここでチェックを追加できます。例では、ユーザー ID やヒストリー・ファイル名をどのように使用できるのかを示していますが、ENV データ域や NFY データ域にあるすべてのフィールドをチェックすることが可能です。

システム A で発生するすべての障害に対して、[図 192: バッチ TSO XMIT ジョブを実行依頼するサンプル通知ユーザー出口 \(NFYEXIT\) ページ 355](#) の出口を呼び出すことができます。この出口を活用するには、次のオプションを IDICNFxx 構成メンバーに追加します (*exec.lib* は REXX EXEC PDS または PDSE データ・セットです)。

```
DataSets (IDIEXEC (exec.lib))
Exits (NOTIFY (REXX (NFYEXIT)))
```

## IDIROBOT: 障害項目を受信するためのサンプル REXX EXEC

IDIROBOT は、データ・セット IDI.SIDISAM1 内のメンバー IDISROBT として、ソフトコピー・フォーマットで使用可能です。この *exec* は以下のアクションを行います。

1. IDIROBOT ユーザー用のファイルを受信し、ステージング・データ・セットに含めます。このデータ・セットから、IDIUTIL バッチ・ユーティリティーを使用してローカル・ヒストリー・ファイルにインポートされます。
2. IDIUTIL IMPORT ユーザー出口、IDIROBEX を作成します ([図 236: REXX IDIUTIL ユーザー出口の例 ページ 510](#)を参照)。

## 図 193. TSO 受信 REXX exec (IDIROBOT) の例、パート 1

```
histfile = 'B.HIST'           /* <--- verify/change */ ⑤
temphist = 'B.TEMP'          /* <--- verify/change */ ⑥
seconds = '60'               /* <--- verify/change */ ⑦
use_exit = 'Y'               /* <--- Y|N. verify/change */ ⑧
address tso
x = prompt('on')
x = outtrap('var.',10,'noconcat')
do forever
  /* Obtain information about transmitted data on the JES output queue */
  if queued() = 0 then queue 'end'
  'receive'
  input = 'N'

  /* Examine the output from the 'dummy' receive command.
     The following variables are initialized:
     dsn      - the 'sending' history file name
     fromid   - the user ID performing the TSO XMIT
     node     - the JES node from which the fault entry was sent
     faultid  - the fault ID (member name) */
do i = 1 to var.0
  parse var var.i msgno t1 t2 t3 t4 t5 t6
  if msgno = 'INMR901I' then do
    dsn = t2
    fromid = t4
    node = t6
  end
  else if msgno = 'INMR902I' then do
    faultid = t2
    input = 'Y'
    leave
  end
end
end
```

## 図 194. TSO 受信 REXX exec (IDIROBOT) の例、パート 2

```

/* Perform actual receive to the staging history file followed by an
   IDIUTIL batch utility import if there is data available */
if input = 'Y' then do
  if faultid <> "" then do
    /* Receiving a PDS/E. */

    say 'Receiving' dsn('faultid') from' node'.'fromid
    queue "DSN('temphist')"
    queue 'END'
    'RECEIVE'
  end
else do
  /* Receiving a sequential data set - assume AMATERSE PACKed. */
  say 'Receiving' dsn 'from' node'.'fromid
  queue "DSN('temprecv')"
  queue 'END'
  'RECEIVE'

  /* Perform AMATERSE UNPACK. */
  "ALLOC DD(SYSPRINT) DUMMY"
  "ALLOC DD(SYSUT1) DA('temprecv') SHR"
  "ALLOC DD(SYSUT2) DSN('temphist'),
    NEW CATALOG UNIT(SYSALLDA) RECFM(V B) LRECL(10000),
    CYLINDERS SPACE(10,100) DIR(5)"
  address tso "CALL *(AMATERSE) 'UNPACK'"
  say 'UNPACK rc =' RC

  /* Get fault ID (member name). */
  "LISTDS 'temphist' MEMBERS"
  /* Sample output: */
  /* FRED.$$TEMP$$HIST */
  /* --RECFM-LRECL-BLKSIZE-DSORG */
  /* VB 10000 27998 PO */
  /* --VOLUMES-- */
  /* E$US21 */
  /* --MEMBERS-- */
  /* F01103 */
  mbr_start = 0
  do i = 1 to var.0
    /*say "var."i"="var.i"*/
    if mbr_start = 0 then do
      if strip(var.i) = "--MEMBERS--" then do
        mbr_start = i + 1
        leave
      end
    end
  end
  if mbr_start = var.0 then do
    /* One, and only one, member. */
    faultid = strip(var.mbr_start)
  end
else do
  say 'ERROR: More than one member found in data set' temphist,
    '- terminating'
  exit 12
end
'FREE DD(SYSUT2)'
'FREE DD(SYSUT1)'
'FREE DD(SYSPRINT)'

"DELETE 'temprecv'"
end

```

## 図 195. TSO 受信 REXX exec (IDIROBOT) の例、パート 3

```

/* The target history file in the 'histfile' variable could be */
/* determined here based on any of the initialized variables */
/* dsn, fromid, node or faultid. This sample EXEC uses a single */
/* history file only. */ ⑤

/* Perform IDIUTIL IMPORT. */
'ALLOC DD(SYSIN) NEW REU UNIT(VIO) RECFM(F B) LRECL(80)'
'ALLOC DD(SYSPRINT) SYSOUT'
if use_exit = 'Y' then
  parms.1 = "EXITS(IMPORT(REXX(IDIROBEX)))"
else
  parms.1 = "* Using IDILOPTM for dump data set names"
  parms.2 = "IMPORT('histfile',"
  parms.3 = " 'temphist'('faultid'),PACKAGE)"
  parms.0 = 3
  "EXECIO * DISKW SYSIN (STEM parms. FINIS"
  address tso "CALL *(IDIUTIL)"
  say 'IMPORT rc =' RC
end
else do
  /* Sleep for 60 seconds before attempting to receive again */
  address tso "call *(idisleep) 'seconds'"
end
end

```



⑤

この項目は、受信された障害項目が組み込まれるターゲット・ヒストリー・ファイルの名前です。障害が最初に発生した場所に基づくヒストリー・ファイルを選択するには、⑥を参照してください。

⑥

この項目は、TSO

受信コマンドに使用されるステージング・データ・セットです。このデータ・セットから障害項目がターゲット・ヒストリー・ファイルにインポートされます。



**重要:** 事前割り振りデータ・セットは使用しないでください。例に示したように、execにより、受信した各障害に対するこのステージング・データ・セットを割り振りおよび削除します。

IDIROBOT exec および IDIUTIL IMPORT

処理が動作する場合、ステージング・データ・セットは通常のヒストリー・ファイルとして使用されることや、複数のメンバーを含むことはできません。通常のヒストリー・ファイルとして使用される場合 (例えば、「障害項目リスト」画面を使用して表示される場合や、IDIUTIL FILES または LISTHF 制御ステートメントのターゲットとして使用される場合)、\$\$INDEX メンバーが作成される可能性があり、それにより処理が動作しなくなります。また、データ・



セットが管理対象の IDIS サブシステムになる可能性があり、それにより後でシリアルサイズの問題が発生する可能性があります。

ステージング・データ・セットが受信および IMPORT 処理の間でのみ存在するようにすることで、これらの問題が発生する可能性が排除されます。

### 7

IDIROBOT exec は、WAIT

状態に入って、受信される障害項目のチェック間でリソースを保持します。障害項目の受信相互間の時間間隔 (秒数) をここに指定できます。選択されたユーザー ID の JES 出力キュー上の障害項目がすべて受信され、IDIROBOT exec が WAIT に移行します。

### 8

- RFRDSN、XDUMPDSN および SDUMPDSN オプションが IDIOPTLM 構成オプション・ロード・モジュールで有効なデータ・セット名のパターンに設定された場合、IDIROBEX ユーザー出口を使用する必要はありません。(Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338 を参照)。この場合、「use\_exit」は「N」に設定します。
- IDIROBEX ユーザー出口を使用する場合、出口により提供されるダンプ・データ・セット名は IDIOPTLM の同等のオプション設定を上書きします。

### 9

このサンプル exec では、受信されるすべての障害項目に対して 1 つのターゲット・ヒストリー・ファイルのみが使用されます。以下のいずれかの項目に基づくターゲット・ヒストリー・ファイルを割り当てることができます。

- 元のヒストリー・ファイル名 (変数「dsn」)。
- 送信ユーザー ID (変数「fromid」)。
- 送信元のノード ID (変数「node」)。
- 障害 ID 自体 (変数「faultid」)。

図 196: IDIROBOT exec (IDISTSQB) を実行する TSO バッチ・ジョブの例 ページ 361 に IDIROBOT exec を実行するバッチ TSO ジョブの例を示します。これは、データ・セット IDI.SIDISAM1 内のメンバー IDISTSQB として、ソフトコピー・フォーマットで使用可能です。



## 図 196. IDIROBOT exec (IDISTSOB) を実行する TSO バッチ・ジョブの例

```
//IDISTSOB JOB <job card parameters>
// SET EXECDSN=exec.lib                <--- verify/change
//TSOBATCH EXEC PGM=IKJEFT01
//SYSEXEC DD DISP=SHR,DSN=&EXECDSN.
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSTSIN DD *
      IDIROBOT
/*
//IDIEXEC DD DISP=SHR,DSN=*.TSOBATCH.SYSEXEC
//IDITRACE DD SYSOUT=*
```

**!** **重要:** IDIROBOT exec が実行されているユーザー ID (この例では、IDISTSOB ジョブの実行依頼者) に、ステージング・データ・セット、およびインポート用のターゲットとして使用されるすべてのヒストリー・ファイルの両方への更新アクセス権があることを確認します。

IDIROBOT exec は終了しないため、IDISTSOB ジョブは無期限に実行されます。ただし、不必要なリソースを使用しないように、この exec により、ジョブが着信データの受信試行間で WAIT 状態に入ります。ジョブを終了するには、非アクティブ期間に MVS™ CANCEL コマンドを使用してください。または、選択されたユーザー ID に送信されると出口を終了させる特定のファイルを、exec に認識させることもできます。

JES イニシエーターを起動しないようにするために、この JCL を実行するための開始済みタスクを代わりに定義できます。

## オンデマンドの実装

フォーマット・ユーザー出口を使用して、一定の基準に一致する障害項目を分析用の別のシステムに送信できます。このプロセスは、対話式再分析レポートから必要に応じて開始されます。

次の図では、フォーマット REXX ユーザー出口、IDIXMIT の例を示しています。これは、データ・セット IDIXMIT 内のメンバー IDI.SIDISAM1 として、ソフトコピー・フォーマットで使用可能です。

図 197. バッチ TSO XMIT ジョブを実行するフォーマット・ユーザー出口 (IDIXMIT) の例 (パート 1/2)

```

nodeid   = 'MVSΒ'                               /* <--- verify/change */ ❶
userid   = 'IDIROBOT'                             /* <--- verify/change */ ❷
jobcard  = '//NOTIFY JOB MSGCLASS=Z'             /* <--- verify/change */ ❸
"MAKEBUF"
queue jobcard
queue '/******'
queue '/* Export fault entry'
queue '/******'
queue "//EXPORT EXEC PGM=IDIUTIL"
queue "//DD1 DD DISP=(,PASS),"
queue "// SPACE=(CYL,(10,100,5),RLSE),"
queue "// DCB=(DSORG=PO,RECFM=VB,LRECL=10000)"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSIN DD *"
queue " EXPORT("ENV.IDIHIST"("ENV.FAULT_ID"),DD1)"
queue "/*"
queue '/******'
queue '/* Terse the export data set'
queue '/******'
queue "//TERSE EXEC PGM=AMATERSE,PARM='PACK'"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSUT1 DD DISP=SHR,DSN=*.EXPORT.DD1"
queue "//SYSUT2 DD DISP=(,PASS),"
queue "// SPACE=(CYL,(10,100),RLSE)"
queue "//SYSPRINT DD SYSOUT=*"
queue '/******'
queue '/* Perform TSO XMIT of the tersed export data set'
queue '/******'
queue "//XMIT EXEC PGM=IKJEFT01"
queue "//DD1 DD DISP=SHR,DSN=*.TERSE.SYSUT2"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSTSIN DD *"
q_rec(" XMIT" nodeid."userid "DDNAME(DD1) -")
q_rec(" NONOTIFY")
queue '/*'

```

図 198. バッチ TSO XMIT ジョブを実行するフォーマット・ユーザー出口 (IDIXMIT) の例 (パート 2/2)

```

/* 'Submit' the stacked TSO batch job.                */
n = queued()
"IDIALLOC DD(DD1) SYSOUT PGM(INTRDR)"
if rc = 0 then do /* if alloc worked */
  address mvs "EXECIO" n "DISKW DD1 (FINIS"
  "IDIFREE DD(DD1)"
end
else ,
  "IDIWTO ALLOCATION OF INTRDR FAILED"
  "DROPBUF"
/*****/ ④
/* #Optionally, update the user title field to show that the fault */
/* has been sent. */
/* For example: */
/* ENV.USER_TITLE = "Sent to" nodeid "on" DATE()". */
/*****/
/*****/ ⑤
/* #Optionally, tell the user that the request has been processed */
/* by uncommenting the following IDIWRITE calls: */
/*"IDIWRITE '<p>History file" ENV.IDIHIST "fault ID" ENV.FAULT_ID, */
/*      "sent to" nodeid"."userid"." */
/*"IDIWRITE '<p>Press PF3 to return to the interactive reanalysis', */
/*      "report.'" */
/*****/
exit 0

/* Pad record with blanks to 80 bytes.                */
q_rec: procedure
parse arg rec
if (length(rec) < 80) then rec = rec||copies(' ',80-length(rec))
queue rec
return 0

```



注:

①

「nodeid」は障害項目の送信先のターゲット・システムを指定します。

②

「userid」は障害項目がターゲット・システムで受信されるユーザー ID を指定します。

③

ジョブ・カードがローカル標準に準拠していることを確認します。

④

このコードのコメントを外すと、この送信が行われた日付と送信先システムをユーザー・タイトル・フィールドに指定することによって、障害項目を送信済みとして識別するためのオプションの方法が提供されます。



5

発行されたコマンドが何を実行したかという情報の画面をユーザーに表示する場合、このコードのコメントを外します。

手動で送信される障害項目の受信側は、[自動化インプリメンテーション ページ 353](#)で説明されている exec にすることもできるし、または受信とその後のインポートを手動で行う任意のユーザー ID またはノードでもかまいません。

手動でのインポートは PACKAGE オプションを指定した IDIUTIL IMPORT を使用して実行する必要があります。[IMPORT 制御ステートメント ページ 448](#) を参照してください。

## オンデマンドの実装の使用

障害項目を分析しているときに、以下の基本コマンドを入力します。

```
EXEC IDIXMIT
```

IDIXMIT の名前は、サンプル IDIXMIT exec のコピーを作成するときに付けた任意の他の名前に変更できます。

## シスプレックス全体での履歴・ファイルの共用

シスプレックス内ですべての MVS™ イメージにまたがって PDSE 履歴・ファイルを効率良く共用するために、Fault Analyzer では IDIS サブシステムからの XCF メッセージングが使用されるようになりました。

この方式を SYS1.PARMLIB の IGDSMSxx メンバー内の PDSESHARING(EXTENDED) オプションと組み合わせることにより、シスプレックス内のすべての MVS™ イメージから、競合なしに個々の障害項目を並行して読み書きすることができます。

シスプレックス全体で PDS 履歴・ファイルを共用すると、複数の MVS™ イメージから同時に同じ履歴・ファイルに対するアクティビティが活発になった場合、アクセス方式がどの更新についてもデータ・セット全体のエンキューを必要とするため、好ましくない競合が発生する可能性があります。この問題は PDSE データ・セットには適用されません。PDSE データ・セットは、メンバー・レベルでのシリアライゼーションのみを必要とするので、こちらが賢明な選択といえます。

以前にセットアップされた環境の中には、シスプレックス内のすべての MVS™ イメージ間でマスター・カタログ (またはユーザー・カタログ) を共用しないという通常とは異なる条件でセットアップされたものがあり、そのような環境では、それぞれの MVS™ イメージは、異なる DASD ボリュームに対してカタログされている同じデータ・セット名を使用することが許されます。このセットアップは Fault Analyzer 履歴・ファイルでは使用しないでください。共通データ・セット名は、IDIS サブシステムおよび各 MVS™ イメージ間の ENQ メカニズムを通じて、不要な競合を起こしてしまいます。各 MVS™ イメージ間で異なる履歴・ファイルが必要であれば、最良の代替案として、履歴・ファイル・データ・セット名の中で &SYSCLONE. 置換変数を使用することを検討してください。固有名を付けることで共通名の ENQ 競合がなくなり、より論理的で実際的な管理が可能になるからです。

共用履歴・ファイルに代わる方法は、Fault Analyzer ISPF インターフェースの「View」機能を使用することです。そこでは、同時に複数の履歴・ファイルを表示できます。この機能の詳細については、[ビューのセットアップ ページ 350](#)を参照してください。

履歴・ファイルを複数の MVS™ イメージで共用する場合、以下の点に注意してください:


- PDSE を共有するシステムは、同じシスプレックスのメンバーでなければならない。また、SYS1.PARMLIB の IGDSMSxx メンバーで PDSESHARING(EXTENDED) が指定された状態ですべてのシステムが稼働していることが推奨される。(IPL を実行する最初のシスプレックス・メンバーが、使用される共有モードを判別し、そのシスプレックスに加わる後続のすべてのメンバーは、EXTENDED か NORMAL にかかわらず、そのモードでの動作が強制されることに注意してください。)
- XCF はアクティブでなければならない。
- GRS (または機能的に同等なサブシステム) が稼働中である必要があります。PDSE 逐次化は、リソースの大分類名 SYSZIGW0 および SYSZIGW1 を使用して管理されます。GRS 以外の逐次化製品が使用される場合を除いて、GRS RNL リストに変更を加える必要はありません。

上記の考慮事項にかかわらず、シスプレックス全体で共有される PDSE データ・セットに対して、OPEN エラーが発生する可能性があります。この状態は、システム間共有の競合の結果発生し、通常、メッセージ IEC143I RC 213-70 が発行されません。

シスプレックスにおける PDSE データ・セットの共有の詳細については、Redbooks® 資料である *“Partitioned Data Set Extended Usage Guide”* ([www.redbooks.ibm.com](http://www.redbooks.ibm.com) からダウンロード可能) の *“PDSE Sharing and serialization”* の章、および第 28 章 *“Processing a PDSE”* の *“Sharing PDSEs”* セクション (*“DFSMS: Using Data Sets”* マニュアル) を参照してください。

システム間共有の競合を減らすためにヒストリー・ファイル・アクセス管理が用意されているため、インストール・システムで PDSESHARING(NORMAL) を使用する必要がある場合は、IDIS サブシステムも必要となります。ただし、Fault Analyzer の外部からヒストリー・ファイルにアクセスする場合は、メッセージ IEC143I RC 213-70 が引き続き表示される可能性があります。

シスプレックスで共有される障害ヒストリー・ファイルを使用する際、特別な Fault Analyzer オプションは必要ありません。

 **ヒント:** IDIS サブシステムの PARM='UPDINDEX' オプションを設定するかどうかは、シスプレックス内のすべての IDIS サブシステムで統一します。このオプションの詳細については、[ヒストリー・ファイル \\$\\$INDEX データのキャッシュ ページ 320](#) を参照してください。

## ヒストリー・ファイル障害項目のアクセスの管理

ヒストリー・ファイルへのアクセスの管理は、セキュリティー・サーバー・データ・セット・プロファイルを使用することでしかできません。例えば、所定のヒストリー・ファイルは、すべてのユーザーに同じアクセス権限が付与されている単一部門内でのみ使用される場合です。この場合、すべてのユーザーには通常、ヒストリー・ファイルに対して少なくとも UPDATE アクセス権限が付与されます。したがって、新規障害項目の作成と、作成者に関係なく、既存の障害項目の表示、変更、または削除を行うことができます。

さらに詳細に個別の障害項目へのアクセスを制御する必要がある場合、Fault Analyzer は XFACILIT リソース・クラスの使用をサポートします。これについて、以下で説明します。



**注:** XFACILIT アクセス権がある程度の処理オーバーヘッドと関連している場合、CICS® または IMS™ などのパフォーマンスが重要なアプリケーションには、履歴・ファイルと通常のデータ・セット・プロファイル・アクセス権を使用することができます。

## 履歴・ファイル障害項目に対しての XFACILIT リソース・クラスの使用

履歴・ファイル障害項目への必要なアクセスが通常のデータ・セット・プロファイルを介してまだ使用可能でない場合は、XFACILIT リソース・クラスを使用して、Fault Analyzer を介してアクセスできるようにできます。

履歴・ファイルに XFACILIT プロセスを使用するには、通常のデータ・セット・プロファイルを介してユーザーに使用可能なアクセスを制限して、Fault Analyzer で検査された XFACILIT プロファイルを介したアクセスを使用可能にします。通常のデータ・セット・プロファイルで可能な範囲を超えて、XFACILIT リソース・クラスを使用して Fault Analyzer アクセスを制限することはできません。このため、このセットアップに対しては通常の履歴・ファイル・アクセスを UACC(NONE) に設定する必要があります。

この設計は、XFACILIT によって、制限ではなく許可を与えるためのものです。例えば、ユーザーが ISPF 3.4 を使用して履歴・ファイルを参照できない場合でも、XFACILIT プロファイルにより Fault Analyzer を介して履歴・ファイルの障害項目を使用する許可を与えることが可能です。

Fault Analyzer で使用される XFACILIT クラス・プロファイル名は、次の 2 つです。

```
IDIHIST_GROUP_DSN.group.hist-dsn
IDIHIST_USERID_DSN.userid.hist-dsn
```

各部の意味は以下のとおりです。

### **group および userid**

これらは、現行ユーザーに関連するセキュリティー・サーバーのデフォルトの GROUP および USER ID (新規障害項目を作成する場合)、または障害項目に関連するセキュリティー・サーバーの GROUP および USER ID です (既存の障害項目を表示、更新、または削除する場合)。

障害項目に関連したセキュリティー・サーバーのグループおよびユーザー ID は、障害項目を当初に作成したユーザーの現行のグループおよびユーザー ID です。

### **hist-dsn**

障害項目が常駐する履歴・ファイルの名前。

XFACILIT クラス・プロファイル名の例を次に示します。

```
IDIHIST_GROUP_DSN.PAYROLL.IDI.HIST
IDIHIST_USERID_DSN.USER01.SYSA.PROD.HIST
```

通常のデータ・セット・プロファイルによるアクセスが使用可能でない場合、Fault Analyzer は、上記の XFACILIT クラス・プロファイル名の両方、つまりグループとユーザー ID を使用して、所定の履歴・ファイル障害項目へのアクセス権限を検査します。いずれかで必要なアクセス権限が提供されることが判別されると、履歴・ファイルのアクションが実行されます。

Fault Analyzer によって使用される XFACILIT クラス・プロファイルのアクセス・レベルは、次のように Fault Analyzer アクションに変換されます。

表 9. Fault Analyzer アクションに必要な XFACILIT プロファイルのアクセス・レベル

アクション	例	XFACILIT アクセス・レベル
読み取り	保管レポートの表示、または再分析の実行。	READ
書き込みまたは作成	既存の障害項目内のユーザー・ノートの更新、またはヒストリー・ファイルへの新規障害項目の作成。	UPDATE または CONTROL
削除	D または DD 行コマンドを使用した明示的な削除のみ。	ALTER

## XFACILIT インプリメンテーション例 1

IDI.COMMON.HIST という名前のヒストリー・ファイルがあり、例えば以下を使用してデータ・セットへの一般アクセスを防ぐとします。

```
ADDSD 'IDI.COMMON.HIST' UACC(NONE)
```

次の XFACILIT プロファイルおよびアクセスを定義します (*group* のインスタンスごとに繰り返します):

```
RDEFINE XFACILIT IDIHIST_GROUP_DSN.group.IDI.COMMON.HIST XFACILIT UACC(NONE)
PERMIT IDIHIST_GROUP_DSN.group.IDI.COMMON.HIST XFACILIT CLASS(XFACILIT) ID(group) ACCESS(UPDATE)
```

必要に応じて、追加のグループまたはユーザーに、XFACILIT スtring へのアクセス権限を付与できます。

これにより、IDI.COMMON.HIST ヒストリー・ファイルへのアクセスは、以下のようになります。

- 障害項目には、Fault Analyzer からのみアクセスできます。
- 障害項目は、以下のユーザーが表示または再分析できます。
  - 障害項目を作成したユーザー
  - その他のユーザーのうち、同じグループのメンバーであるユーザー、または障害項目作成者のデフォルト・グループ ID が含まれた XFACILIT プロファイルへの明示的なアクセス権限が付与されたユーザー

## XFACILIT インプリメンテーション例 2 グローバル・アクセス・テーブルの使用

ヒストリー・ファイル TEST.ZZ.HISTORY.DEFAULT を使用し、グループ ID およびユーザー ID により障害項目を保護するには、まずデータ・セットへの一般アクセスを防止します。

```
ADDSD 'TEST.ZZ.HISTORY.**' UACC(NONE)
```

次に、Fault Analyzer がグループ ID およびユーザー ID に基づき PDS または PDSE データ・セット内の個々の障害項目メンバーへのアクセスを認可できるように、グローバル・アクセス・テーブルを使用して XFACILIT クラスをセットアップします。

```
SETROPTS GLOBAL(XFACILIT)
RDEFINE GLOBAL XFACILIT (<-- not required if already defined)
RALTER GLOBAL XFACILIT ADDMEM(IDIHIST_GROUP_DSN.&RACGPID.TEST.ZZ.HISTORY.**/ALTER)
```

```
RALTER GLOBAL XFACILIT ADDMEM(IDIHIST_USERID_DSN.&RACUID.TEST.ZZ.HISTORY.**/ALTER)
SETROPTS GLOBAL(XFACILIT) REFRESH
```

グローバル・アクセス・テーブルでは &RACUID および &RACGPID を許可するので、管理作業が減ります。



1. RACF® の場合、&RACUID および &RACGPID は、グローバル・アクセス・テーブルにリストされているプロファイルにのみ有効です。
2. Fault Analyzer のアクセス許可の判断方法が決まっている場合、グローバル・アクセス・テーブルのいずれの XFACILIT プロファイルも、一致する実際の XFACILIT プロファイルによって裏付けられていることが RACF® の要件です。例えば、次の実際の XFACILIT プロファイルを追加して、グローバル・アクセス・テーブルのプロファイルをここで使用可能にします。

```
PROFILE NOPREF
RDEFINE XFACILIT IDIHIST_GROUP_DSN.*.TEST.ZZ.HISTORY.** UACC(NONE)
RDEFINE XFACILIT IDIHIST_USERID_DSN.*.TEST.ZZ.HISTORY.** UACC(NONE)
SETR REFR RACLIST(XFACILIT)
```

上記の XFACILIT 定義は、例えば以下のような TEST.ZZ.HISTORY 修飾子で始まるすべてのヒストリー・ファイルのデータ・セット名に該当します。

```
TEST.ZZ.HISTORY.DEFAULT
TEST.ZZ.HISTORY.PAYROLL
TEST.ZZ.HISTORY.CICS.SYS01
```

## XFACILIT インプリメンテーション例 3ACF2 の使用

OEM セキュリティー・サーバーは、以下の ACF2 コマンドのように、特定の製品に変換されたコマンドを必要とします。このコマンドの要件は、RACF® の場合と同様に、SAF RACROUTE 要求がその OEM 製品に対して同等の戻り情報を提供することです。以下に、ACF2 の例を示します。

```
$KEY(TEST)
ZZ.HISTORY.- UID(<string for MVS support>) READ(A) WRITE(A) ALLOC(A) EXEC(A)
ZZ.HISTORY.- UID(*)
```

この場合、TEST.ZZ.HISTORY.\* ヒストリー・ファイルには、<string for MVS™ support> リスト内のユーザーのみが直接アクセスできます。

```
$KEY(IDIHIST_USERID_DSN) TYPE(XFC) or $KEY(IDIHIST_GROUP_DSN) TYPE(XFC)
-.TEST.ZZ.HISTORY.- UID((<string for MVS support>) ALLOW
-.TEST.ZZ.HISTORY.- UID(*) SERVICE(READ,UPDATE) ALLOW
```

TYPE(XFC) は、XFACILIT クラスのデフォルトです。このセットアップでは、一般ユーザーは READ 権限および UPDATE 権限のみ許可され、MVS™ サポートのみが障害項目を明示的に削除できます。ただし、各ヒストリー・ファイルの SetMaxFaultEntries/SetMinFaultEntries 設定によっては、Fault Analyzer が自動的に削除を行います。



## IDIXFXIT ユーザー出口の使用

IDIHIST\_GROUP\_DSN.group.hist-dsn または IDIHIST\_USERID\_DSN.userid.hist-dsn XFACILIT プロファイルを使用してアクセス権限が許可されなかった場合は、オプションの IDIXFXIT ユーザー出口が呼び出されます。IDIXFXIT 出口は、使用可能であれば、最後は、障害項目のアクセスを認可する可能性があります。この方法は、インストール済み環境が以前は別のアクセス許可スキームを採用しており、XFACILIT プロファイルの代わりに、あるいは XFACILIT プロファイルと併せてこの出口の使用を続けたい場合に役立ちます。

IDIXFXIT ユーザー出口は、APF 許可ライブラリー内になければならず、IDIXFXIT という名前で LINKLIST 内でロード・モジュールとして使用可能になっている必要があります。LE に準拠している場合、C や PL/I の main 関数を含める必要はありません。IDIXFXIT ユーザー出口は、NORENT オプションを使用してリンク・エディットされている必要があります。

## エントリーの指定

IDIXFXIT へのエントリーについて、レジスターの内容は次のとおりです。

### レジスター

#### 内容

1

入力パラメーター・リストのアドレス (下を参照)

13

72 バイトのレジスター保管域のアドレス

14

戻りアドレス

15

IDIXFXIT のエントリー・ポイント・アドレスです。

## 入力パラメーター・リスト

以下のパラメーター・リストのアドレスは、R1 内の IDIXFXIT ユーザー出口に渡されます。すべてのパラメーターは、C スタイルのヌル終了文字ストリングとして提供されます。値には 'X'00' を含む 1 バイトが後ろに続きます。

表 10. IDIXFXIT 入力パラメーター

パラメーター	バイト数	説明
パラメーター 1	可変	障害項目に対する必要なアクセス権限のレベル: 読み取り、更新、または削除。
パラメーター 2	可変	IDIXFXIT ユーザー出口呼び出し元のセキュリティー・サーバーのユーザー ID 1。

表 10. IDIXFXIT 入力パラメーター

(続く)

パラメーター	バイト数	説明
パラメーター 3	可変	IDIXFXIT ユーザー出口呼び出し元のセキュリティー・サーバーのデフォルト・グループ ID <sup>1</sup> 。
パラメーター 4	可変	障害項目を作成したジョブの名前。
パラメーター 5	可変	障害項目が含まれているヒストリー・ファイル・データ・セット名。
パラメーター 6	可変	障害項目の HD セグメント・データ域。  HD セグメント・データ域のマッピングは、データ・セット IDI.SIDISAM1 メンバーが以下で提供されます。  <ul style="list-style-type: none"> <li>• IDISXPLA (アセンブラー)</li> <li>• IDISXPLC (C)</li> <li>• IDISXPLB (COBOL)</li> <li>• IDISXPLP (PL/I)</li> </ul>



注:

1. パラメーター 6 が指している HD セグメント・データ域から、障害項目作成者のセキュリティー・サーバーのユーザー ID とデフォルト・グループ ID の情報を取得します。

## 戻り指定

IDIXFXIT から戻るときに、レジスターの内容は、次のものでなければなりません。

### レジスター

#### 内容

0 ~ 1

未定義。

2 ~ 14

未変更。

15

戻りコード:

0

アクセス権限無認可。

1

Read アクセス権限認可済み。

2

Update アクセス権限認可済み。

3

Delete アクセス権限認可済み。

## 例 (C)

以下は、C で書かれた IDIXFIT 出口の例です。

```
#include "IDISXPLC"
int IDIXFIT(char *action, char *userid, char *group, char *jobnm,
            char *histDSN, DDIR_DATA_HD *pHD) {
    if (strcmp("MY.HIST", histDSN) == 0) return 2; /* Update access ok */
    else return 0 ;                               /* No access */
}
```

## 第 19 章. サイトのデフォルト・オプションの設定および変更

製品のデフォルトをオーバーライドし、IDICNFxx のすべてのユーザーが自分のサイトで使用可能な Fault Analyzer オプションを含む parmlib メンバー Fault Analyzer の作成方法について、以下に説明します。

### parmlib メンバー IDICNFxx

サイトのデフォルト・オプションは parmlib メンバー IDICNFxx に保持されます。ここで、xx は以下のいずれかです。

- 00
- 現在の MVS™ シンボル &SYSCLONE に一致する値。

&SYSCLONE は、1 文字または 2 文字の固有のシステム ID を表す標準 MVS™ システム・シンボルです。MVS™ の &SYSCLONE シンボルについては、「[MVS™ 初期設定およびチューニング解説書](#)」を参照してください。

IDICNF&SYSCLONE メンバー名を使用することで、インストール済み環境のシスプレックスで各 MVS™ イメージに異なるオプションを指定するとともに、引き続き共通 PARMLIB データ・セットを使用することができます。

初期検索の対象は IDICNF&SYSCLONE メンバー名になります。このメンバー名がいずれの parmlib データ・セットにも見つかからない場合は、IDICNF00 メンバーが検索されます。

メンバー IDICNFxx は、SYS1.PARMLIB 内、または論理 parmlib 連結の一部である、その他のデータ・セット内に作成できません。

論理 parmlib 連結のデータ・セットにはすべて、汎用 READ アクセス権が与えられます。



**注:** IDICNFxx メンバーを論理 parmlib 連結に配置せずに、代わりに別のデータ・セットを使用する場合には、[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#)を参照してください。

IDICNFxx メンバーが存在しない場合、Fault Analyzer は製品提供のデフォルト・オプションを使用します。リアルタイムの場合、メッセージ [IDI0018W ページ 699](#) が発行されます。

これは、IDICNFxx parmlib メンバーの例です。

## 図 199. IDICNFxx parmlib メンバーの例

```

/*-----*/
/* IBM Fault Analyzer Configuration */
/*-----*/

Exclude(TYPE(TSU)) /* Exclude TSO users */
Exclude(TYPE(STC) NAME(VTAM)) /* Exclude VTAM started task */

RetainDump(AUTO) /* Automatic dump retention */

/* Data sets where installation application compiler listings are kept */
DataSets(
  IDIHIST (IDI.HIST) /* Fault History file */
  IDILC (MY.LISTING.C /* C compiler listings */
        XY.LISTING2.C) /* more C compiler listings */
  IDILCOB (APP1.LISTING.COBOL /* COBOL compiler listings */
          MY.LISTING.COBOL) /* COBOL compiler listing for IVP */
  IDILCOB0(APP2.LISTINGS) /* OS/VS COBOL compiler listings */
  IDILPLI (MY.LISTING.PLI) /* PL/I compiler listings */
  IDILANGX(MY.IDILANGX) /* LANGX files */
  IDIADATA(MY.SYSADATA) /* SYSADATA files */
  IDIMAPS(IDI.SIDIMAPS) /* Data area models */
)

```

IDICNFxx parmlib メンバーの例 (上記のものとは異なる場合があります) は IDI.SIDISAM1 データ・セット内のメンバー IDICNF00 に該当します。

IDICNFxx メンバーの 1 から 71 カラムまでのみが処理されることにご注意ください。

Fault Analyzer をインストールする際は、IDICNF00 サンプル・メンバー内の DataSets オプションを確認してください。

- Fault Analyzer 以外の上位修飾子を使用して IDI をインストールした場合、以下の DDname のデータ・セットを指定して DataSets オプションを組み込む必要があります。

IDIMAPS

IDIDOC

- IDI.IDIVSENU 以外の名前を使用して VSAM メッセージおよび異常終了コード説明リポジトリを割り振った場合、IDIVSENU DDname のデータ・セット名を指定して DataSets オプションを組み込む必要があります。
- IDI.HIST 以外の名前を使用してデフォルトのヒストリー・ファイルを割り振った場合、IDIHIST DDname のデータ・セット名を指定して DataSets オプションを組み込む必要があります。

DataSets オプションについて詳しくは、[DataSets ページ 576](#)を参照してください。

parmlib メンバーは必要に応じて変更できます。例えば、他のジョブ・タイプを除外するように parmlib メンバーを変更できます。ただし、1つのジョブについてのみオプションを変更する場合は、ユーザー・オプション・ファイル内のオプションを調整してください。

個々のジョブのオプションを変更するさまざまな手法の説明が[オプションの指定位置 ページ 569](#)、[バッチ再分析オプション ページ 154](#)、および[対話式再分析オプション ページ 162](#)に記載されています。

個々のオプションについては、[オプションの説明 ページ 573](#)を参照してください。



**注:** IDICNFUM ユーザー・オプション・モジュールを使用すると、IDICNFxx デフォルト・オプションを置換できません。詳しくは、[ユーザー・オプション・モジュール IDICNFUM ページ 570](#)を参照してください。

次のセクションでは、Fault Analyzer parmlib メンバーに対して指定できる IDICNFxx のその他いくつかのオプションについて簡単に説明します。ただし、Fault Analyzer をインストールし、支給された IVP ジョブのみを実行できればよい場合は、ここに示すいずれのオプションも必要ありません。

本書の残りの部分で、IDICNFxx parmlib メンバーについての言及がある場合、その名前は IDICNF00 または IDICNF&SYSCclone のいずれかを意味します。

## Exclude 処理による分析対象のジョブの制御

このセクションでは、バッチ・ジョブ、開始タスク、または TSO ユーザーのいずれかを参照するために「作業単位」という用語を使用しています。

Exclude オプションおよび Include オプション ([Exclude/Include ページ 590](#)) を使用して、どの作業単位の異常終了を分析対象とするかを選択できます。

どちらのオプションも指定しない場合は、デフォルトによりすべての作業単位が含まれます。

各オプションを任意の回数だけ指定して、特定の作業単位を包含または除外できます。例えば、すべてのジョブを除外してから特定のジョブのみを含める、あるいはその逆に、すべてのジョブを含めて (デフォルト) 一部のジョブのみを除外できます。

選択プロセスは以下のとおりです。

- 作業単位の初期「状態」の選択は、すべてを包含することです。
- 指定された各 Include オプションまたは Exclude オプションが、異常終了した作業単位に照会してテストされます。
  - 異常終了した作業単位の特性と基準が一致した場合、現在の状態が「include」(Include オプションが一致した場合) または「exclude」(Exclude オプションが一致した場合) に設定されます。
  - 基準が一致しない場合、状態は未変更のままです。
- すべての Include オプションおよび Exclude オプションを処理した後の最終状態が「exclude」である場合、異常終了した作業単位は分析から除外されます。この場合、分析レポートは出力されず、障害項目はヒストリー・ファイルに書き込まれません。



**注:**



- リアルタイム分析を除外した場合は、ヒストリー・ファイル項目がないため、後で再分析を行うことはできません。
- 分析の除外を、異常終了した作業単位に対して「透過的」にするには、Quiet オプションを使用することを検討してください。このオプションを使用しない場合、除外が行われたことを示すメッセージが表示されます。

上の説明からわかるように、Include および Exclude オプションを指定する順序は重要です。例えば、以下が指定された場合、ジョブ名または実行クラスにかかわらず、ユーザー ID FRED の下で実行しているバッチ・ジョブはすべて、分析に含まれます。

```
Exclude(TYPE(JOB) NAME(TEST*)) /* Exclude all batch jobs with names
                                starting with TEST */
Exclude(CLASS(Z))             /* Exclude all batch jobs in class Z */
Include(TYPE(JOB) USERID(FRED)) /* Include batch jobs belonging to FRED */
```

ただし、以下に示されているように、最後の2つのオプションが逆になった場合、クラス Z で実行されていれば、FRED のバッチ・ジョブは除外されます。

```
Exclude(TYPE(JOB) NAME(TEST*)) /* Exclude all batch jobs with names
                                starting with TEST */
Include(TYPE(JOB) USERID(FRED)) /* Include batch jobs belonging to FRED */
Exclude(CLASS(Z))             /* Exclude all batch jobs in class Z */
```



**注:** 最終的な「exclude/include」状態は、ENV.EXCLUDE 値を「Y」または「N」に設定することで、分析制御ユーザー出口 (ダンプ登録を含む) によって制御できます。

## 高速 Exclude オプション処理

IDIS サブシステムが始動され、PARM='FASTEXCLUDE' オプションが有効である場合 (このオプションがデフォルト)、IDICNFxx parmlib メンバーからのすべての Include オプションおよび Exclude オプションの指定は、IDIS サブシステム内にキャッシュされます。Fault Analyzer 呼び出し出口は入出力を必要とせずこの情報を入手でき、それ以降、通常のオプション処理が実行されるメインライン IDIDA ロード・モジュールをアタッチする前に、障害をその後の処理から除外すべきかどうかを判別できます。

CICS® は例外ですが、高速 Exclude オプション処理には、IDIOPTS DDname を含んでいないジョブまたは開始済みタスクのみが適格です。CICS® が長期システムである場合、Fault Analyzer は定期的にオプション・データ・セットから Exclude/Include オプションの指定を読み取り、その後のアプリケーション異常終了発生時に CICS® 呼び出し出口によって使用できるように、それらの指定をキャッシュに入れます。

障害が高速で除外されると、[IDI0034I ページ 703](#) のメッセージが発行されます。

高速 Exclude オプション処理のために除外された障害については、IDITRACE 情報は提供されません。

IDICNFxx parmlib メンバー内の Include または Exclude オプションを更新した場合は、それらのオプションが再び読み取られるように、IDIS サブシステムを一度停止してから再始動してください。

高速 Exclude オプション処理を無効にするには、IDIS サブシステム PARM='NOFASTEXCLUDE' オプションを指定します。

## レポートの詳細レベルの制御

[Detail ページ 586](#)で説明されている Detail オプションを使用すれば、分析レポートで必要となる詳細の程度を指定できます。

このオプションは、オペレーター・コンソールに表示される障害の要約には影響しません。

[リアルタイム分析レポート ページ 37](#) 分析レポートの概要と、レポートに対する各種 Detail 値の効果が説明されています。

この値は、再分析用に変更できます。

## ミニダンプのサイズの制限

MaxMinidumpPages オプション ([MaxMinidumpPages ページ 610](#)) により、ヒストリー・ファイルに書き込まれるミニダンプのサイズを制限できます。ミニダンプでのページの数、指定された限度を超えた場合、ミニダンプは書き込まれません。

ミニダンプ・サイズ制限を選択するときには、障害ヒストリー・ファイルに割り振られるスペースを考慮する必要があります。各ミニダンプ・ページは、4KB です。

## リストおよび REXX exec ライブラリーのポインティング

DataSets オプション ([DataSets ページ 576](#)) により、リスト・ファイルまたはサイド・ファイルの保管場所を指定できます。指定は累積的に行われるため、グローバル値を指定してから、これに特定のジョブまたは障害再分析のための追加オプションを指定できます。コンパイラー・リストまたはサイド・ファイルは、ユーザー出口を介して指定することもできます。詳しくは、[コンパイラー・リスト読み取りユーザー出口 ページ 480](#)を参照してください。

リストおよびサイド・ファイルのデータ・セットとは別に、ヒストリー・ファイル、REXX exec ユーザー出口ライブラリーなど、他のデータ・セットを選択するために DataSets オプションを使用することもできます。

## マイナーな SYSLOG メッセージの抑止

通知および警告レベルのメッセージを SYSLOG に書き込まないようにするには、Quiet オプションを使用します。

Quiet オプションについては、[Quiet ページ 624](#)を参照してください。

## Fault Entry List 画面のカスタマイズ

「障害項目リスト」画面に表示される情報をインストール済み環境全体で変更しなければならない場合、これを実現するには、FAISPFopts(HistCols(...)) オプションを使用して、表示される情報の特定の列を必要な順序で指定します。これにより、必要に応じてさらに変更を加えることができる共通のベースが、ユーザーに提供されます。

FAISPFopts オプションについては、[FAISPFopts ページ 598](#)を参照してください。

## 国/地域別環境の指定

Locale オプション ([Locale ページ 609](#)) により、使用するロケールを指定することで、国/地域別環境に対応した表示を可能にします。



## 第 20 章. コンパイラー・リストまたは Fault Analyzer サイド・ファイルの提供

異常終了の分析時に、Fault Analyzer は、コンパイラー・リストまたはサイド・ファイルから取得したソース行情報を提供しようとします。(検索対象のコンパイラー・リストまたはサイド・ファイルのタイプおよび優先順位の詳細については、[コンパイラー・リストまたはサイド・ファイルの検索 ページ 384](#)を参照してください。)

リアルタイム分析中に LangxCapture オプションを有効にして、コンパイラー・リストまたはサイド・ファイルのソース行情報を障害項目に書き込みます。これにより、異常終了したプログラムが後で再コンパイルされても、元のソース情報は使用可能なままになります。[LangxCapture ページ 608](#)を参照してください。

関連ファイルをサイド・ファイルとして検出できないが、コンパイラー・リストが検出された場合には、Fault Analyzer は、サイド・ファイルをリストから自動的に生成し、それを一時データ・セットに配置します。この一時データ・セットは、分析が完了すると削除されます。

リストも検出できない場合は、Fault Analyzer はソース行の詳細を提供できませんが、その場合でも異常終了の分析結果は提供できます。

コンパイラー・リストやサイド・ファイルは、リアルタイム分析中に見つからなかった場合でも、後でヒストリー・ファイル障害項目を再分析するときに提供することができます。コンパイラー・リストやサイド・ファイルは、必要に応じて、異常終了発生後に異常終了プログラムを再コンパイルすることで作成します。ただし、異常終了が発生した時点のロード・モジュールとコンパイラー・リストやサイド・ファイルが一致するようにし、それによって Fault Analyzer ソース・レベル分析を容易にするには、再コンパイルからのオブジェクト・デッキを変更せずに保つことが重要です。

一部のコンパイラー・オプション (XREF、LIST、SOURCE、MAP など) は、コンパイラー・リスト内に指定された情報のみを変更します。それ以外のオプション (OPTIMIZE、SSRANGE など) を指定した場合は、生成されたオブジェクト・コードが変更されます。再コンパイルを行ってソース・サポートを追加するときは、すべてのオプションを元のコンパイルと同じにする必要があります。ただし、リストにのみ影響するオプション (LIST、XREF など) は除きます。必然的に、すべての入力ソース、およびコンパイラー・バージョン/保守レベルが同じになります。

High Level Assembler で作成されたプログラムの場合、Fault Analyzer はアセンブリー・リストを使用しませんが、アセンブラー ADATA オプションを指定するときには、代わりに SYSADATA ファイルが作成されます。アSEMBル時には、このデータ・セットは SYSADATA DDname によって参照されますが、Fault Analyzer 処理時に、IDIADATA DDname から再び読み取られます。

リストまたはサイド・ファイルのいずれかを保管するよう選択できます。ただし、以下の 2 つの理由から、サイド・ファイルの方をお勧めします。

1. 使用するディスク・スペースが少ない。
2. Fault Analyzer はリストを変換しなければならない。サイド・ファイルを使用することで、分析の合計時間を削減できます。

コンパイラー・リストまたはサイド・ファイルのために PDS または PDSE データ・セットを設定している場合は、分析が必要と思われるプログラムがこれに配置されていることを確認してください。また、コンパイラー・リストは順次データ・セットに保管できます。

Fault Analyzer は、所定の DDname に別個に指定されたデータ・セットをすべて検索して、最適にマッチングするコンパイラー・リストまたはプログラムのサイド・ファイルを選択します。したがって、同じプログラムの異なるバージョン (例えば、開発バージョン、テスト・バージョン、実動バージョン) を含むコンパイラー・リストまたはサイド・ファイル・データ・セットが、すべて同時に提供されることがあります。

コンパイラー・リストまたはサイド・ファイルの命名についての情報は、[コンパイラー・リストまたはサイド・ファイルの命名 ページ 382](#)を参照してください。

## COBOL 報告書作成プログラム・プリコンパイラー

COBOL 報告書作成プログラム・プリコンパイラー (プログラム番号 5798-DYR) を使用している場合、COBOL コンパイラー EXIT オプションを介して起動するのではなく、スタンドアロンのプリコンパイラーとして実行することが重要です。これを行わない場合、ソース・コード・ステートメントの障害点を特定するために Fault Analyzer が必要とする情報が、コンパイラー・リストから欠落する可能性があります。

COBOL 報告書作成プログラム・プリコンパイラーを COBOL コンパイラー出口として使用する場合、以下の状態になる可能性があります。

- COBOL コンパイラー・リスト・ファイルをサイド・ファイルに変換しようとしているときに、IDILANGX が戻りコード 3114 を戻す。
- 障害分析時に以下のメッセージが発行される。
  - IDISF8100S COBOL LISTING file contains NO recognized records
  - IDISF8132S Input or Output file format invalid
- 障害点ソース行を判別できない。

## IDILANGX の必須コンパイラー・オプション

LANGX サイド・ファイルを作成するには、IPVLANGX ユーティリティを使用します。IPVLANGX ユーティリティは ADFz Common Components に付属しています。IDILANGX は IPVLANGX の別名です。IPVLANGX ユーティリティの実行について詳しくは、『*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*』の「*IPVLANGX - コンパイラー・リストからサイド・ファイルへの変換ユーティリティ*」の章を参照してください。

Fault Analyzer に適したリストまたはサイド・ファイルを作成するために必要なコンパイラー・オプションは、次のとおりです。コンパイラー生成 TEST(SEPARATE) SYSDEBUG サイド・ファイルを使用する場合、これらのオプションは関係ありません。

C:	AGGREGATE LIST NOOFFSET NOOPT (注 1) SOURCE
----	--

	<p>XREF NOIPA</p>
C++:	<p>ATTRIBUTE (注 4) LIST LONGNAME NOOFFSET NOOPT (注 1) SOURCE XREF NOIPA</p>
OS/VS COBOL:	<p>DMAP NOLST NOOPT (注 1) PMAP, NOCLIST (注 2) SOURCE VERB XREF または SXREF</p>
OS/VS COBOL 以外の COBOL コンパイラー:	<p>LIST, NOOFFSET (注 2) NOOPT (注 1) MAP SOURCE XREF(SHORT) (注 3)</p>
Enterprise PL/I:	<p>AGGREGATE ATTRIBUTES(FULL) NOBLKOFF LIST (注 5) MAP NEST NONUMBER OFFSET NOOPT (注 1) OPTIONS SOURCE</p>

	STMT XREF(FULL)
Enterprise PL/I 以外の PL/I コンパイラー:	AGGREGATE ATTRIBUTES(FULL) ESD LIST (注 5) MAP NEST NOOPT (注 1) OPTIONS SOURCE STMT XREF(FULL)
アセンブラー:	ADATA



注:

1. NOOPT が推奨されますが、OPTIMIZE (C の場合の OPT(1) または OPT(2) も含む) の使用が可能であり、その場合、コンパイラーは、コンパイルされたコードにステートメント番号をマージし、再配置します。Fault Analyzer 分析は、最適化されたコンパイラー・リストから判別できるものに限定されます。これは、Fault Analyzer レポートに影響のないものから、失敗したソース行の不正確な識別番号までにわたります。ソース行番号は、通常は近いものになりますが、OPTIMIZE では必ずしも正確ではありません。これは、最適化プロセス中のコンパイラーの再配列またはソース・ステートメントの除去に依存します。ストレージが割り当てられていない場合、つまり、値がレジスター内のみある場合には、データ・フィールドの値は表示できません。OPTIMIZE が有効な場合、OFFSET はコードの移動に対応しないので、LIST および NOOFFSET を使用します。
2. LIST および NOOFFSET (OS/VS COBOL の場合は PMAP および NOCLIST) が推奨されますが、NOLIST および OFFSET (OS/VS COBOL の場合は NOMAP および CLIST) の使用は可能です。この場合、Fault Analyzer は、コンパイラー・リストがストレージにあるものと完全には一致しなくても、ユーザーに警告を出すことができません。
3. XREF(SHORT) は最小必要要件です。XREF(FULL) も可能で、問題はありません。



4. ATTRIBUTE は最小必要要件です。ATTRIBUTE(FULL) も可能で、問題はありません。
5. Enterprise PL/I の STATIC EXTERNAL 変数およびパラメーターを正しく報告するには、LIST が必要です。ただし、LIST オプションが有効な場合でも、PL/I V6 コンパイラーは 64 ビット・パラメーターを使用可能にしません。64 ビット・パラメーターは PL/I V5 で使用できます。

## TEST オプションの考慮事項

すべてのコンパイラーで、TEST オプションを使用すると、サイド・ファイルから入手できる情報に加えて、プログラム情報が提供されます。

COBOL プログラムのコンパイル時に TEST(NONE,SYM,SEPARATE) を使用する場合、または Enterprise PL/I プログラムのコンパイル時に TEST(STMT,SYM,NOHOOK,SEPARATE) を使用する場合、Fault Analyzer での使用に適した SYSDEBUG ファイルが書き込まれます。(Fault Analyzer が SYSDEBUG ファイルを検索する方法の詳細については、[コンパイラー・リストまたはサイド・ファイルの検索 ページ 384](#)を参照してください。)コンパイラー・リスト、またはコンパイラー・リストから作成された LANGX サイド・ファイルの代わりに SYSDEBUG ファイルを使用する場合は、z/OS® Debugger および Fault Analyzer で使用するために保存しておく必要があります。

COBOL SYSDEBUG サイド・ファイルは一般的に、Fault Analyzer IDILANGX サイド・ファイルより約 30% 少ない DASD スペースを占有します。

Enterprise PL/I SYSDEBUG サイド・ファイルは、スタンドアロンのデバッグ援助機能ではありません (これに相当する COBOL のサイド・ファイルとは異なります)。ロード・モジュールには、常にステートメント番号テーブルが含まれ (ソース行オフセットを提供)、またプログラムに GET/PUT DATA ステートメントが存在する場合には、シンボル情報も含まれています。

## DEBUG オプションの考慮事項

XL C/C++ コンパイラーで DEBUG オプションを使用して DWARF デバッグ情報を書き込みます。このデバッグ情報は MVS™ データ・セットまたは HFS ファイルに書き込むことができます。Fault Analyzer はロード・モジュール内の情報から DWARF ファイルを見つけます。

ロード・モジュール内のすべてのコンパイル・ユニットに対応する DWARF ファイルは、ソースと組み合わせることで、1つの MDBG サイド・ファイルにすることができます (このためには、CDADBGLD ユーティリティを使用します)。DWARF と同様に、MDBG ファイルは MVS™ データ・セット内または HFS ファイル内に置くことができます。MVS™ データ・セット内に保管された MDBG ファイルは、IDISYSDB DD によって位置指定されます。HFS 内の MDBG サイド・ファイルは自己位置指定タイプです。そのため、以下の事項を守る必要があります。

- MDBG ファイルは、元の DWARF と同じ HFS ディレクトリー内に常駐する必要があります。
- MDBG ファイルの名前は、そのファイルの作成元のロード・モジュールと同じであり、ファイル拡張子に .mdbg を付ける必要があります。

- MDBG ファイル名および拡張子は、大/小文字混合で構成することはできません。ファイル名と拡張子を大文字にするか小文字にするかは、当該コンパイル単位の元の DWARF ファイルのファイル名と拡張子に基づいて決まります (したがって、ファイル名が大文字でありながら、ファイル拡張子が小文字であったり、その逆になる可能性もあります)。

## COBOL NOTEST(DWARF、SOURCE) 使用時の制限

PTF UI60153 がインストールされておらず、COBOL プログラムを NOTEST(DWARF、SOURCE) を使用してコンパイルする場合、コンパイラーには DWARF の相互参照情報が含まれません。

結果として、データ・フィールド値および宣言は、COBOL Explorer と同様に、レポートの「Synopsis」および「Event Details」セクションから欠落します。

## PL/I TEST (NOSEPARATE NOSOURCE) 使用時の制限

NOSOURCE が指定されていて、プログラム・ソースが使用できない場合でも、ソース行番号とストレージ域が報告されません。

TEST(SEPARATE) および TEST(NOSEPARATE SOURCE) は同等です。前者は別の SYSDEBUG サイド・ファイルにソースを配置し、後者はプログラム・オブジェクト内に配置します。

## コンパイラー・リストまたはサイド・ファイルの命名

コンパイラー・リストまたはサイド・ファイルを順次データ・セットに保管、または PDS または PDSE データ・セットのメンバーとして保管します。

コンパイラー・リストまたはサイド・ファイルを PDS または PDSE データ・セットに保存する場合、使用するメンバー名はアプリケーションのプログラミング言語によって異なります。

- 単一 CSECT アセンブラー・プログラムの場合 (つまり、アセンブリーごとに 1 つの CSECT) は、CSECT 名またはロード・モジュール名を使用します。ロード・モジュール名よりも CSECT 名を使用することをお勧めします。これは、メンバーの位置指定のオーバーヘッドが少ないためです。また、アセンブラー CSECT のみのサブルーチンの場合には、ロード・モジュール名を使用しないでください。
- 複数 CSECT アセンブラー・プログラムの場合 (つまり、アセンブリーごとに複数の CSECT) は、ロード・モジュール名を使用します。
- COBOL プログラムの場合は、アプリケーション・プログラムの名前を使用します。
- PL/I プログラムの場合は、アプリケーション・プログラムの 1 次エントリー・ポイント名または CSECT 名を使用します。
- z/OS XL C および C++ プログラムの場合は、アプリケーション・プログラムの CSECT 名を使用します。

別の名前で保管すると、Fault Analyzer はコンパイラー・リストまたはサイド・ファイルを検出できなくなります。

コンパイラー・リストまたはサイド・ファイルが順次データ・セットに保管される場合、およびデータ・セット名が、データ・セット名の一部にプログラム名を許可する規則に従う場合、DataSets オプションのこのデータ・セットは、[DataSets オプションのデータ・セット名の置換シンボル ページ 580](#)に説明されているように、変数置換を使用して容易に指定できます。

## Fault Analyzer 用の CSECT の命名

ソース・コード情報を使いやすくするために、Fault Analyzer は CSECT 名を、提供されるコンパイラー・リストまたはサイド・ファイルと一致させることができる必要があります。

これを可能にするには、すべての CSECT に名前を付ける必要があります。ほとんどの高水準言語で記述されたプログラム内の CSECT の名前は、自動的に割り当てられますが、C で記述されたプログラムには特別の要件が適用されます。これらの要件に従わないと、このようなタイプのプログラムではソース・コード情報が判別されません。

### C プログラムの CSECT 命名要件

IDILC または IDILANGX 連結から C プログラムへの自動ソース・サポートを有効にする。これは CSECT が次の `#pragma` ステートメントを使用して命名されるという要件です。

```
#pragma csect(code,"csect_name")
```

ここで `csect_name` は、PDS または PDSE を使用する場合、コンパイラー・リスト名または LANGX ファイルのメンバー名です。これにより、サイド・ファイル検索が自動的にコンパイラー・リストを検索できます。

z/OS® XL C および C++ プログラムを Fault Analyzer と使用するための準備についての情報は、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* の “Quick start guide for compiling and assembling programs for use with the ADFz family of products” の章を参照してください。

C プログラムが `#pragma csect` オプションを使用してコンパイルされなかった場合を処理する。次の 2 つのサンプル EXEC がデータ・セット IDI.SIDISAM1 で提供され、ソースのプロビジョニングを容易にします。

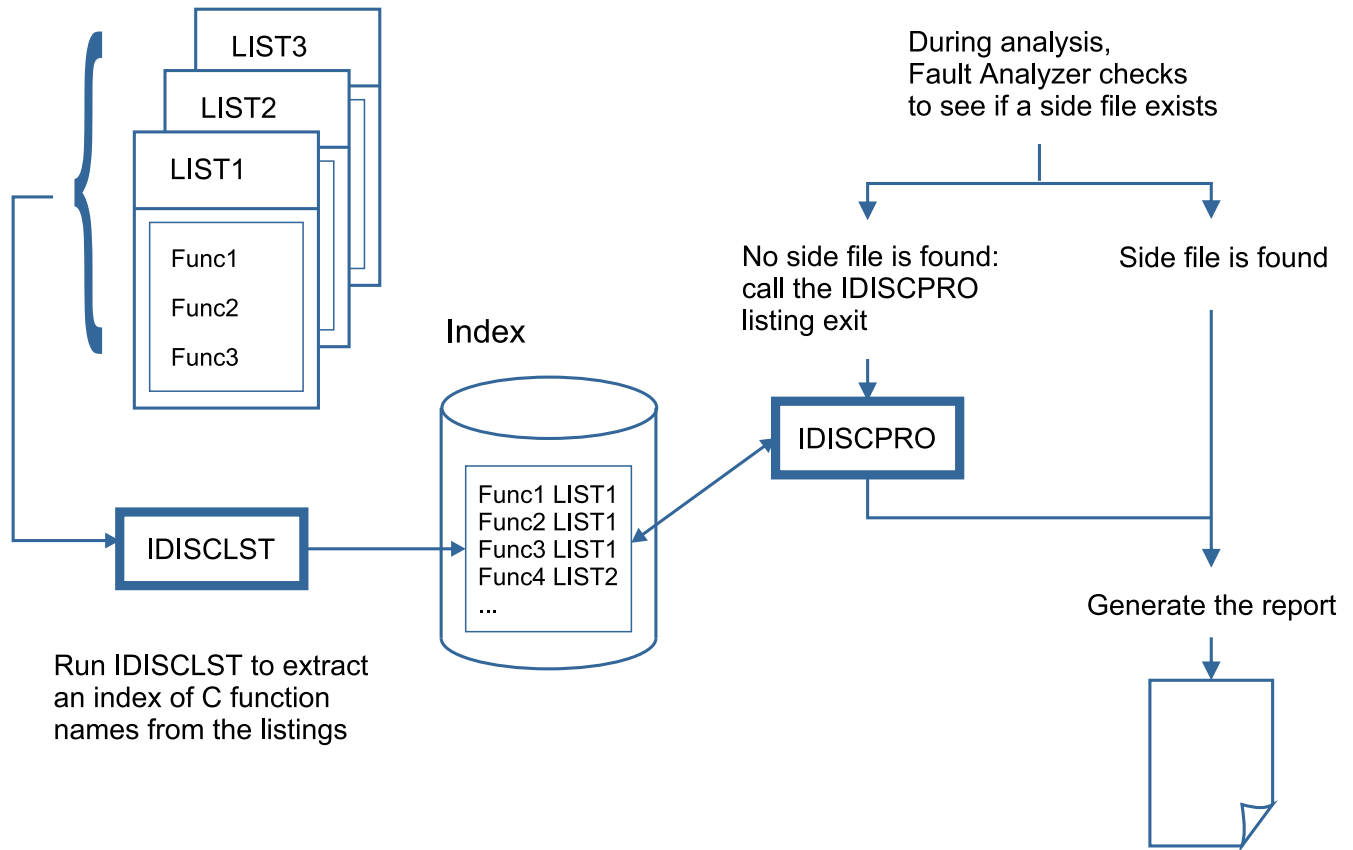
#### IDISCLST

この EXEC を使用して、C コンパイラー・リストを処理し、C 関数名を PPA1 から抽出し、関数名の索引およびその一致するリスト・データ・セットを作成することができます。

#### IDISCPRO

この EXEC は Fault Analyzer リスト出口です。これは IDISCLST により作成された索引ファイルをスキャンし、指定した関数の一致するリスト・データ・セットの名前を返します。

図 200. #pragma csect を使用してコンパイルされなかった C プログラムからの関数名の抽出  
C compiler listings



## コンパイラー・リストまたはサイド・ファイルの検索

Fault Analyzer は、以下のソース候補のうち、最初に一致したコンパイラー・リストまたはサイド・ファイルを選択します。

### 1. TEST オプション・データ:

- プログラムが TEST オプション (COBOL または Enterprise PL/I の TEST(SEPARATE) 以外) を指定してコンパイルされた場合、一時 Fault Analyzer サイド・ファイルは、ロード・モジュールに含まれているデバッグ情報から生成されます。
- COBOL V4 または Enterprise PL/I プログラムが TEST(SEPARATE) を指定してコンパイルされた場合、関連する SYSDEBUG サイド・ファイル・データ・セット名は、コンパイラーによって作成されてロード・モジュールに入れられたデバッグ情報から取得されます。

推奨される言語依存サブオプションについては、[TEST オプションの考慮事項 ページ 381](#)を参照してください。

- TEST(SEPARATE) を指定してコンパイルされた COBOL V4 プログラムの場合、SYSDEBUG サイド・ファイル・データ・セット名が、オプションの COBOL IGTIUXB 出口への入力として使用されます。この出口は、異なるサイド・ファイル・データ・セット名を返すことがあります。



- COBOL V6 プログラムが TEST(SEPARATE) を指定してコンパイルされ、DSNAME サブオプションが指定されている場合、SYSDEBUG サイド・ファイル・データ・セット名はプログラム・オブジェクトに保管されます。それ以外の場合では、SYSDEBUG サイド・ファイル・データ・セット名は IDISYSDB 連結、コンパイラー・リスト読み取りユーザー出口、COBOL IGZIUXB 出口または EQUUEDAT から取得されます。



**注:** CICS® では、Fault Analyzer は生成された TCB 内で CICS® コマンド・アクセスを使用せずに実行されるため、バッチ形式の EQUUEDAT が使用されます。

## 2. z/OS® Debugger EQUUEDAT 出口 (オプション)。

EQUUEDAT 出口の詳細については、「*IBM®z/OS® Debugger カスタマイズ・ガイド*」を参照してください。

TEST(SEPARATE) を指定してコンパイルされた COBOL または Enterprise PL/I プログラムの場合、ロード・モジュールからの SYSDEBUG サイド・ファイル・データ・セット名が、EQUUEDAT 出口への入力として使用されます。

IDILANGX またはコンパイラー・リストを検索する場合、入力サイド・ファイル名は提供されません。

この出口は、異なるサイド・ファイル・データ・セット名を返すことがあります。



**注:** CICS® では、Fault Analyzer は生成された TCB 内で CICS® コマンド・アクセスを使用せずに実行されるため、バッチ形式の EQUUEDAT が使用されます。

3. COBOL または Enterprise PL/I プログラムの場合は、IDISYSDB DDname によって提供される SYSDEBUG データ・セットが検索されます。XL C/C++ プログラムの場合、IDISYSDB DDname によって提供される MDBG データ・セットが検索されます。データ・セットは個別に検索されます。つまり論理連結として扱われません。

## 4. Fault Analyzer サイド・ファイル:

- a. Compiler Listing Read サイド・ファイルの要求によって呼び出された Fault Analyzer ユーザー出口。
- b. Fault Analyzer DDname によって提供された IDILANGX のサイド・ファイル・データ・セットが検索されます。データ・セットは個別に検索されます。つまり論理連結として扱われません。

## 5. コンパイラー・リスト:

- a. コンパイラー・リスト (またはアセンブラー SYSADATA ファイル) の要求によって呼び出された Compiler Listing Read ユーザー出口。
- b. 適切な DDname によって提供される言語特有のコンパイラー・リスト・データ・セットが検索されます (例えば、IDILCOB など)。データ・セットは個別に検索されます。つまり論理連結として扱われません。

コンパイラー・リストまたはサイド・ファイルが、プログラムをマップするために Fault Analyzer によって使用されるのに適しているかどうかの判断は、以下の事項によって決まります。

### • NOTEST コンパイラー・オプションが使用される場合

Fault Analyzer がプログラムのコンパイラー・リストまたは LANGX サイド・ファイルを検出すると、使用するプログラミング言語に応じた多くのテストが実行されて、ファイルがロード・モジュールに一致することが確認されず。

### ◦ すべての言語

Fault Analyzer はコンパイラ出力のアセンブラー・リスト・セクションから、最後のいくつかのアセンブラー命令を抽出し、それらをロード・モジュールと比較します。これらのアセンブラー命令は、リストに従ってロード・モジュール内の正しいオフセットにある必要があります。これらの命令がロード・モジュール内の正しいオフセットで見つからない場合は、この検査が失敗し、コンパイラー・リストまたはサイド・ファイルが使用されなくなり、ソース・レベルの情報は提供されなくなります。

### ◦ COBOL 特有のテスト (OS/VS COBOL を除く)

COBOL の場合は、追加の 4 つの長さ値がリストから抽出されます。これらは、コンパイラー・リストの以下の 4 行で見つかった、TGT 長、WORKING-STORAGE 長、Data Division ステートメント数、および Procedure Division ステートメント数です。

```
TGT      WILL BE ALLOCATED FOR nnnnnnnn BYTES
WRK-STOR WILL BE ALLOCATED FOR nnnnnnnn BYTES
Data Division statements = nnnnnnn
Procedure Division statements = nnnnnnn
```

これらの 4 つの値は、ロード・モジュールで見つかった値と比較され、一致しない場合は、ソース・レベルの情報の提供にコンパイラー・リストまたはサイド・ファイルは使用されません。

### • TEST コンパイラー・オプションが使用される場合

COBOL SYSDEBUG ファイルの場合、日時の比較は「シグニチャー」検査に置き換えられます。これにより、変更のないプログラムの再コンパイルが可能になります。シグニチャー検査が不合格だった場合でも、コンパイラー・リスト・ミスマッチ・ポップアップ画面の指示に従うことで、対話式再分析では SYSDEBUG ファイルを引き続き使用できます。

COBOL プログラム (Enterprise COBOL V4 までを含むが、VS COBOL II は除外) の場合、コンパイラー・リスト・データ・セット名は、コンパイラーによって格納されたロード・モジュール内のデバッグ情報から取得されます。この場合、リスト内の日時が、ロード・モジュール内のコンパイル日時と照合されます。コンパイラー・リスト・ファイルが移動された (または VS COBOL II である) 場合、コンパイラー・リストは IDILCOB DD、EQUAEDAT、またはコンパイラー・リスト読み取りユーザー出口を使用して指定できます。

TEST(NOSOURCE) または NOTEST(DWARF) でコンパイルされた Enterprise COBOL V5 (またはそれ以降) のプログラムの場合、ソース情報はプログラム・オブジェクトに含まれず、コンパイラー・リスト・データ・セット名は DWARF で使用できません。ただし、これらのプログラムのコンパイラー・リストは、IDILCOB DD、EQUAEDAT、またはコンパイラー・リスト読み取りユーザー出口を使用して指定できます。

Enterprise PL/I SYSDEBUG ファイルの場合、日時の比較は、一連の「保水性」検査に置き換えられます。これにより、SYSDEBUG ファイルがストレージ内のロード・モジュールで使用可能か判断されます。いずれかの保水性検査が失敗した場合、SYSDEBUG ファイルは使用できません。

データを使用する前に Fault Analyzer がこれらの検査を実行する理由は、コンパイラー・リストまたはサイド・ファイルが正しくないと、誤ったレポート情報が生成される可能性があるためです。

対話式障害再分析時に COBOL、PL/I、C/C++、またはアセンブラー・プログラムに適合したコンパイラー・リストまたはサイド・ファイルが検出されなかった場合、Fault Analyzer は、オプションとしてコンパイラー・リスト、Fault Analyzer サ

イド・ファイル、または COBOL SYSDEBUG サイド・ファイルの場所を指定するようにユーザーに求めます。このプロンプトの詳細については、[コンパイラー・リストまたはサイド・ファイルのプロンプト ページ 239](#)を参照してください。

## IDITRACE 情報

IDITRACE DDname をジョブに追加すると、Fault Analyzer によりトレース情報が提供されます。この情報は、特定のコンパイラー・リストまたはサイド・ファイルが選択または拒否された原因を理解するのに役立ちます。例:

```
//IDITRACE DD SYSOUT=*
```

CICS® 環境でこのトレースを有効にする別の方法については、[IDITRACE CICS において ページ 407](#)を参照してください。「**CTL.IDITRACE data area**」フィールドによる動的トレースについては、[分析制御ユーザー出口 ページ 475](#)を参照してください。

「**CTL.IDITRACE data area**」フィールドによる動的トレースを使用しない場合、トレース情報は IDITRACE DDname 宛先に書き込まれます。コンパイラー・リストまたはサイド・ファイルの検索トレース情報の例は、次のとおりです。

### 図 201. コンパイラー・リスト/サイド・ファイルの検索トレースの例

```
Listing/Side File search trace for IDISCBL1
Execution program IDISCBL1 compile date 2013/02/18 time 12:17:24
Rejected - DA.LISTING.COBOL(IDISCBL1)
Failed -
  OPEN error, member not found.
Rejected - NWILKES.$$TEMP$$.LISTINGS(IDISCBL1) Built 2013/02/18 12:17:41
Failed -
  COBOL Working Storage length mismatch - load x'2A' listing x'32'
  COBOL Data Division Statements mismatch - load 6 listing 7
Passed -
  COBOL TGT lengths match load and listing both 148
  COBOL Procedure Division Statements match, both 10
  Object code length check passed.
Accepted - NWILKES.IVPCB.LISTINGS(IDISCBL1) Built 2013/02/18 12:17:24
Passed -
  COBOL TGT lengths match load and listing both 148
  COBOL Working Storage lengths match load and listing both x'2A'
  COBOL Data Division Statements match, both 6
  COBOL Procedure Division Statements match, both 10
  Object code length check passed.
```

対話式再分析を実行するときに、Compiler Listing Not Found (コンパイラー・リストが見つかりません) 画面からオプションを選択することにより、IDITRACE DDname を割り振る必要なしに、コンパイラー・リストまたはサイド・ファイルの検索に関する IDITRACE 情報が入手可能になります。[コンパイラー・リストまたはサイド・ファイルのプロンプト ページ 239](#)を参照してください。

ただし、バッチで生成されたものと同様の完全な IDITRACE 情報が必要な場合、次の手順に従います。

1. Fault Analyzer ISPF インターフェース障害エントリー・リスト画面を表示します。
2. 基本コマンド行から、次のコマンドを発行します。
  - TSO FREE DD(IDITRACE)
  - TSO ALLOC DD(IDITRACE) DSN(*dsm*) DSORG(PS) SPACE(2,1) TRACKS LRECL(137) RECFM(V,B) NEW (ここで DSN はトレース・データ・セットの名前です)

3. 故障エントリーの対話式再分析を実施します。
4. PF3 を故障エントリー・リスト画面に戻します。
5. オプションとして、基本コマンド行から次のコマンドを発行できます。
  - TSO FREE DD(IDITRACE)

## IDIXSFOR コンパイラー・リスト/サイド・ファイルの検索出口およびオーバーライド出口の使用

IDIXSFOR 出口が使用可能であれば、Fault Analyzer はその出口を呼び出し、コンパイラー・リストまたはサイド・ファイル・データの場所をプログラマチックに指定できるようにします。

入力パラメーター・リストは EQUEDAT で使用されるものと似ていますが、これに加えて FORCE パラメーターが末尾に追加されています。FORCE パラメーターを使用すると、オブジェクト・コードの最後の 12 命令が適合しない場合でも、サイド・ファイルを受け入れるように Fault Analyzer に指示できます。また、FORCE パラメーターを使用して、現行コンパイル単位に対してこれ以上のサイド・ファイル処理を実行しないように要求することもできます。

## IDIXSFOR 呼び出し

IDIXSFOR のエントリーおよび戻りの指定について、以下で説明します。

### エントリーの指定

IDIXSFOR へのエントリーについて、レジスターの内容は次のとおりです。

#### レジスター

##### 内容

1

入出力パラメーター・リストのアドレス (下記参照)。

13

72 バイトのレジスター保管域のアドレス

14

戻りアドレス

15

IDIXSFOR のエントリー・ポイント・アドレスです。

## 入力パラメーター・リスト

レジスター 1 には、アドレスのリストで構成されるパラメーター・リストのアドレスが含まれていなければなりません (OS リンケージ)。パラメーター・リスト内の各アドレスは 1 つのパラメーターを指します。

表 11. IDIXSFOR 入力パラメーター

パラメーター番号	長さ	入出力	説明
1	パラメーター #2 を参照	入出力	<p>サイド・ファイル・データ・セット名</p> <p><b>入力</b></p> <p>コンパイル・オプションが TEST(SEPARATE) の場合、Fault Analyzer が 入力データ・セット名を提供することがあります。</p> <p><b>出力</b></p> <p>出口から提供される試行対象の新規データ・セット名/パス。この名前の長さは 1024 バイトを超えてはなりません。このデータ・セットが PDS または PDSE の場合は、メンバー名が含まれている必要があります。</p>
2	4	入出力	<p>サイド・ファイル・データ・セット名 (パラメーター #1) の長さ</p> <p><b>入力</b></p> <p>入力サイド・ファイル名の長さ。</p> <p><b>出力</b></p> <p>新規データ・セット名/パスの長さ。</p>
3	4	入力	<p>言語コード</p> <p><b>値</b></p> <p><b>意味</b></p> <p><b>3</b></p> <p>C ソース</p> <p><b>4</b></p> <p>COBOL</p> <p><b>7</b></p> <p>アセンブラー、OS/VS COBOL、非 LE VS COBOL II</p> <p><b>10</b></p> <p>PL/I (Enterprise PL/I を除く)</p> <p><b>11</b></p> <p>Enterprise PL/I</p>

表 11. IDIXSFOR 入力パラメーター (続く)

パラメーター番号	長さ	入出力	説明
			<b>35</b> FORMAT(DWARF) を指定してコンパイルされた C サイド・ファイル
			<b>37</b> C MDBG サイド・ファイル
			<b>41</b> VS COBOL II (LE の下)
4	パラメーター #5 を参照	入力	コンパイル単位名
5	4	入力	コンパイル単位名 (パラメーター #4) の長さ
6	パラメーター #7 を参照	入力	ロード・モジュール名
7	4	入力	ロード・モジュール名 (パラメーター #6) の長さ
8	パラメーター #9 を参照	入力	ロード・ライブラリー名
9	4	入力	ロード・ライブラリー名 (パラメーター #8) の長さ
10	4	出力	強制戻りオプション・ワード
			<b>値</b>
			<b>意味</b>
			<b>0</b> 強制オーバーライドなし
			<b>1</b> 最新 12 個のオブジェクト命令の受け入れ不一致を強制
			<b>-1</b> このコンパイル単位のサイド・ファイル処理を無視

## 戻り指定

IDIXSFOR から戻るときのレジスターの内容は次のとおりです。

### レジスター

#### 内容

0 ~ 1

未定義。

2 ~ 14

未変更。

15

未定義。

## 例 (アセンブラー)

サンプル IDIXSFOR ユーザー出口が、メンバー IDIXSFOR としてデータ・セット IDI.SIDISAM1 に提供されます。

## コンパイラー・リストおよびサイド・ファイルの属性

コンパイラー・リストおよびサイド・ファイルは、以下の属性を使用して割り振る必要があります。

### DDname

#### 属性

#### IDIADATA

順次データ・セットまたは PDS または PDSE、RECFM=VB、LRECL≥8188

#### IDILC

次の属性いずれかを含む、順次データ・セットまたは PDS または PDSE:

- RECFM=VB または VBA および LRECL≥137
- RECFM=FB/FBA および LRECL=133

#### IDILCOB

順次データ・セットまたは PDS または PDSE、RECFM=FBA、LRECL133

#### IDILCOBO

順次データ・セットまたは PDS または PDSE、RECFM=FBA、LRECL133

#### IDISYSDB

順次データ・セットまたは PDS または PDSE、RECFM=F または FB、80≤LRECL≤1024

#### IDILANGX

順次データ・セットまたは PDS または PDSE、RECFM=VB、LRECL≥1562

#### IDILPLI

順次データ・セットまたは PDS または PDSE、RECFM=VBA、LRECL≥125

#### IDILPLIE

順次データ・セットまたは PDS または PDSE、RECFM=VBA、LRECL≥137

Fault Analyzer がコンパイラー・リストまたはサイド・ファイルを読み取れるようにするため、コンパイラー・リストまたはサイド・ファイルを一時データ・セットとして (例えば、JCL で `&&dsname` タイプのデータ・セット名を使用して) 割り振らないでください。

ディスク・スペース節約のために、コンパイラー・リストを ISPF パック・フォーマットで保管できます。この保管は、ファイルの ISPF エディット内から PACK ON オプションを使用して行います。ISPF パック・フォーマットは、IDILANGX または IDIADATA データ・セットでは許可されません。

## CSECT マッピングに IDIRLOAD DDname の使用

Fault Analyzer は通常、8 桁の大文字から成る CSECT 名を使用してコンパイラー・リストまたはサイド・ファイルを検索します。つまり、CSECT 名の解決が障害点ソース行/ステートメントなどのソース・レベル情報を提供するための前提条件です。

Fault Analyzer はリアルタイム処理時に MVS™ バインダー・プログラムを自動的に起動し、アプリケーション・ロード・モジュール内の CSECT のマッピングを実行します。CSECT の判別は、通常、パフォーマンス上の理由から MVS™ オペレーティング・システムや Language Environment® に属するロード・モジュールなど、非アプリケーション・ロード・モジュールでは実行されません。

また、**Fault Entry List** 画面から **File > Analyze MVS Dump Data Set** を使用して MVS™ ダンプ分析を実行する場合、CSECT マッピングは一般的に利用できません。その理由は、ロード・モジュールの当初の読み込み元であるデータ・セット名を決定できない、あるいはダンプが書き込まれたシステムと分析されるシステムが異なるからです。

ロード・ライブラリーの連結を IDIRLOAD DDname に事前割り振りすることで、非アプリケーション・ロード・モジュールまたは MVS™ ダンプ分析の場合のあらゆるロード・モジュールの CSECT マッピングを使用可能に設定できます。この方法による IDIRLOAD DDname の割り振りは、例えば、TSO ALLOCATE コマンドをダンプ分析前に使用するか、あるいは分析制御ユーザー出口から IDIALLOC REXX コマンドを使用することで実行可能です。

分析制御ユーザー出口を使用する場合、代替 DDname を使用するオプションもあります。代替 DDname は ENV.IDIRLOAD\_DD フィールドによって識別する必要があります。この方法は、別々の ISPF 分割画面セッションで複数の Fault Analyzer ダンプ分析を同時に実行する場合に利用できます。最初の RFR 障害項目再分析時に代替 DDname を使用する方法については、[ENV - 共通出口環境情報 ページ 650](#)に記載されている IDIRLOAD\_DD フィールドの説明を参照してください。

Fault Analyzer は、再分析中に、前述のいずれかの理由でロード・モジュールに関するリンク・エディット情報を持っていない場合、IDIRLOAD 連結を使用して同じ名前のモジュールを検索し、IEWBIND を使用して CSECT 名情報を取り出します。この方法により、これらの CSECT に対する潜在的な IDILANGX ソース・マッピングが容易になります。

IDIRLOAD DDname に割り振られたロード・ライブラリーのいずれかがマッピング対象のロード・モジュールと一致することが大切であり、一致しない場合は CSECT およびソース行の判別に誤りが生じる可能性があります。

PL/X プログラムのソース・レベル分析を容易にするには、`PARM='mbr_name (PLX'` を使用して Fault Analyzer を呼び出し、SYSADATA および SYSLOGIC 入力データを指定して作成されたマッチング用の IDILANGX サイド・ファイルを用意する必要があります。



## IDIDATST サイド・ファイルの使用可能性テスト・ユーティリティー

ユーティリティー・プログラム IDIDATST は、ロード・モジュール内のプログラムに対して適合するソース・マッピング・サイド・ファイルが使用可能かどうかの判別を支援するために提供されています。通常、既存のコンパイル・リストやサイド・ファイル再作成プロセスによって使用可能なサイド・ファイルが生成されているかどうかを検査する場合は、プログラム・スイート内で異常終了が発生するとインストール・フェーズにおいて IDIDATST が使用されます。この方法により、適合するコンパイル・リストまたはサイド・ファイルがプログラムに対して使用可能であるかどうかを異常終了の発生前に判別できます。

入力パラメーターを使用しない IDIDATST の基本操作では、最初の `//STEPLIB DD` 連結データ・セットのロード・モジュールにおいて、すべてのプログラムがスキャンされます。各プログラムまたは各 CSECT は、サイド・ファイル・データ・セットの対応する IDICNFxx DataSets オプション・リスト内に、適合する使用可能なサイド・ファイルがあるかどうかについてテストされます。さらに、ジョブ・ステップに `//IDITRACE DD` がある場合は、検索レポートがそこに書き込まれます。COBOL の `//IDILCOB` など、各言語のすべてのサイド・ファイル DD ステートメントを IDIDATST ジョブ・ステップに追加できます。対応するプログラム言語について、IDICNFxx オプション・データ・セットを検索する前に、まずこれらのステートメントが検索されます。

IDIDATST ユーティリティーを呼び出すサンプル・ジョブは以下のようになります。

```
//IDIDATST JOB ...
//RUN      EXEC PGM=IDIDATST,PARM='parms'
//STEPLIB DD DISP=SHR,DSN=load_lib_dsn
//IDICSV   DD  SYSOUT=*
```

`EXEC PGM=IDIDATST` JCL ステートメントに追加できるオプション PARM フィールドの形式は次のとおりです。

### 図 202. Syntax

```
▶▶ PARM=PGM= load_module_name '▶▶
```

各部の意味は以下のとおりです。

#### ***load\_module\_name***

最初の `//STEPLIB DD` 連結データ・セット内のロード・モジュールの名前。指定されたロード・モジュール名は、8 文字より短い場合、すべてのロード・モジュール名の同じ先頭文字に対して突き合わせを行うフィルターとして使用されます。

例えば、次のように指定することは

```
PARM='PGM=KACBB092'
```

と指定すると、ロード・モジュール KACBB092 のみがテストされ、

```
PARM='PGM=KAC'
```

と指定すると、最初の STEPLIB データ・セットで KAC で始まる名前のロード・モジュールがすべてテストされます。

IDIDATST の各実行から統合レポートが作成され、プログラム検索ごとの結果を 1 行で示す形で `//IDICSV DD` に書き込まれます。`IDICSV` 出力のフォーマットはコンマ区切り値 (CSV) です。これは、例えば、スプレッドシート・プログラムにデータをインポートする場合に役立ちます。同一データのコピーを、列位置合わせフォーマット (エディター・レビューに、より適している) で取得するには、`//IDIPLAT DD` を追加して、空白で CSV データを位置合わせします。

STEPLIB の IDIDATST プロセスは、後続のロード・モジュールに発生するプログラムまたは CSECTS を再テストしません。

IDIDATST は、異常終了条件下で行われるサイド・ファイル・テストを正確に複製することはできません。

オプションの `//IDINOTST DD` を追加すれば、サイド・ファイル検査から除外するプログラム名の接頭部の空白区切りリストを指定できます。例:

```
//IDINOTST DD *
XYZ ERR1*
ABCD
/*
```

プログラム名接頭部へのアスタリスクの追加はオプションです。これにより IDIDATST の動作が変わることはありません。

## ISPF パック・コンパイラー・リスト

ISPF には、PACK データ・セットを対象とする機能があります。このフォーマットでは ISPF は、反復文字を、文字の反復回数を示すシーケンスで置き換えます。この機能により、直接アクセス・ストレージ・デバイス (DASD) をさらに効率的に使用できます。コンパイラー・リストを開くと、Fault Analyzer によって、リストが PACKed フォーマットであるかどうかを検出され、使用前にアンパックされます。

ISPF 環境でコンパイラー・リストをパックするには、EDIT セッション内からリストに対して PACK 基本コマンドを実行し、リストを保管します。

また、パック・コンパイラー・リストの作成は、追加ステップをコンパイル・ジョブに挿入することによって自動化できます。ジョブの例を以下に示します。この例では、REXX exec から ISPF LMCOPY コマンドを使用し、PACK オプションを指定してデータ・セット間でリストをコピーします。

```
//PACKLIST EXEC PGM=IKJEFT01
//SYSPROC DD DSN=rex-exec-library,DISP=SHR
//ISPPPLIB DD DISP=SHR,DSN=ISP.SISPPENU
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSLIB
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU
//ISPLOG DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1210)
//ISPLIST DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1210)
//SYSTSPRT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=1210)
//SYSPRINT DD SYSOUT=*
//LMIN DD DISP=SHR,DSN=compile-step-listing-data-set
//LMOUT DD DISP=SHR,DSN=packed-listing-data-set
//ISPPROF DD SPACE=(TRK,(5,1,4)),UNIT=SYSALLDA,
// DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=6160)
//SYSTSIN DD *
ISPSTART CMD(LMCOPY member-name)
/*
```

*compile-step-listing-data-set* と *packed-listing-data-set* の両方が PDS または PDSE データ・セットの名前です。メンバー指定はなく、*member-name* はコンパイラー・リストのメンバー名です (通常は、コンパイルされるプログラム名に一致します。コンパイラー・リストまたはサイド・ファイルの命名 ページ 382を参照してください)。

LMCOPY REXX exec は、メンバー LMCOPY として *rexx-exec-library* データ・セットに存在する必要があります。以下に例を示します。

```
/* LMCOPY REXX */
Address ISPEXEC
Arg member
'LMINIT DATAID(INDD) DDNAME(LMIN) ENQ(SHRW)'
'LMINIT DATAID(OUTDD) DDNAME(LMOUT) ENQ(SHRW)'

'LMCOPY FROMID('INDD') TODATAID('OUTDD')
  FROMMEM('member') TOMEM('member') REPLACE PACK'

'LMFREE DATAID('INDD')'
'LMFREE DATAID('OUTDD')'
```

ISPF PACK はコンパイラー・リストで使用される DASD スペースを節約しますが、コンパイラー・リストを LANGX サイド・ファイルに変換してリストを破棄すると、より多くのスペースを節約できます。(詳細については、「*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*」の「*IPVLANGX* コンパイラー・リストからサイド・ファイルへの変換ユーティリティ

」の章の「*IPVLANGX* を使用したサイド・ファイルの作成」のセクションを参照してください。)

オリジナルのコンパイラー・リスト情報の大半を LANGX サイド・ファイルから再印刷できます。「*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*」の IPVLANGP ユーティリティの説明を参照してください。

## Java ソース情報の提供 Fault Analyzer

Java™ アプリケーション・ソース情報を Fault Analyzer へ提供するためのオプションが 2 つあります。

1. IDIJAVA オプションまたは DD 連結
2. Java™ クラスパスのソース jar の指定

Fault Analyzer は IDIJAVA オプションまたは IDIJAVA DD 連結で指定した各パスで一致するソース・ファイルを検索しません。

IDIJAVA パスが指定されなかった、またはソース・ファイルが見つからなかった場合、Fault Analyzer は明示的に指定した各 jar ファイルおよびディレクトリーでアプリケーション・イベントと一致するソース・ファイルを検索します。

### Java ソース情報の最適化

Java ソースが要求されることがない、または使用可能にはならない場合、NOSOURCE オプションを指定して、Fault Analyzer に一致する Java™ ソース・ファイルを検索しないように伝えます。ソース jar ファイルまたは Java™ ソース・ファイルがアプリケーション・クラスパスにない場合、`-dropcp-` オプションを指定して、Fault Analyzer にソース・ファイルの検索時にアプリケーション・クラスパスを無視するように伝えます。その代わりに、IDIJAVA オプションを使用して指定したパスだけが、ソース情報の検索時に考慮されます。

## 例: IDIJAVA DD および IDIJAVA IDIOPTS の使用

このトピックでは、さまざまなシナリオで、IDIJAVA DD および IDIJAVA IDIOPTS を使用して Java™ アプリケーション・ソース情報を Fault Analyzer に提供する方法を示します。

### BPXBATCH/BPXBATSL

IDIJAVA DD と BPXBATCH を使用するには、代替エントリー・ポイント BPXBATSL を使用する必要があります。これが DD データ・セットおよびパス・アプリケーションへの spawn されたプログラム・アクセスを許可するからです。

```
//RUNJAVA EXEC PGM=BPXBATSL,REGION=500M
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDPARM DD *
PGM /bin/sh java AnApp arg1 arg2
/*
//STDENV DD *
CLASSPATH=<the application classpath>
JAVA_HOME=<path to java>
LIBPATH=<paths to application DLLs>
PATH=${PATH}:${JAVA_HOME}/bin
IBM_JAVA_OPTIONS=-Xmx400m
JAVA_TOOL_OPTIONS=-agentlib-...
/*
//IDIJAVA DD PATHOPTS=ORDONLY,
// PATH='/u/hunter2/src'
// DD PATHOPTS=ORDONLY,
// PATH='/opt/apis/coffee-src.jar'
```

### IBM® JZOS バッチ・ランチャー

IDIJAVA DD と JZOS バッチ・ランチャーを使用することは BPXBATSL の場合と同じです。JZOS バッチ・ランチャー・ジョブ・ステップで IDIJAVA DD を指定してください。

```
cd <JAVA_HOME>/mvstools
cp samples/jcl/JVMJCL80 "'SYS1.SAMPLIB(JVMJCL80)'"
cp samples/jcl/JVMJCL80 "'SYS1.PROCLIB(JVMJCL80)'"
cp -X JVMLDM80 "'SYS1.SIEALNKE(JVMLDM80)'"
```

バッチ・ランチャーは Java™ の非 SMP/E インストールの Java™ に含まれています。詳細は、IBM® SDK, Java™ Technology Edition のドキュメントの『[Java™ Batch Launcher and Toolkit for z/OS \(JZOS\)](#)』を参照してください。

### IDIJAVA DD での長いパス値の指定

JCL で 72 文字よりも長いパスを指定するには、JCL 記号を使用してパスを表すためのオプションがあります。例:

```
/* Define each part of the path as a symbol:
// SET QQ=''' * This sets symbol &QQ. to a single quote
// SET SRCPART1='/u/hunter2/org/very/large/financial/'
// SET SRCPART2='organisation/JavaSourceFiles/prod'
// SET SRCPART3=&QQ.&SRCPART1.&SRCPART2&QQ.
/*
...
//IDIJAVA PATHOPTS=ORDONLY,
// PATH=&QQ.&SRCPART3/Version1/.QQ.
```

## IDIJAVA オプションの使用

IDIJAVA オプションは parmlib メンバー IDICNFxx にシステム全体で指定することができます。または IDIOPTS DD を使用して特定のジョブ・ステップで指定することができます。

```
//IDIOPTS DD *
  Datasets(
    IDIJAVA(
      -dropcp-
        /u/hunter2/srcFiles
        /u/hunter2/deps/dep2-src.jar
    )
  )
/*
```

各パス (ここでは “xxxxx...” で示します) はスペースで区切る必要があります。長いパスは、パスの各パートの間に「+」を指定することで、行をまたいで続けることができます。パスの長さは最高 1023 文字まで指定できます。

```
//GO.IDIOPTS DD DATA,DLM='##',SYMBOLS=JCLONLY
  datasets(idijava(
    /01xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /02xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /03xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /04xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /05xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /06xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /07xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /08xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /09xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx +
    /10xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx))
```

## 最適なデバッグのための Java のコンパイル

コンパイルされた .class ファイルに含まれるデバッグ情報の大きさとタイプは、ビルド・ツールおよび明示的に指定したデバッグ・オプションによって異なります。

### 以下のデバッグ情報オプション IBM® Java™

これらは IBM® javac コンパイラーに関連するデバッグ・オプションです。

#### -g:source

コンパイルされたクラス・ファイルに Java™ ソース・ファイル名が含まれます。

#### -g:lines

クラス・ファイルに LineNumberTable が含まれます。このテーブルは、各ソース・ファイルの行番号と生成された java バイトコード指示をマップします。マシン・コード・ステートメントは Java™ 仮想マシンにより使用されます。

**-g:vars**

クラス・ファイルに各非ネイティブおよび非抽象メソッドに対する LocalVariableTable が含まれます。このテーブルには、各ローカル変数の名前とタイプ、および変数が有効な範囲が含まれていません。



**重要:** Java™ プログラムのデバッグが必要とする likelihood がある場合、次のオプションを使用して、すべてのデバッグ情報をすべてのコンパイルされた .class ファイルに含めます。

```
javac -g:source,lines,vars <class files>
```

このオプションの結果として、例外のために表示されるスタック・トレースに各スタック・エントリーのロケーション情報が含まれます。例:

```
Exception in thread "main" java.lang.NullPointerException
    at com.example.Book.getTitle(Book.java:16)
    at com.example.Author.getBookTitles(Author.java:25)
    at com.example.Bootstrap.main(Bootstrap.java:14)
```

デバッグ・オプションが指定されていない、または `-g:none` オプションが使用されている場合、問題判別用に使用できる情報は少なくなります。

```
Exception in thread "main" java.lang.NullPointerException
    at com.example.Book.getTitle(Unknown location)
    at com.example.Author.getBookTitles(Unknown location)
    at com.example.Bootstrap.main(Unknown location)
```

これはスタック・トレースの値を減少させるだけでなく、プログラムのデバッグ時にローカル変数の名前が表示されるのを防ぎます。その代わりに、すべての変数が “<local variable>” または同様の役に立たない名前として (開発環境により異なる) 表示されるだけです。

## 共通 Java ビルド・ツールを使用したデバッグ情報の生成

大規模な Java™ アプリケーションの構築には多くのアプローチがあります。Apache Ant、Maven および Gradle などの一般的なビルド・ツールでは、それぞれ異なる方法で、含めるデバッグ情報に対するデバッグ・フラグを指定する必要があります。

### Apache Ant javac タスク

build.xml ファイルで、次の例のように、javac タスクのインスタンスがデバッグ・オプションを指定するようにします。

```
<javac debug="true" debuglevel="source,lines,vars" <other attributes> >
<javac/>
```

<other attributes> はその他のサイト固有またはプロジェクト固有の javac タスクのコンパイル属性を参照します。javac Ant タスクについての詳細は、Ant のドキュメントを参照してください。

## Maven maven-compiler-plugin

Maven は詳細な指示なしで、Java™ クラス・ファイルをコンパイルできます。しかし、このデバッグ情報を含むようにするには、maven-compiler-plugin 用の特定のエントリーを追加します。デバッグ・オプションがプラグインの設定セクションに表示されることを確認します。

```
<build>
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>${maven.compiler.version}</version>
    <configuration>
      <compilerArgs>
        <arg>-g:source,lines,vars</arg>
      </compilerArgs>
    </configuration>
  </plugin>
</plugins>
</build>
```

maven-compiler-plugin の使用方法の詳細は、Maven のドキュメントを参照してください。

## Gradle

Gradle ビルド・ファイルで、次のプロパティが指定されていることを確認します。

```
compileJava.options.debugOptions.debugLevel = "source,lines,vars"
compileTestJava.options.debugOptions.debugLevel = "source,lines,vars"
```

## 第 21 章. CICS 環境のカスタマイズ

CICS® でトランザクションの異常終了の場合に Fault Analyzer を呼び出せる出口点は 3 つあります。

### XPCABND

プログラム IDIXCX53 を使用するグローバル・ユーザー出口。この出口は、CICS® トランザクション障害分析用に Fault Analyzer を呼び出すために提供されているメイン出口です。

### XDUREQ

プログラム IDIXCX53 を使用するグローバル・ユーザー出口。この出口は、EXEC CICS® DUMP コマンドで生成された CICS® ダンプを行う Fault Analyzer を呼び出すために使用できます。

この出口点で Fault Analyzer により実行される分析は、XPCABND 出口点の場合と同じです。

### LE Exit

プログラム IDIXCCEE を使用する LE 異常終了出口。この出口は、CEECXTAN 出口が設定されているとき、Language Environment® ・ベースのアプリケーション・プログラムを使用する場合にのみ有効です。

CICS® AKCS 異常終了は、次の両方の状況が当てはまる場合は、この出口を使用して分析できます。

- 障害のあるプログラムで LE が使用可能。
- AKCS の CICS® ダンプ・テーブルに入力が存在し、トランザクション・ダンプの実行が指定されている。

3 つのうち最初の 2 つの出口は CICS® グローバル・ユーザー出口点であり、Fault Analyzer はこれらのポイントで CICS® 呼び出しを使用することで使用可能または使用不可になります。デフォルトでは、これらの出口点は、CICS® 領域では有効になっていません。これらは、CICS® PLT に新規エントリーを追加するか ([必要なプログラムの始動 PLT への追加 ページ 402](#)を参照)、CICS® の初期化後に、指定された CFA トランザクションを使用する ([CICS トランザクション異常終了分析の制御 ページ 405](#)を参照) ことにより使用可能になります。

ただし、LE 異常終了出口を使用する場合は LE に修正を加える ([Language Environment for CICS を呼び出すための構成 Fault Analyzer ページ 401](#)を参照) 必要があるため、その影響はシステム全体に及びます。Fault Analyzer には、この出口の CICS® 領域レベルでの使用を制御するメカニズムがありますが、このメカニズムを有効にするためには、まず LE 出口をシステム全体で使用可能にする必要があります。システム全体で使用可能にすると、CICS® 領域内の初期設定が有効になります。

上記のいずれの出口を使用する場合も、それらを正常に機能させるには IDI.SIDIAUTH を CICS® JCL の DFHRPL 連結に追加する必要があります。

Fault Analyzer を CICS® で使用するには、以下のステップで実行する必要があります。

#### 1. CICS® による Fault Analyzer 呼び出しのための Language Environment® の構成

詳しくは、[Language Environment for CICS を呼び出すための構成 Fault Analyzer ページ 401](#)を参照してください。



## 2. 必要なプログラムの CICS® システムへの定義

詳しくは、[必要なプログラムの CICS システムへの定義 ページ 401](#)を参照してください。

## 3. 必要なプログラムの始動 PLT への追加

詳しくは、[必要なプログラムの始動 PLT への追加 ページ 402](#)を参照してください。

## 4. 必要なプログラムのシャットダウン PLT への追加

詳しくは、[必要なプログラムのシャットダウン PLT への追加 ページ 402](#)を参照してください。

## 5. Fault Analyzer のトランザクションの定義

詳しくは、[CICS トランザクション異常終了分析の動的制御の有効化 ページ 403](#)を参照してください。

## Language Environment for CICS を呼び出すための構成 Fault Analyzer

Fault Analyzer は、CICS XPCABND グローバル出口への Fault Analyzer を起動する別のメソッドとして、CICS® 用の Language Environment® 異常終了出口を提供します。詳しくは、『[CICS LE 異常終了 CEECXTAN CSECT 出口 IDIXCCEE ページ 303](#)』を参照してください。この出口を使用可能にするには、この出口を Language Environment® 用の CICS® の CEECXTAN CSECT に追加する必要があります。これを行うには、CEE.SCEESAMP データ・セット内に CEEWCEXT ソフトコピー・サンプル・メンバーのコピーを作成します。このサンプル・メンバーで推奨される変更を加え、次の行を変更します。

```
<<< REPLACE THESE 2 LINES WITH A COPY OF CEECXTAN
AND OVERRIDE AS DESIRED >>>
```

これらの行を、以下の行で置き換えてください。

```
CEEXAHD      ,User exit header
CEEXART      TERMxit=IDIXCCEE
CEEXAST      ,Terminate the list
```

CICS® 用の Language Environment® 異常終了出口の実装に関する一般情報については、『[Language Environment® のカスタマイズ](#)』を参照してください。



**注:** データ・セット CEE.SCEESAMP でサンプル・ジョブ CEEWDEXT とともにインストールされる この出口の非 CICS バージョン (CEEEXTAN) は、この出口と非常によく似ています。両方ともインストールする場合は、正しい名前を使用していること、および同じ出口をインストールしていないことを重ねて確認してください。

## 必要なプログラムの CICS システムへの定義

CICS® 自動インストール・プログラムがアクティブでない限り、CICS の始動時に使用するグループ・リストに含まれるグループで次のアセンブラー・プログラムと BMS マップを定義する必要があります:

- IDIPLT
- IDIPLTD
- IDIPLTS
- IDIXCX53 (CICS の現在のすべてのバージョンおよびリリースに適用可能)

- IDIXMAP (BMS マップ)
- IDIXFA (オプションの CICS トランザクション)

データ・セット IDI.SIDISAM1 のメンバー IDISCICS 内のサンプル・ジョブでは、FA という名前のグループ内のプログラム、BMS マップ、オプションの制御トランザクションを定義します。個別のインストール用にサンプル・ジョブをカスタマイズするには:

- サンプル・ジョブのデータ・セット名の接頭部 xxx を、インストール用の適切な名前と置き換えます。
- *list-name* を適切な CICS 始動 SIT GRPLIST 名に置き換えます。
- FA のグループ名を変更できます。
- オプションの制御トランザクションは、サンプル・ジョブでは CFA という名前です (この資料の他の箇所では「CFA」と呼ばれます)。ただし、この名前は変更できます。このトランザクションの詳細については、[CICS トランザクション異常終了分析の動的制御の有効化 ページ 403](#) を参照してください。

Fault Analyzer を CICS で起動できるようにするには、IDI.SIDIAUTH を DFHRPL 連結に追加する必要があります。

Fault Analyzer で CICS トレース情報を表示するには、CICS トレースがアクティブである必要があります。

Fault Analyzer を起動するには、EXEC CICS ABEND ステートメントで ABCODE キーワードを使用する必要があります。例:

```
EXEC CICS ABEND ABCODE('abcd') END-EXEC
```

EXEC CICS ABEND ステートメントで NODUMP キーワードを使用すると、Fault Analyzer は IDIXCEE 出口から起動された場合にのみ分析を実行します。

## 必要なプログラムの始動 PLT への追加

CICS® 始動時に Fault Analyzer をインストールするには、プログラム名 IDIPLT を始動 PLT に追加します。IDIPLT は、CICS® 起動時に XPCABND グローバル・ユーザー出口として IDIXCX53 を有効にします。



**注:** CICS® オープン TCB ユーザーの場合は、Fault Analyzer IDIPLT プログラム名を始動 PLT に必ず追加しなければなりません。

通常の CICS® サービスが使用可能になる前に IDIPLT が呼び出されると、別の PLT プログラムで異常終了が発生するので、Fault Analyzer は PLT リストの末尾に配置する必要があります。また、プログラム名 Fault Analyzer を始動 PLT に追加することにより、XDUREQ グローバル・ユーザー出口にて IDIPLTD を使用可能にできます。

IDIPLTS を始動 PLT に含めることができます (SDUMP スクリーニングのインストールを指定した場合)。

## 必要なプログラムのシャットダウン PLT への追加

すべての Fault Analyzer アクティビティを適切に終了できるように、次のエントリーを CICS® シャットダウン PLT に追加する必要があります。

```
DFHPLT TYPE=ENTRY,PROGRAM=IDIPLT
```

このエントリーは、DFHDELIM エントリー (シャットダウン・フェーズ 1) の前に追加する必要があります。

Fault Analyzer は、IDIPLT を正常に終了させるだけでなく、現在使用可能なすべての CICS® 出口点にある Fault Analyzer も使用不可にします。この使用不可化により、これ以降、CICS® シャットダウン中に異常終了分析が行われなくなります。シャットダウン中に分析が必要な場合には、IDIPLT をシャットダウン PLT に追加しないでください。ただし、その場合は、シャットダウン中にシステム 33E 異常終了が発生する可能性があることに注意してください。

Fault Analyzer SDUMP 選別機能がインストールされている場合、オプションで対応する項目を CICS SHUTDOWN PLT に追加して、この機能を CICS® シャットダウン中は無効にすることができます。

```
DFHPLT TYPE=ENTRY,PROGRAM=IDIPLTS
```

## CICS トランザクション異常終了分析の動的制御の有効化

Fault Analyzer がインストールされている場合、CICS® で Fault Analyzer の動作を動的に制御する制御トランザクションをオプションでインストールできます。

制御トランザクションをインストールするには、必要なプログラムの定義に使用したものと同一グループでプログラム IDIXFA の CICS® トランザクションを定義します。(詳しくは、[必要なプログラムの CICS システムへの定義 ページ 401](#)を参照してください。)これには CEDA トランザクションを使用できます。パラメーターにはデフォルト値を使用します。

データ・セット IDI.SIDISAM1 のメンバー IDISCICS 内のサンプル・バッチ・ジョブによって、CFA という名前の制御トランザクション(この資料の他の箇所では「CFA」と呼びます)を定義します。ただし、この名前は変更できます。

制御トランザクションには十分に高い優先順位を割り当て、予期せぬ問題が発生した場合に、迅速に Fault Analyzer を使用できないようにする必要があります。

## CFA を使用した現在分析中のタスクでの FORCEPURGE の実行

IDIXFA プログラム(トランザクション CFA)によって提供される機能の 1 つに Fault Analyzer によって現在分析されているタスクに FORCEPURGE を実行する機能があります。

IDIXFA を実行する CICS® 領域がトランザクション分離を ACTIVE にして実行されている場合、この機能を使用可能にするには EXECKEY(CICS) を指定して IDIXFA プログラムを定義する必要があります。

## SVC ダンプ・スクリーニング

CFA トランザクションに、CICS® 領域の SVC ダンプのキャプチャー性能を向上できる機能が提供されました。この機能は、CFA トランザクションを使用してインストールおよびアンインストールことができ、SDUMP SVC (SVC 51) スクリーニングにおいてダンプが取得されている間、SDATA の XESDATA および GRSQ ダンプ・オプションを常にオフにします。



**注:** SDUMP SVC (SVC 51) のスクリーニングする別のツールまたはプロダクトをお持ちの場合は、Fault Analyzer dump screening を (CFA トランザクション経由、または IDIPLTS CICS® PLT プログラムのいずれを問わず) インストールしないようお勧めします。Fault Analyzer ダンプ・スクリーニングをインストールしようとしているときに別



のスクリーニング・プログラムが見つかった場合、その Fault Analyzer コードはインストールされず、メッセージが表示されます。

## 定義ジョブの例

図 203 : Fault Analyzer CICS プログラムとトランザクションの定義ジョブの例 ページ 404 は、上記のすべてのプログラムおよびトランザクションを CICS® に定義する場合に使用できる、サンプル・バッチ・ジョブを示しています。xxx 接頭部を含むデータ・セット名を、ご使用のシステムに適した名前に置き換え、*list-name* を適切な CICS® 始動 SIT GRPLIST 名に置き換えてください。この例ではグループ名 FA が選択されていますが、必要に応じて変更できます。

図 203. Fault Analyzer CICS プログラムとトランザクションの定義ジョブの例

```
//IDICICS JOB ...
//IDICICS EXEC PGM=DFHCSDUP,REGION=1024K,
//          PARM='CSD(READWRITE),PAGESIZE(60),NOCOMPAT'
//STEPLIB DD DISP=SHR,DSN=xxx.SDFHLOAD
//DFHCSD DD DISP=SHR,DSN=xxx.DFHCSD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEF PROGRAM(IDIPLT) GROUP(FA) EXECKEY(CICS)
    CEDF(NO) DATALOCATION(ANY)
    CONCURRENCY(QUASIRENT)
  DEF PROGRAM(IDIPLTD) GROUP(FA) EXECKEY(CICS)
    CEDF(NO) DATALOCATION(ANY)
    CONCURRENCY(QUASIRENT)
  DEF PROGRAM(IDIPLTS) GROUP(FA) EXECKEY(CICS)
    CEDF(NO) DATALOCATION(ANY)
    CONCURRENCY(QUASIRENT)
  DEF PROGRAM(IDIXCX53) GROUP(FA)
    CEDF(NO) DATALOCATION(ANY)
    CONCURRENCY(QUASIRENT)
  DEF PROGRAM(IDIXCCEE) GROUP(FA)
    CEDF(NO) DATALOCATION(ANY)
    CONCURRENCY(QUASIRENT)
  DEF PROGRAM(IDIXFA) GROUP(FA) EXECKEY(CICS)
    CEDF(NO) DATALOCATION(ANY)
    CONCURRENCY(QUASIRENT)
  DEF PROGRAM(IDIVPCLE) GROUP(FA)
    CONCURRENCY(QUASIRENT)
  DEF TRANSACTION(CFA) GROUP(FA)
    PROGRAM(IDIXFA) TASKDATALOC(ANY)
    SHUTDOWN(ENABLED)
  DEF MAPSET(IDIXMAP) GROUP(FA)
    ADD G(FA) L(list-name)
/*
```

上図に示すサンプル・ジョブは、メンバー IDISCICS としてデータ・セット IDI.SIDISAM1 に提供されます。

Fault Analyzer を CICS® で起動するには、IDI.SIDIAUTH を DFHRPL 連結に追加する必要があります。

CICS® CICS® トレース情報を表示するには、Fault Analyzer のトレースがアクティブである必要があります。

Fault Analyzer を起動するには、EXEC CICS® ABEND ステートメントに ABCODE キーワードを使用する必要があります。  
例:

```
EXEC CICS ABEND ABCODE('abcd') END-EXEC
```

CICS® ABEND ステートメントに NODUMP キーワードが指定されると、Fault Analyzer は IDIXCCEE 出口から起動された場合のみ分析を実行します。

## CICS トランザクション異常終了分析の制御

CFA トランザクションがインストールされると (上記により、別の名前でインストールすることを選択している可能性があります)、以下の Fault Analyzer 呼び出し出口のインストールまたはアンインストールに使用できます。

- XPCABND CICS® グローバル・ユーザー出口
- XDUREQ CICS® グローバル・ユーザー出口
- LE 異常終了出口
- XEIN グローバル・ユーザー出口

また、CFA トランザクションは、Fault Analyzer の SDUMP スクリーニング機能のインストールおよびアンインストールにも使用できます。

XPCABND または XDUREQ 出口のいずれかのインストールに先立ち、CFA トランザクションはこれらの出口が両方とも「アンインストール済み」の状態である場合、プログラム IDIXCX53 に CICS® NEWCOPY コマンドを発行します。したがって、IDIXCX53 の新規コピーを (例えば保守の適用後に) ロードするには、CFA トランザクションを使用して XPCABND および XDUREQ 出口をアンインストールし、その後 IDIXCX53 出口を再インストールします。

CFA トランザクションとの対話には 2 つの方法があり、CICS® 端末からの対話、または MVS™ コンソールからの対話のいずれかを実行できます。これらの方法を、それぞれ以下のセクションで説明します。

## CICS® 端末からの CFA の使用

CFA トランザクションを CICS® 端末から使用するには、単に CFA を入力します。オプションとして、[MVS コンソールからの CFA の使用 ページ 408](#)に示すようにコマンド・パラメーターを渡すこともできます。初期表示画面には、次のように表示されます。

図 204. CFA トランザクション画面の例

```

Fault Analyzer Control Transaction

Options: I=Install U=Uninstall
Current Status/Error Message
- XPCABND      Installed
- XDUREQ      Installed
- LE Exit     Installed
- SDUMP Screening Installed
- XEIIIN      Uninstalled

Current   HWM   Setting   MWS
Active   0000  0000    0002
Waiting  0000  0000    0020  0123

IDITRACE OFF DeferredReport ON

PF1=Help PF3=Exit PF4=Opts PF5=Clear FND Area PF9=IVP PF11=TRACE ENTER=Update

```

画面にはまず、CICS® Fault Analyzer 出口点の現在の状況と、アクティブおよび待機中の Fault Analyzer タスクの詳細が表示されます。I または U (I: インストール (Install)、U: アンインストール (Uninstall)) を特定の出口点に入力すると、それに応じて状況が変更されます。(上記の例のように) アクティブな分析タスクが存在する場合、アクティブなタスクの詳細の横にある入力フィールドに F を入力して CICS® TASK FORCEPURGE を発行できます。この機能は、CICS® トランザクション分離が INACTIVE であるか、ACTIVE である場合は IDIXFA プログラムが CICS® の EXECKEY を指定して定義されている場合に限り、使用可能です。

PF9 を使用して、IVP テストのリストを表示することができます。詳しくは、[CICS での Fault Analyzer のカスタマイズの検証 ページ 432](#)を参照してください。

PF11 機能については、[IDITRACE CICS において ページ 407](#)を参照してください。

特定の CICS® 出口についてのヘルプ情報を表示するには、メインパネルの出口選択フィールドにカーソルを合わせて PF1 を押します。

PF4 を押すと、Fault Analyzer 終了オプションの画面が表示されます。この表示画面は、当該 CICS® 領域に対して有効な現行デフォルト・オプションに関する情報を提供します。[図 205: CFA 終了オプション画面のサンプル ページ 407](#)に、この画面の例が表示されています。

図 205. CFA 終了オプション画面のサンプル

Fault Analyzer Exit Options		
Description	Option	Setting
Debug trace . . . . .	ZZDEBUG . . . . .	OFF
Suppress Msg: IDI0034I . . . . .	QUIET (IDI0034I) . . .	OFF
IDI0066I . . . . .	QUIET (IDI0066I) . . .	OFF
IDI0118W . . . . .	QUIET (IDI0118W) . . .	OFF
Transaction Dump Table Check . . . . .	CICSDUMPTABLEEXCLUDE	OFF
Check MAX and Current values. . . . .	CheckMaxCurr	OFF
Retain CICS dumps . . . . .	RETAINCICSDUMP . . .	AUTO
Suppress dumps from EXEC CICS DUMP.	IncludeExecCicsDump	NO
Fast duplicate minutes . . . . .	NODUP . . . . .	00000
Include EXEC CICS DUMP. . . . .	IECD . . . . .	ON
Use signed-on user for analysis . . . . .	CICSTRANANALYSISUSER	NO

PF3=Exit

これらのオプションは、新しい障害項目が作成されるたびに更新され、IDIOPTS DD ステートメントまたは IDICNFxx parmlib メンバーを経由して有効になる値を反映します。

## NoDup(CICSFAST(...)) 記録域の消去

CICS® のもとでは、Fault Analyzer は、FND (fast-no-dup) 域と呼ばれる、処理された最近の異常終了を維持管理しています。この区域は、NoDup(CICSFAST(...)) オプションとともに使用します。CFA トランザクション画面から、PF5 を押してヒストリー画面にアクセスします。その後 PF5 を使用してこの区域を消去できますが、これは、以降の FAST 重複判別で、以前の異常終了が使用されないことを意味します。FND 域を本当に消去したいかどうかを確認するために、PF5 を再び押すようにプロンプトが出されます。

## IDITRACE CICS において

CFA トランザクションから PF11 を使用して、Fault Analyzer デバッグ・トレースの IDITRACE を有効または無効にすることができます。トレースは、JES スプールに動的に割り振られた IDITRACE DDname 書き込まれます。例えば、SDSF 下の CICS® 領域に対して「?」を入力すると、以下の画面が表示され、「S」で選択可能な IDITRACE 出力が ① に表示されることがあります。

図 206. CICS 領域の SDSF Job Data Set (SDSF ジョブ・データ・セット) 画面の例

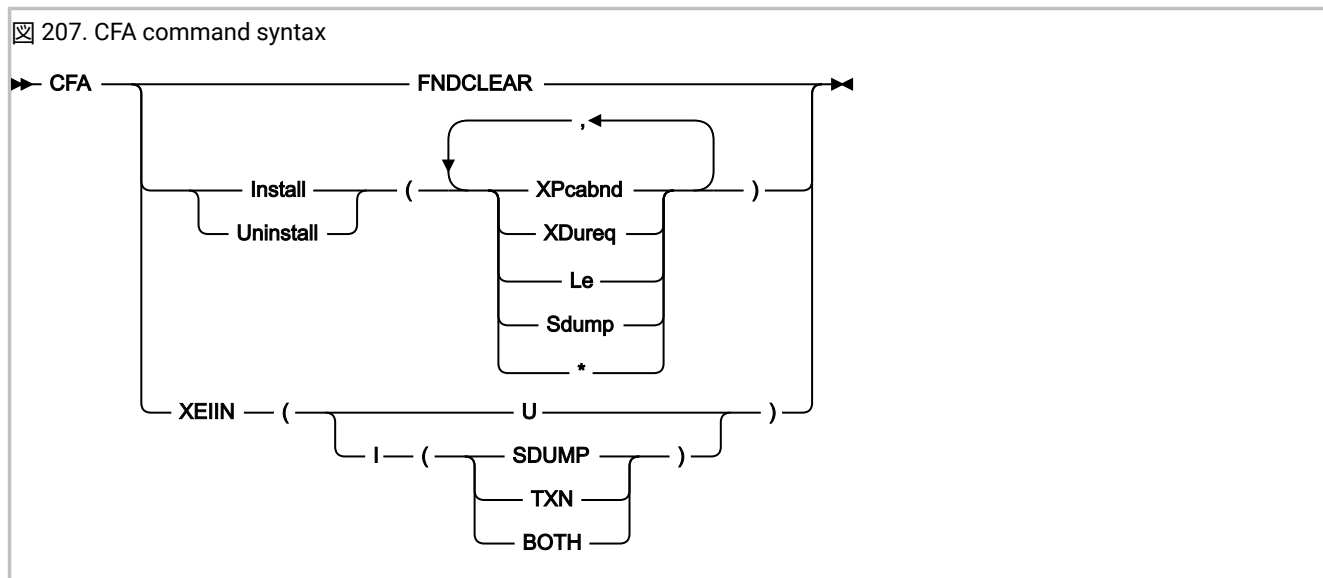
```

Display Filter View Print Options Help
-----
SDSF JOB DATA SET DISPLAY - JOB AS650F1 (JOB31639)      DISPLAY RESET
COMMAND INPUT ==>                                       SCROLL ==> CSR
PREFIX=* DEST=(ALL) OWNER=SOMEONE SYSNAME=
NP  DDNAME  StepName ProcStep DSID Owner  C Dest          Rec-Cnt Page
JESMSG LG JES2          2 SOMEONE X           4
JESJCL  JES2          3 SOMEONE X          259
JESYSMSG JES2          4 SOMEONE X        2,173
SYSPRINT AS650F1 AS650F1 103 SOMEONE X          126
MSGUSR  AS650F1 AS650F1 105 SOMEONE X        2,328
PLIMSG  AS650F1 AS650F1 106 SOMEONE X           0
DFHCXRF AS650F1 AS650F1 107 SOMEONE X          641
COUT    AS650F1 AS650F1 108 SOMEONE X           0
CEEMSG  AS650F1 AS650F1 109 SOMEONE X           0
CEEOUT  AS650F1 AS650F1 110 SOMEONE X           0
PRINTER AS650F1 AS650F1 116 SOMEONE X           0
① IDITRACE AS650F1 AS650F1 117 SOMEONE X          172
CRPO    AS650F1 AS650F1 119 SOMEONE X           0
    
```

### MVS コンソールからの CFA の使用

CFA トランザクションを MVS™ コンソールから使用するには、CFA コマンド・パラメーターとともに MODIFY (F) コマンドを発行します。CFA コマンド構文を以下に示します。

図 207. CFA command syntax



各コマンドの説明。

#### FND CLEAR

FND 領域がクリアされます (CICS 端末からの CFA の使用 ページ 405 の説明を参照)。



**INSTALL | UNINSTALL**

特定の Fault Analyzer CICS® 出口 (XEIIN グローバル・ユーザー出口以外) がインストール/アンインストールされます。以下の各オプションを括弧で囲み、任意の順序で指定できます。

**XPCABND**

CICS® XPCABND グローバル・ユーザー出口がインストール/アンインストールされます。

**XDUREQ**

CICS® XDUREQ グローバル・ユーザー出口がインストール/アンインストールされます。

**LE**

CICS® IDIXCCEE LE 異常終了出口がインストール/アンインストールされます。

**SDUMP**

CICS® SVC ダンプ・スクリーニングがインストール/アンインストールされます。

\*

上記すべて (XPCABND、XDUREQ、LE、および SDUMP) がインストール/アンインストールされます。

**XEIIN**

Fault Analyzer CICS® XEIIN グローバル・ユーザー出口がインストール/アンインストールされます。

**U**

XEIIN 出口がアンインストールされます。

**I**

指定の操作の XEIIN 出口がインストールされます。

**SDUMP**

システム・ダンプ用の CICS® XEIIN  
グローバル・ユーザー出口がインストールされます。

**TXN**

トランザクション・ダンプ用の CICS® XEIIN  
グローバル・ユーザー出口がインストールされます。

**BOTH**

システム・ダンプ用とトランザクション・ダンプ用の CICS® XEIIN  
グローバル・ユーザー出口がインストールされます。

**例:**

- CICS® 領域 CICS01 からすべての出口をアンインストールするには、次のコマンドを入力します。

```
F CICS01,CFA U(*)
```

- CICS® 領域 TESTCICS が使用する LE 出口をインストールするには、次のコマンドを入力します。

```
F TESTCICS,CFA I(L)
```

- CICS® 領域 PRODCICS が使用する XPCABND および XDUREQ 出口をインストールするには、次のコマンドを入力します。

```
F PRODCICS,CFA I(XP,XD)
```

- CICS® 領域 MYCICS にある FND 領域をクリアするには、次のコマンドを入力します。

```
/F MYCICS,CFA FNDCLEAR
```

- CICS® 領域 CICSP1 から XEIIIN をアンインストールするには、次のコマンドを入力します。

```
/F CICSP1,CFA XEIIIN(U)
```

- CICS® 領域 CICSА でシステム・ダンプ用に XEIIIN 出口をインストールするには、次のコマンドを入力します。

```
/F CICSА,CFA XEIIIN(I(SDUMP))
```

## CICS Web サービス・トランザクション

動的制御トランザクション CFA に加えて (「[CICS トランザクション異常終了分析の動的制御の有効化 ページ 403](#)」を参照)、Fault Analyzer では CICS Web サービス・トランザクションを使用した場合も同じ機能が提供されます。

このオプションの機能をインストールするには、以下のステップが必要です。

1. CICS 文書テンプレートを保持するディレクトリーを z/OS® UNIX System Services に作成します。  
CICS Web サービス・トランザクションは、z/OS® UNIX System Services ディレクトリーから読み取られ、そのディレクトリー/パスが CICS 文書テンプレート・リソース定義で参照される CICS 文書テンプレートを使用します。ローカル命名標準 (例えば `/faultanalyzer/doctemplate`) を使用して、z/OS® UNIX System Services ディレクトリーを作成します。
2. CICS 文書テンプレートをディレクトリーにコピーします。  
データ・セット IDI.SIDISAM1 にあるサンプル・ジョブ IDISCHFZ が提供され、文書テンプレートをステップ 1 で作成した z/OS® UNIX System Services ディレクトリーにコピーします。サンプル・ジョブの指示を確認します。主な要件は、z/OS® UNIX System Services パスを、上記で作成されたパス (例えば `/faultanalyzer/doctemplate`) に更新することです。
3. 必要な CICS リソース定義を作成します。  
必要な CICS リソース定義を作成するために、データ・セット IDI.SIDISAM1 にあるサンプル・ジョブ IDISCIC2 が提供されています。サンプル・ジョブの指示を確認します。ステップ 1 で作成されたディレクトリー名 (例えば `/faultanalyzer/doctemplate`) は、2 つの DOCTEMPLATE 定義で参照する必要があります。  
  
サンプル・ジョブでは CICS TCPIPSERVICE が定義されています。ただし、既存の TCPIPSERVICE も使用可能です。定義が完了したら、GROUP をインストールします。
4. CICS Web トランザクションをテストします。  
Fault Analyzer CICS Web サービスをテストするには、ブラウザーに以下のトランザクションを入力します。

```
http://host:port/IDIXFAW
```

各表記の意味は次のとおりです。

- `<host>` は、CICS が実行されているホスト名/IP アドレスです。
- `<port>` は、TCPIP SERVICE 定義で指定されているポート番号です。

正常に実行されると、以下のような画面が表示されます。

図 208. CICS Web トランザクション

## Fault Analyzer for z/OS V15

Region	Version	Release
C74E1FA1	060100	0740

Refresh

Exit	Status	Actions
XPCABND	Installed	Uninstall
XDUREQ	Installed	Uninstall
LE Exit	Installed	Uninstall
SDUMP screening	Installed	Uninstall
XEIN Exit	Uninstalled	Install

	Current	HWM	Value	Max Wait Seconds
Active	0000	0001	<input type="text" value="5"/>	
Waiting	0000	0000	<input type="text" value="17"/>	<input type="text" value="0"/>

Update Settings

IDITRACE  OFF

Options in effect

Analysis History

## トランザクション異常終了分析が DUMP(NO) に抑止されないことを確認

異常終了するトランザクションへのアクティブ・トランザクション定義に DUMP(NO) オプションが指定されている場合、CICS® は XPCABND グローバル・ユーザー出口を呼び出さないため、Fault Analyzer が呼び出されません。トランザクションの DUMP 設定を確認するには、以下のいずれか 1 つを実行します。

- CEDA トランザクションを使用して、該当のトランザクション定義を表示させ、DUMP(YES|NO) 設定を確認します。この方法を使用する場合は注意が必要です。1 つのトランザクションに複数の定義が存在する可能性があるため、CICS® によりインストールされる定義の順番が重要となります。
- ダンプ内のアクティブ・トランザクション定義を確認します。
- CICS® が提供されたトランザクション CECI を使用して、アクティブ・トランザクションの DUMP 設定を確認します。この確認は、次のコマンドを発行することによって行うことができます。

```
CECI INQUIRE TRANSACTION(nnnn)
```

*nnnn* は該当のトランザクション ID です。

このコマンドを発行することで表示される DUMPING 値は、以下を示します。

- 値 00186 は DUMP(YES) を示す。
- 値 00187 は DUMP(NO) を示す。

## CICS NoDup(CICSFAST) オーバーライド・アセンブラー出口 (IDINDFUE)

特定の状況では、Fault Analyzer によって CICS® NoDup(CICSFAST) 重複と見なされる異常終了が検出された場合でも (NoDup オプションについての情報は、[NoDup ページ 611](#)を参照してください)、この検出をオーバーライドして通常の異常終了の分析を行うことが望ましい場合があります。このような状況では、例えば特定の異常終了またはトランザクション、あるいはそれらの組み合わせに対して、完全分析が必要になる場合があります。このような状態に対応するために、NoDup(CICSFAST) アセンブラー出口、IDINDFUE があります。Fault Analyzer は CICS® NoDup(CICSFAST) メッセージ [ID10066I ページ 710](#) を発行する前にこの出口プログラムのロードを試みます。ロードに成功すると、プログラム IDINDFUE が呼び出されます。

## 呼び出し

IDINDFUE はアセンブラー言語で作成する必要があります。この出口は OS リンケージ規則を使用して呼び出されます。

IDINDFUE は、IDIXCCEE または IDIXCX53、すなわち CICS® XPCABND グローバル・ユーザー出口 (GLUE) から呼び出される可能性があります。この呼び出しは、XPCABND GLUE プログラムをコーディングするための CICS® 規則に準拠する必要があります。詳細については、「CICS® カスタマイズ・ガイド」を参照してください。IDINDFUE をアセンブルしているときは、CICS® コマンド・レベル変換プログラムを使用しないでください。

出口プログラムには以下を行う必要があります。

- IDINDFUE というロード・モジュールにリンク・エディットします。
- 自動ロードがアクティブの場合は CICS® 領域の DFHRPL 連結に配置します。
- 自動ロードがアクティブでない場合は PPT に組み込みます。

## エントリーの指定

IDINDFUE に対するエントリーでのレジスターの内容:

### レジスター

#### 内容

1

入力パラメーター・リストのアドレス (下を参照)

13

72 バイトのレジスター保管域のアドレス

14

戻りアドレス

## 入力パラメーター・リスト

R1 はアドレスのリストから構成されるパラメーター・リストのアドレスを含みます (OS リンケージ)。パラメーター・リスト内の各アドレスは、記述されているとおりに 1 つのパラメーターを指します。

表 12. IDINDFUE 入力パラメーター

パラメーター	バイト数	説明
パラメーター 1	256	ユーザー作業域
パラメーター 2	4	異常終了コード
パラメーター 3	4	トランザクション ID
パラメーター 4	8	異常終了プログラム名

## 戻り指定

IDINDFUE から戻るときのレジスターの内容

### レジスター

#### 内容

0 から 14.

未変更。

15

戻りコード:

0

異常終了を重複として扱い続けることを示します。

1

異常終了分析を進行する必要がある、つまり NoDup(CICSFAST) 検出を無効にすることを示します。



**注:** IDINDFUE 出口が、分析が実行される必要があることを示す戻りコード 1 を戻した場合、NoDup(NORMAL) 重複検出は実行されません。

## 例

以下に示すサンプル・ユーザー出口のソフトコピーも、データ・セット IDINDFUE のメンバー IDI.SIDISAM1 として提供されます。トランザクション ID が FTN1 で、異常終了コードが FABN である場合、このサンプル・アセンブラー出口は R15 を 1 に設定します。

```

IDINDFUE CSECT
          PRINT ON,NOGEN
          STM  R14,R12,12(R13)
          LR   R12,R15
          USING IDINDFUE,R12
          L    R4,0(,R1)           Work Area
          L    R2,4(,R1)           Abend code
          L    R3,8(,R1)           Transaction ID

          SR   R15,R15             Assume no continuance.
          CLC  0(4,R2),=C'FABN'    Check Abend code
          BNE  RETURN
          CLC  0(4,R3),=C'FTN1'    Check Transaction ID
          BNE  RETURN
          LA   R15,1               Indicate analysis to be performed

*
RETURN   DS   0H
          L    R14,12(,R13)
          LM   R0,R12,20(R13)
          BR   R14                 Return to Fault Analyzer
          END  IDINDFUE

```

## CICS トレース考慮事項

Fault Analyzer が、異常終了するタスクのトレース項目を収集するときは、トレース項目のコピー先としてストレージ域が使用されます。このストレージ域のサイズ、およびそれに応じてコピーされるトレース項目の数は、次のように決まります:

- CICS® TS 5.1 以上の場合は、CICS® システム初期設定パラメーター TRTRANSZ が使用されます。
- それより前の CICS® リリースの場合は、64K を超える値が CICS® システム初期設定パラメーター TRTRANSZ で指定されない限り、最小値 64K が使用されます。64K を超える値が使用される場合は、その値 (上限 256K) が使用されます。



## LE による CICS トレースの循環を回避

CICS® トランザクションが異常終了し、Language Environment®が異常終了しているエンクレーブ内でアクティブな場合、Language Environment®は、診断情報をデフォルトで一時データ・キュー CESE に書き込みます。この状態が発生するのは、IBM 提供のLanguage Environment®デフォルト・ランタイム・オプション TERMTHDACT(TRACE) が有効になっている場合です。Fault Analyzer に制御が渡され、異常終了の処理が行われる以前に、これらの診断が記録されるため、CICS® トレース・テーブルが先頭から上書きされる可能性が高く、その結果アプリケーションのトレース・データが失われる可能性があります。ご使用の MVS™ のレベルに合わせ、TERMTHDACT オプションを次のいずれかに設定することをお勧めします。

### TERMTHDACT(TRACE,CICSDDS,...)

この値の場合、Language Environment®は CICS® トランザクション・ダンプ・データ・セットに診断を書き込みます。

### TERMTHDACT(QUIET)

この値の場合、多くのLanguage Environment®診断が抑止されます。

## IDIOPTS DDname を使用した CICS オプションの指定

コンパイラ・リストまたはサイド・ファイルのデータ・セットが変更されたときに CICS® のリサイクルを行わないようにするには、IDIOPTS DDname がポイントするユーザー・オプション・ファイルの DataSets オプションでこれらを指定します。詳細については、[DataSets ページ 576](#)および [User options file IDIOPTS ページ 571](#)を参照してください。

また、データ・セットをポイントし以下を含む IDIOPTS DDname を使用することが、

```
DataSets (IDIHIST (history-file-dsn))
```

CICS® JCL では、次のステートメントを使用するよりも推奨されます。

```
//IDIHIST DD DISP=SHR,DSN=history-file-dsn
```

以下を使用したとします。

```
//IDIHIST DD DISP=SHR,DSN=history-file-dsn
```

この場合、JCL 割り振りでは、CICS® の稼働中に、フルタイムでヒストリー・ファイル・データ・セット名に ENQ が保持されます。ただし、

```
DataSets (IDIHIST (history-file-dsn))
```

Fault Analyzer の要求に応じて、割り振りが動的に取得されリリースされます。結果として、ヒストリー・ファイルへの書き込みが行われた CICS® システムが引き続き稼働中であっても、そのヒストリー・ファイルは保守したり名前変更したりできます。この

```
DataSets (IDIHIST (history-file-dsn))
```

オプションにより、CICS® が実行中に、新規のヒストリー・ファイルをポイントするように変更することができ、この変更は次の障害項目が作成される時点で有効になります。

CICS® の実行中に ISPF EDIT を使用する更新を許可するために、使用される IDIOPTS データ・セットは PDS または PDSE である必要があります。

## 言語環境プログラム異常終了の考慮事項

Language Environment® 下でトランザクション異常終了から復旧しようとしているときに CICS® で異常終了が発生した場合、Fault Analyzer は起動されません。これは、これらのタイプの異常終了に対する CICS® の通常動作では、XPCABND 出口が呼び出されないためです。ただし、IDIXCCEE に付属の CICS® LE 異常終了 CEEEXTAN CSECT 出口 Fault Analyzer を使用可能にすると、これらのタイプの異常終了が分析されます。この出口の有効化の詳細については、[Language Environment for CICS を呼び出すための構成 Fault Analyzer ページ 401](#)を参照してください。

## CICS ユーザー・キー・オープン TCB (L9 TCB) での実行時の異常終了の取り込み

Fault Analyzer が、オープン (L9) TCB での実行時に CICS® トランザクションの異常終了分析を行うためには、Fault Analyzer XPCABND ユーザー出口を使用可能にしておく必要があります。この出口を有効にする方法については、[CICS トランザクション異常終了分析の制御 ページ 405](#)を参照してください。

## MVS ポスト・ダンプ出口のインストール IDIXTSEL

Fault Analyzer ポスト・ダンプ出口 IDIXTSEL は、MVS™ IEAVTSEL CSECT インストール済み環境出口リストにインストールされます。SVC ダンプ用にのみ呼び出されるこの出口は、USERMOD、IDIWTSEL によりインストールされます。これは、CICS® システム異常終了ダンプ、リカバリー障害記録 SDUMP、SLIP ダンプを登録して、Java™ 障害を容易に取り込めるようにするために使用します。

この USERMOD をインストールするには、サンプル・ジョブ IDIWTSEL を編集して実行依頼します。このサンプル・ジョブでは、IEAVTSEL インストール済み環境出口リストに IDIXTSEL が組み込まれています。このリストに他の出口が定義されている場合は、IDIXTSEL 出口を最後に追加します。

IEAVTSEL への出口の追加に関する詳細は、「[MVS™ インストール・システム出口](#)」を参照してください。

この変更内容をアクティブにするには、再 IPL するか、DUMPSRV アドレス・スペースを取り消して、新しい出口で再始動するようにします。

この出口を介してダンプを登録するには、次のことも必要です。

- Fault Analyzer IDIS サブシステムを開始します。この詳細については、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照してください。
- デフォルトの Fault Analyzer ヒストリー・ファイルが PDSE であることを確認するか、または分析制御ダンプ登録ユーザー出口を使用して登録障害項目を PDSE ヒストリー・ファイルに割り当てられていることを確認します。ダンプ登録障害項目は、PDSE ヒストリー・ファイルにのみ作成できます。

デフォルトのヒストリー・ファイル名の指定の詳細については、[parmlib メンバー IDICNFxx ページ 372](#)を参照してください。

分析制御ダンプ登録ユーザー出口の詳細については、[分析制御ユーザー出口 \(MVS SVC ダンプ登録\) ページ 479](#)を参照してください。

ダンプ登録のセットアップをテストする方法についての情報は、[ダンプ登録の検証 \(IEAVTSEL\) ページ 439](#)でご覧いただけます。

## CICS トランザクション異常終了分析パフォーマンスの最大化

CICS® トランザクション異常終了分析パフォーマンスを最大化するには、以下のことを考慮する必要があります。

- Fault Analyzer IDIS サブシステムを使用して、ヒストリー・ファイル \$\$INDEX メンバーへのアクセスを管理する。詳しくは、[ヒストリー・ファイル \\$\\$INDEX データのキャッシュ ページ 320](#)を参照してください。
- DeferredReport オプションを使用する。詳しくは、[DeferredReport ページ 584](#)を参照してください。

このオプションを CICS® だけに適用できるようにしたい場合、[IDIOPTS DDname を使用した CICS オプションの指定 ページ 417](#)で説明されているように、IDIOPTS DDname を使用して指定することができます。

## XEIIIN グローバル・ユーザー出口の実装

CICS® タスク終了によってストレージの最終解放が行われるまでオーバーレイが検出されない場合、XEIIIN グローバル・ユーザー出口が役立つことがあります。明示的な FREEMAIN の間、またはタスク終了の間、CICS® は Storage Accounting Area (SAA) オーバーレイを検出できます。タスク終了で検出された場合、タスクは CICS® によって異常終了されないため、Fault Analyzer は起動しません。CICS® はダンプ・コード SM0102 でシステム・ダンプを取得しようと試みますが、CICS® システム・ダンプが抑止されている場合、それ以降の分析は可能ではありません。

SAA オーバーレイに役立つことを目的に、Fault Analyzer には、必要に応じて XEIIIN グローバル・ユーザー出口にインストールできるプログラムが用意されています。インストールされている場合、このプログラムは EXEC CICS® RETURN ステートメントごとに、すべての CICS® Storage Accounting Area で現在のタスクを検証します。オーバーレイが見つかったと、以下の 2 つのアクションのいずれかが実行されます。

1. CICS® トランザクション・ダンプをダンプ・コード IDIS で取るための呼び出しが行われます。

Fault Analyzer が XDUREQ グローバル・ユーザー出口にインストールされている場合、このトランザクション・ダンプは通常の方法で分析され、ヒストリー・ファイル障害項目が作成されます。

2. CICS® 領域の SDUMP が取られます。

この SDUMP を取る要求は IDIS サブシステムによって出されるため、CICS® システム・ダンプ抑止設定による影響はありません。SDUMP が取り込まれると、「File」メニューのオプション 5 で対話式で分析できるようになります。

Fault Analyzer XEIIIN グローバル・ユーザー出口プログラムの使用可能化、および実行されるアクションの設定は、Fault Analyzer 提供の CFA トランザクションを使用して行われます。他の Fault Analyzer CICS® 出口とは異なり、CICS® PLT 処理の間に XEIIIN 出口を有効にするオプションはありません。

呼び出されたときの初期 CFA 画面は、次のようになっています。

```

Fault Analyzer Control Transaction
Options: I=Install U=Uninstall
Current Status/Error Message
- XPCABND      Installed
- XDUREQ      Installed
- LE Exit     Installed
- SDUMP Screening Installed
- XEIIIN      Uninstalled

Current   HWM   Setting   MWS
Active   0000  0001    0001
Waiting  0000  0000    0017  0000

IDITRACE OFF DeferredReport ON

PF1=Help PF3=Exit PF4=Opts PF5=Clear FND Area PF9=IVP PF11=TRACE ENTER=Update

```

この画面で「I」をXEIIINの横に入力すると、その出口をインストールできます。この後、次のような画面が表示されま  
す。

```

Fault Analyzer Control Transaction
Options: I=Install U=Uninstall
Current Status/Error Message
- XPCABND      Installed
- XDUREQ      Installed
- LE Exit     Installed
- SDUMP Screening Installed
- XEIIIN      Installed   N TXN dump N SDUMP Set at least one option to Y

Current   HWM   Setting   MWS
Active   0000  0001    0001
Waiting  0000  0000    0017  0000

IDITRACE OFF DeferredReport ON

PF1=Help PF3=Exit PF4=Opts PF5=Clear FND Area PF9=IVP PF11=TRACE ENTER=Update

```

この画面で、TXN dump または SDUMP の横に「Y」を入力して、SAA オーバーレイが検出されたときに実行するアクションを  
設定します。

I (Install) オプションは、TXN フィールドまたは SDUMP フィールドが変更されるたびに使用する必要があります。



**注:** Fault Analyzer 出口をインストール/有効にしたときのオーバーヘッドはありませんが、XEIIN 出口だけは EXEC CICS® コマンドごとに CICS® によって呼び出されるため、わずかなオーバーヘッドを生じます。そのため、SAA オーバーレイを調べる必要がある場合のみ、XEIIN 出口を使用するようお勧めします。

## 3270 画面バッファ・キャプチャーの無効化

3270 画面データを取り込むために、Fault Analyzer はコマンド RECEIVE BUFFER を端末に対して発行します。ただし、以下のエラーが発生する可能性があります。

- 端末が実際の 3270 装置ではない
- 端末がコマンド を処理するための状態になっていない F2 READBUF

このような場合、CICS® は RECEIVE BUFFER をアベンド ATNI で異常終了させます。これにより、グローバル・ユーザー出口 XPCABND および XDUREC の再帰呼び出しが行われます。

これが ATNI 異常終了の問題である場合は、次の JCL ステートメントを CICS® 始動プロシージャーに追加します。

```
//IDINOCRB DD DUMMY
```

このステートメントは Fault Analyzer に、領域内のすべての 3270 装置について、3270 画面キャプチャー (RECEIVE BUFFER) をスキップするよう指示します。これにより、セクション **Last CICS 3270 Screen Buffer** は、Fault Analyzer 分析レポートには含まれません。

## 第 22 章. DB2 環境のカスタマイズ

このセクションでは、DB2® 環境の Fault Analyzer に特有な情報を記載します。

### DB2 のバインディング

DB2® を使用するアプリケーションで発生する異常終了に対して Fault Analyzer を実行するには、DB2® コール・レベル・インターフェース (CLI) がインストールされていること、およびプラン DSNACLI をバインドするために必要なセットアップが実行されていることを確認する必要があります。

DSNACLI プランは、DB2® SDSNSAMP データ・セットのメンバー DSNTIJCL で、サンプル・ジョブを使用して作成できます。DSNTIJCL ジョブが戻りコード 0 で終了することが重要です。ジョブ・コメントに示されるように、これを達成するには SQLERROR(CONTINUE) parm の追加が必要になる場合があります。詳細については、「*DB2® for OS/390® Call Level Interface Guide and Reference*」を参照してください。

Fault Analyzer IDIS サブシステムが、必要な DB2® システム・カタログ表に対する SELECT 権限を持っていることを確認してください。詳しくは、[IDIS DB2 のサブシステム要件 ページ 324](#)を参照してください。

### DB2 および言語環境プログラム

CEEEXTAN LE 異常終了出口 (IDIXCEE) がインストールされていない場合は、Fault Analyzer を DB2® 異常終了時に起動させるために、TERMTHDACT(UATRACE)、TERMTHDACT(UADUMP)、TERMTHDACT(UAONLY)、または TERMTHDACT(UAIMM) のいずれかを LE に渡す必要があります。

LE オプションを渡す方法を示す COBOL/DB2 の例は、次のとおりです。

```
//MYJOB      JOB
//STEP1      EXEC PGM=IKJEFT01
//DBRMLIB    DD DSN=TEST.DB2.DBRMLIB.DATA,DISP=SHR
//SYSTSPRT   DD SYSOUT=*
//SYSPRINT   DD SYSOUT=*
//SYSOUT     DD SYSOUT=*
//IDIAREPRT  DD SYSOUT=*
//IDIHIST    DD DISP=SHR,DSN=TEST.DB2.HIST
//IDILCOB    DD DISP=SHR,DSN=TEST.LISTING.DB2.COBOL
//SYSIN      DD *
//SYSTSIN    DD *
DSN SYSTEM(DSN1)
BIND PLAN(DSNTST1) QUALIFIER(DSN8510) MEMBER(DACBD001) -
ACT(REP) ISOLATION(CS)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA51) -
LIB('DSN510.RUNLIB.LOAD')
RUN PROGRAM(DACBD001) PLAN(DSNTST1) -
LIB('CTEST.DB2.LOAD') -
PARMS('/TERMTHDACT(UADUMP) ')
END
//*
```

### Fault Analyzer DB2 パフォーマンスの向上

以下の情報は DB2® V10 より前の DB2® バージョンに関する情報です。DB2® V10 以降の場合、この情報は無視できます。

DB2® SYSIBM.SYSDBRM カタログ表に定義された索引がありません。Fault Analyzer は、DB2® 障害の分析を実行するたびに、SYSIBM.SYSDBRM カタログ表にアクセスします。SYSIBM.SYSDBRM へのアクセス時にパフォーマンスが低下することを回避するには、SYSIBM.SYSDBRM カタログ表にユーザー定義の索引を作成します。非固有索引によって、次の列が(順序どおりに)指定されます。

```
PLNAME
NAME
```

次のサンプル DDL を使用して、索引を定義できます。

```
CREATE TYPE 2 INDEX nnnnnn.DBRMX           ❶
      ON SYSIBM.SYSDBRM
      (
        PLNAME ASC,
        NAME ASC
      )
      USING STOGROUP mmmmmm                ❶
          PRIQTY pp                          ❷
          SECQTY ss                          ❷
CLOSE NO;
```



❶

索引名 (*nnnnnn*) とストレージ・グループ名 (*mmmmmm*) を変更して、要件に適合するようにします。例えば、索引名 SYSIBM とストレージ・グループ STOGROUP を使用します。

❷

1 次 (*pp*) と 2 次 (*ss*) の数量を変更して、要件に適合するようにします。例えば、1 次と 2 次の両方の数量に 250 を使用します。

サンプル・ジョブ IDISDB2X が IDI.SIDISAM1 内に配布されており、次の操作に役立ちます。

## 第 23 章. IMS 環境のカスタマイズ

このセクションでは、IMS™ 環境の Fault Analyzer に特有な情報を記載します。

### IMS および言語環境プログラム

CEEEXTAN LE 異常終了出口 (IDIXCEE) がインストールされていない場合は、Fault Analyzer を IMS™ 異常終了時に起動させるために、TERMTHDACT(UATRACE)、TERMTHDACT(UADUMP)、TERMTHDACT(UAONLY)、または TERMTHDACT(UAIMM) のいずれかを LE に渡す必要があります。

実行されるロード・モジュールに CEEUOPT CSECT をリンクすることにより、LE オプションを渡す方法を示す COBOL/IMS の例は、次のとおりです。

```
//IMSLE1 JOB ...
//*
//*      STEP 1: ASSEMBLE CEEUOPT CSECT
//*
//HLASM  EXEC PGM=ASMA90,PARM='LINECOUNT(0)'
//SYSPRINT DD SYSOUT=*
//SYSUT1  DD DISP=(,PASS),UNIT=SYSALLDA,SPACE=(TRK,(1,5))
//SYSLIN  DD DISP=(,PASS),UNIT=SYSALLDA,SPACE=(TRK,(1,5,1)),DSN=&TEMP(CEEUOPT)
//SYSLIB  DD DSN=CEE.SCEEMAC,DISP=SHR
//        DD DSN=SYS1.MACLIB,DISP=SHR
//SYSIN   DD *
        TITLE 'CEEUOPT'
CEEUOPT  CSECT
CEEUOPT  AMODE ANY
CEEUOPT  RMODE ANY
        CEEUOPT TERMTHDACT=(UADUMP)
        END
//*
//*      STEP 2: COMPILE COBOL PROGRAM
//*
//COBCOMP EXEC IMSCOBOL
//COB.SYSIN  DD DSN=DA.IMSSAMP.COBOL(BATCHJ2),DISP=SHR
//COB.SYSPRINT DD DSN=DA.LISTING.COBOL(BATCHJ2),DISP=SHR
//LKED.FRED  DD DSN=*.HLASM.SYSLIN,DISP=OLD
//LKED.SYSIN  DD *
        Include FRED(CEEUOPT)
        NAME  BATCHJ2(R)
//*
//*
//*      STEP 3: RUN THE PROGRAM
//*
//PROGRUN  EXEC PROC=DLIBATCH,MBR=BATCHJ2,PSB=PSB1,COND=(4,LT),
//          DBRC=Y,MON=Y,FMTQ=D,TIME=5
//          UNIT=3390,
//          DCB=BLKSIZE=6144
//SYSPRINT DD SYSOUT=*
//DFSIVD1  DD DISP=SHR,DSN=IMS.DFSIVD1
//DFSIVD1I DD DISP=SHR,DSN=IMS.DFSIVD1I
//DFSCTL   DD DISP=SHR,
//          DSN=IMS.PROCLIB(DFSSBPRM)
//IDIREFPT DD SYSOUT=*
//SYSTSIN  DD *
//*
```



## 第 24 章. Fault Analyzer 日本語フィーチャーのカスタマイズ

このセクションは、Fault Analyzer の日本語フィーチャーのユーザーのみに適用されます。

### ISPF データ・セットの割り振り

[ISPF 環境の変更 ページ 328](#)に記載された Fault Analyzer の基本機能に必要なものに加え、以下の ISPF DDname およびデータ・セットを割り振る必要があります。

#### **DDname**

##### **データ・セット名**

#### **IDIIPJPN**

IDI.SIDIPJPN

#### **IDIIMJPN**

IDI.SIDIMJPN

#### **IDIISJPN**

IDI.SIDISJPN

#### **IDIITJPN**

IDI.SIDITJPN

通常、ISPF アプリケーションのデータ・セットは、TSO ログオン・プロシージャ、または ISPF を呼び出す前に実行されたプログラムまたは EXEC のいずれかに割り振られるか、もしくは ISPF LIBDEF サービスを使用するアプリケーションの呼び出し前に動的に (例えば EXEC で) 割り振られます。

Fault Analyzer が Language(JPN) オプションを使用して呼び出される場合、Fault Analyzer は ISPF LIBDEF サービスを使用して IDIIPJPN、IDIIMJPN、IDIISJPN、および IDIITJPN DDname に割り振られたデータ・セットを ISPLIB、SPMLIB、SPSLIB、および ISPTLIB DDname に割り振られたデータ・セットの前に論理的に配置します。LIBDEF サービスのスタッキング・フィーチャーは、Fault Analyzer を呼び出す前に LIBDEF を使用して定義されたデータ・セットが、出口にリストアされていることを確認するために使用されます。

ISPLIB、SPMLIB、SPSLIB、または ISPTLIB のいずれかの LIBDEF が Fault Analyzer の起動時に既にアクティブだった場合は、既存の LIBDEF データ・セットが、IDIIPJPN、IDIIMJPN、IDIISJPN、および IDIITJPN データ・セットの後に続いて、新規 LIBDEF に組み込まれます。DATASET オプションを使用して LIBDEF に指定できるデータ・セットの最大数は 15 に制限されるため、LIBDEF を使用して既に 14 個を超えるデータ・セットが指定されている場合、超過分は使用可能になりません。(この演算では、通常の場合のように、IDIIPJPN、IDIIMJPN、IDIISJPN、および IDIITJPN の DDname に対して 1 つのデータ・セットのみが指定されることを前提としています。)このため、Fault Analyzer の起動時に LIBDEF を使用して指定される Fault Analyzer 基本機能の ISPF データ・セットも、Fault Analyzer によって設定される LIBDEF に組み込まれるデータ・セットの中に含める必要があります。そうしないと、未翻訳メンバーが検出されないことが原因で、ISPF が失敗する恐れがあります。

## 各国語サポートの設定

Language オプション ([Language ページ 607](#)ページを参照) は、適切な言語依存データ・セットの選択に使用する各国語 ID を指定します。

すべてのユーザーに対して日本語をデフォルト言語に設定するには、IDICNFxx parmlib メンバーに次のオプションを指定します。

```
LANGUAGE(JPN)
```

## 第 25 章. Fault Analyzer のカスタマイズの検証

以下のトピックでは、異なるプログラム言語と実行環境別に、Fault Analyzer のインストールおよびカスタマイズが正常に実行されたことを検証する手順を示しています。

### XDUMP の使用の検証

どの IVP を使用して Fault Analyzer インストールをテストするかに関係なく、オプション MaxMinidumpPages(0) を一時的に指定することで XDUMP をテストできます。

このオプションは、IDIOPTS DDname を介して指定することも、IDICNFxx parmlib メンバーに指定することもできます。

このオプションを使用すると、すべてのミニダンプ・ページが XDUMP データ・セットに強制的に「スピル」されます。その結果、メッセージ [IDI0194W ページ 739](#) も発行されます。これは正しく、この状態では予期されます。

### アセンブラーによる Fault Analyzer の使用の検証

#### AMODE 31

Fault Analyzer を AMODE 31 アセンブラーで検証するには、データ・セット IDIVPASM のサンプル・ジョブ IDI.SIDISAM1 を編集し、実行依頼します。詳細については、サンプル・ジョブ内の説明を参照してください。

ジョブはプログラムをアセンブルし、実行します。このプログラムはシステム異常終了コード 0C7 で異常終了します。

このプログラムは非 LE プログラムであるため、Fault Analyzer は MVS™ オプション変更/ダンプ抑止出口 IDIXDCAP によって呼び出されます。

Fault Analyzer に書き込まれた IDIREPRT レポートの概要セクションには、以下が含まれます。

Fault Analyzer の日本語フィーチャーでは、Language(JPN) オプションが有効である場合、これは日本語になっているはずですが。

```
A system abend 0C7 reason code X'0' occurred in module G0 CSECT IDISASM1 at
offset X'44'.

A program interruption code 0007 (Data Exception) is associated with this abend
and indicates that:

    A decimal digit or sign was invalid.

The cause of the failure was module G0 CSECT IDISASM1. The Assembler source
code that immediately preceded the failure was:

List
Stmt #
-----
000042          CVB    R5,WORKD
```

The Assembler source code for data fields involved in the failure:

```
List
Stmt #
-----
000107  WORKD          DS D
```

Data field values at time of abend:

```
WORKD = X'000000000222278' (Address 00007E50 = R12 + X'110')
```

この IVP の実行により作成される完全なサンプル・レポートが、IDI.SIDISAM1 データ・セット内のメンバー IDISRP03 として用意されています。

この IVP の LE 準拠バージョンは、データ・セット IDI.SIDISAM1 内のメンバー IDIVPBLE として提供されます。詳しくは、[IDIXCEE Language Environment 出口の有効化の検証 ページ 431](#) を参照してください。

## AMODE 64

Fault Analyzer を AMODE 64 アセンブラーで検証するには、データ・セット IDI.SIDISAM1 のサンプル・ジョブ IDIVPAS6 を編集し、実行依頼します。詳細については、サンプル・ジョブ内の説明を参照してください。

## COBOL による Fault Analyzer の使用の検証

このセクションは、ご使用のサイトに COBOL がインストール済みである場合のみ適用できます。

Fault Analyzer を COBOL で検証するには、データ・セット IDIVPCOB のサンプル・ジョブ IDI.SIDISAM1 を編集し、実行依頼します。詳細については、サンプル・ジョブ内の説明を参照してください。

ジョブは COBOL プログラムをコンパイルし、実行します。このプログラムは戻りコード 3000 で異常終了します。

TER(UADUMP) LE オプションの結果として、Fault Analyzer は MVS™ 変更オプション/抑止ダンプ出口 IDIXDCAP から呼び出されます。

Fault Analyzer に書き込まれた IDIREPRT レポートの概要セクションには、以下が含まれます。

Fault Analyzer の日本語フィーチャーでは、Language(JPN) オプションが有効である場合、これは日本語になっているはずですが。



**注:** ご使用のコンパイラーのバージョン、リリースまたは保守レベルが異なる場合、プログラムのオフセット情報が下記の例とは異なることがあります。

```
A system abend 0C7 occurred in module IDISCBL1 program IDISCBL1 at offset
X'3D4'.
```

```
A program interruption code 0007 (Data Exception) is associated with this abend
and indicates that:
```

```
A decimal digit or sign was invalid.
```

The cause of the failure was program IDISCBL1 in module IDISCBL1. The COBOL source code that immediately preceded the failure was:

```
Source
Line #
-----
000029      CLEAR SECTION.
000030      START001.
000031      DIVIDE NUMBERX BY ERROR-COUNT GIVING BAD-RESULT.
```

The COBOL source code for data fields involved in the failure:

```
Source
Line #
-----
000011      01  NUMBERX PIC 999999 COMP-3.
000013      05  ERROR-COUNT PIC 999999 COMP-3.
000016      01  BAD-RESULT PIC 99 COMP-3.
```

Data field values at time of abend:

```
BAD-RESULT  = X'0000' *** Invalid numeric data ***
ERROR-COUNT = X'C1C2C3C4' *** Invalid numeric data ***
NUMBERX     = 986888
```

この IVP の実行により作成される完全なサンプル・レポートが、IDI.SIDIDOC1 データ・セット内のメンバー IDISRP01 として用意されています。

## PL/I による Fault Analyzer の使用の検証

このセクションは、ご使用のサイトに PL/I がインストール済みである場合のみ適用できます。

### AMODE 31

ご使用の PL/I のバージョンに応じて、2 つの異なる AMODE 31 IVP ジョブが使用可能です。

- Enterprise PL/I を使用する場合は、データ・セット IDI.SIDISAM1 のサンプル・ジョブ IDIVPPLE を編集し、実行依頼します。
- Enterprise PL/I 以外を使用する場合は、データ・セット IDI.SIDISAM1 のサンプル・ジョブ IDIVPPLI を編集し、実行依頼します。

詳細については、サンプル・ジョブ内の説明を参照してください。

各ジョブは PL/I プログラムをコンパイルし、実行します。このプログラムは戻りコード 3000 で異常終了し、ジョブ・ステップを終了させます。

TER(UADUMP) LE オプションの結果として、Fault Analyzer は MVS™ 変更オプション/抑止ダンプ出口 IDIXDCAP から呼び出されます。

Fault Analyzer に書き込まれた IDIREPRT レポートの概要セクションには、以下が含まれます。

Fault Analyzer の日本語フィーチャーでは、Language(JPN) オプションが有効である場合、これは日本語になっているはずですが。



**注:** ご使用のコンパイラーのバージョン、リリースまたは保守レベルが異なる場合、プログラムのオフセット情報が下記の例とは異なることがあります。

```
A system abend 0C9 occurred in module IDISPLI1 program IDISPLI1 at offset
X'286'.

A program-interruption code 0009 (Fixed-Point-Divide Exception) is associated
with this abend and indicates that:

    The divisor was zero in a signed binary division.

The cause of the failure was program IDISPLI1 in module IDISPLI1.  The PL/I
source code that immediately preceded the failure was:

List
Stmt #
-----
000011      Array1(1) = Array1(2) / Divisor;

Data field values at time of abend:

List
Stmt #
-----
000009  ARRAY1(1)  FIXED BIN(31,0)  AUTO = X'00000001'
000009  ARRAY1(2)  FIXED BIN(31,0)  AUTO = X'00000003'
000009  DIVISOR    FIXED BIN(31,0)  AUTO = X'00000000'
```

この IVP の実行により作成されるすべてのレポート例は、IDI.SIDIDOC11 データ・セット内のメンバー IDISRP02 に用意されています。

## AMODE 64

Fault Analyzer を AMODE 64 Enterprise PL/I で検証するには、データ・セット IDI.SIDISAM1 のサンプル・ジョブ IDIVPPL6 を編集し、実行依頼します。詳細については、サンプル・ジョブ内の説明を参照してください。

## C による Fault Analyzer の使用の検証

このセクションは、ご使用のサイトに C がインストール済みである場合のみ適用できます。

Fault Analyzer を C で検証するには、データ・セット IDIVPC のサンプル・ジョブ IDI.SIDISAM1 を編集し、実行依頼します。詳細については、サンプル・ジョブ内の説明を参照してください。

ジョブは、IDI.SIDISAM1(IDISRC1) で C プログラムをコンパイルして実行します。このプログラムは異常終了し、ジョブ・ステップを戻りコード 3000 で終了します。

書き込まれた Fault Analyzer レポートの概要セクションには、以下が含まれます。

Fault Analyzer の日本語フィーチャーでは、Language(JPN) オプションが有効である場合、これは日本語になっているはずですが。

```
A system abend 0C9 occurred in module G0 program IDISRC1 at offset X'47A'.

A program-interruption code 0009 (Fixed-Point-Divide Exception) is associated
with this abend and indicates that:

    The divisor was zero in a signed binary division.

The cause of the failure was program IDISRC1 in module G0.  The C source code
that immediately preceded the failure was:

Source
File # Line #
-----
000000 000097      d = (a + b) / (c - 42);      /* abend */

where source file #:

000000 = system-id://'IDI.SIDISAM1(IDISRC1)'
```

この IVP の実行により作成される完全なサンプル・レポートが、IDI.SIDIDOC1 データ・セット内のメンバー IDISRP06 として用意されています。

## IDIXCEE Language Environment 出口の有効化の検証

IDIXCEE Language Environment® 出口を使用する場合、正常に動作していることを次の手順で検証できます。

1. 以下のいずれかを実行します。
  - a. COBOL または PL/I を使用している場合、IDIVPCOB または IDIVPPLI ジョブのいずれかをデータ・セット IDI.SIDISAM1 で編集し、GO ステップ LE オプション `TER(UADUMP)` を `TER(TRACE)` に変更します。
  - b. アセンブラーを使用している場合、データ・セット IDI.SIDISAM1 で IDIVPBLE ジョブを編集します。この IVP は IDIVPASM IVP に相当しますが、LE 準拠アセンブラーを使用します。
2. ジョブを実行依頼します。
3. IDIREPRT 出力の下部に移動し、以下で開始される段落が含まれていることを確認します。

```
Fault Analyzer was invoked via the LE CEEEXTAN exit (IDIXCEE)...
```

上記が含まれている場合は、IDIXCEE 出口は正常に動作しています。

## CICS での Fault Analyzer のカスタマイズの検証

このセクションは、CICS® がインストール済みであり、[CICS 環境のカスタマイズ ページ 400](#) の説明に従い CICS® のカスタマイズが完了している場合のみ適用できます。

CFA トランザクションおよび必要な出口をインストールした後、CICS® 端末から CFA トランザクションを呼び出し、PF9 を押して次の画面を表示させます。

図 209. CFA IVP Testing 画面の例

```

                                Fault Analyzer IVP Testing

Options: S=Select

                                IVP Description
- 0C1 in program IDIXFA
- EXEC CICS DUMP DUMPCODE(FAD1) - XDUREQ exit must be installed
- EXEC CICS ABEND ABCODE(FLT1)
- EXEC CICS ABEND ABCODE(FLT2) - LE Assembler

Use S to Select the IVP to execute
    PF3=Exit      ENTER=Execute

```

この画面で、実行するテストの横に S を入力し (EXEC CICS® DUMP DUMPCODE(FAD1) テスト用に XDUREQ 出口がインストール済みである必要があります)、Enter を押します。

DeferredReport は CICS® のデフォルト・オプションなので、各 IVP のヒストリー・ファイル名と障害 ID を [IDI0003I ページ 696](#) メッセージから判別し、その後、Fault Analyzer の ISPF インターフェースから各障害項目に対して「V」行コマンドを発行し、保管されたレポートを表示してください。

### CICS IVP: 0C1 in program IDIXFA

Fault Analyzer レポートの概要セクションには、「0C1 in program IDIXFA」テストについて、以下が含まれているはずで

```
A CICS abend ASRA occurred in module IDIXFA CSECT IDIXFA at offset X'5FC'.
```

```
A program interruption code 0001 (Operation Exception) is associated with this
abend and indicates that:
```

```
    An attempt was made to execute an instruction with an invalid operation code.
```

```
The abend was caused by an undetermined instruction.
```



```
NOTE: Source code information could not be presented because the search for a
       compiler listing or side-file was unsuccessful for CSECT IDIXFA.
```

この IVP の場合、Fault Analyzer は、XPCABND 出口を介して呼び出されることが予想されています。

### CICS IVP: EXEC CICS® DUMP DUMPCODE(FAD1)

Fault Analyzer レポートの概要セクションには、「EXEC CICS® DUMP DUMPCODE(FAD1)」テストについて、以下が含まれているはずですが。

```
Fault Analyzer was invoked using the EXEC CICS DUMP interface.
```

この IVP の場合、Fault Analyzer は XDUREQ 出口を介して呼び出されることが予想されています。

### CICS IVP: EXEC CICS® ABEND ABCODE(FLT1)

Fault Analyzer レポートの概要セクションには、「EXEC CICS® ABEND ABCODE(FLT1)」テストについて、以下が含まれているはずですが。

```
A CICS abend FLT1 occurred in module IDIXFA CSECT IDIXFA at offset X'666'.
```

```
The abend was caused by machine instruction 05EF (BRANCH AND LINK).
```

```
NOTE: Source code information could not be presented because the search for a
       compiler listing or side-file was unsuccessful for CSECT IDIXFA.
```

この IVP の場合、Fault Analyzer は、XPCABND 出口を介して呼び出されることが予想されています。

### CICS IVP: EXEC CICS ABEND ABCODE(FLT2)

Fault Analyzer レポートの概要セクションには、「EXEC CICS® ABEND ABCODE(FLT2)」テストについて、以下が含まれているはずですが。

```
A CICS abend FLT2 occurred in module IDIVPCLE CSECT IDIVACLE at offset X'98'.
```

```
The abend was caused by machine instruction 05EF (BRANCH AND LINK).
This was an EXEC CICS ABEND command.
```

```
NOTE: Source code information could not be presented because the search for a
       compiler listing or side-file was unsuccessful for CSECT IDIVACLE.
```

この IVP の場合、Fault Analyzer は CICS LE 出口 (インストールされている場合) を介して呼び出されることが予想されています。それ以外の場合、これは、XPCABND 出口を介して呼び出されることが予想されています。

## DB2 による Fault Analyzer の使用の検証

このセクションは、ご使用のサイトに DB2® がインストール済みである場合のみ適用できます。

DB2® による Fault Analyzer の検証には、2 つの異なる方式があります。

- C プログラムを使用する方法。

この IVP は必要なものを完備しているため、実行に先立って DB2® の特別なセットアップは必要ありません。詳しくは、[C プログラムを使用する方法 ページ 434](#)を参照してください。

- COBOL プログラムを使用する方法。

この IVP は実行する前に、サンプル DB2® データベースのセットアップが必要です。詳しくは、[COBOL プログラムを使用する方法 ページ 435](#)を参照してください。

## C プログラムを使用する方法

C プログラムを使用して DB2® で Fault Analyzer を検証するには、データ・セット IDIVPDB2 のサンプル・ジョブ IDI.SIDISAM1 を編集し、実行依頼します。詳細については、サンプル・ジョブ内の説明を参照してください。

ジョブは、既にコンパイルされてリンク済みの ODBC C プログラムを実行します。このプログラムは、データ・セット IDIVPDB2 内のロード・モジュール IDI.SIDIAUTH として提供されています。このプログラムは計画的にシステム異常終了コード S0C4 で異常終了します。



**注:** この IVP は、DB2® ODBC IVP に基づいています。これは通常、メンバー DSNTJ8 (JCL) および DSN80IVP (C ソース・コード) として DB2® の DSN.SDSNSAMP データ・セット内にあります。この IVP は、DB2® との接続中に故意に異常終了するように変更されています。これにより、Fault Analyzer が呼び出され、DB2® 情報のレポート・セクションが組み込まれます。Fault Analyzer バージョンのソース・コードが、参考のために IDIVPDB2 サンプル・メンバーの末尾に用意されています。

TER(UATRACE) LE オプションの結果として、Fault Analyzer は MVS™ 変更オプション/抑止ダンプ出口 IDIXDCAP から呼び出されます。

Fault Analyzer に書き込まれた IDIREPRT レポートの概要セクションには、以下が含まれます。

Fault Analyzer の日本語フィーチャーでは、Language(JPN) オプションが有効である場合、これは日本語になっているはずですが。

```
A system abend 0C4 reason code X'4' occurred in module IDIVPDB2 program IDIVCDB2
at offset X'C74'.
```

```
A program-interruption code 0004 (Protection Exception) is associated with this
abend and indicates that:
```

```
A protection exception occurred due to one of the following:
```

- An attempt to access a protected storage location using an incorrect storage access key.
- An attempt to store, in the access-register mode, by means of an access-list entry which has the fetch only bit set to one.

- An attempt to store into the range 0-511 or 4096-4607 with low-address protection enabled.
- An attempt to store into a protected page with DAT on.

The abend was caused by machine instruction 50000000 (STORE).

NOTE: Source code information for program IDIVCDB2 could not be presented because no compiler listing or side-file data sets were provided. The source line # from the GONUMBER option is 123 for offset X'C74'.

この IVP の実行により作成される完全なサンプル・レポートが、IDI.SIDIDOC1 データ・セット内のメンバー IDISRP04 として用意されています。

SYSPRINT に書き込まれるデータには、次のものがあります。

```

IDIVPDB2 INITIALIZATION
IDIVPDB2 SQLAllocEnv
IDIVPDB2-henv=1
IDIVPDB2 SQLAllocConnect
IDIVPDB2-hdbc=1
IDIVPDB2 SQLConnect
IDIVPDB2 successfully issued a SQLconnect
IDIVPDB2 SQLAllocStmt
IDIVPDB2 hstmt=1
IDIVPDB2 successfully issued a SQLAllocStmt
IDIVPDB2 SQLExecDirect
IDIVPDB2 sqlstmt=SELECT * FROM SYSIBM.SYSDUMMY1
IDIVPDB2 successfully issued a SQLExecDirect
IDIVPDB2 SQLFetch
IDIVPDB2 successfully issued a SQLFetch
IDIVPDB2 SQLTransact
IDIVPDB2 successfully issued a SQLTransact
IDIVPDB2 Abend S0C4 to invoke Fault Analyzer...

```

## COBOL プログラムを使用する方法

COBOL プログラムを使用して DB2® で Fault Analyzer を検証するには、データ・セット IDI.SIDISAM1 のサンプル・ジョブ IDIVPDBB を編集し、実行依頼します。このジョブは、COBOL プログラム・ソース・コードが入っている別のサンプル・メンバー IDISDB2B を入力として使用します。詳細については、サンプル・ジョブ内の説明を参照してください。

ジョブは COBOL プログラムをコンパイルし、実行します。このプログラムはシステム異常終了コード S0C9 で異常終了します。



**注:** この IVP は、DB2® COBOL IVP に基づいています。これは通常、メンバー DSNTEJ2C (JCL) および DSN8BC3 (C ソース・コード) として DB2® の DSN.SDSNSAMP データ・セット内にあります。この IVP は、DB2® データベース・アクセスを実行した後、故意に異常終了するように変更されています。これにより、Fault Analyzer が呼び出され、DB2® 情報のレポート・セクションが組み込まれます。



この IVP を実行する前に、DB2® サンプル・データベース環境が正しくセットアップされていることを確認する必要があります。DSNTEJ2C DB2® IVP の実行については、『DB2® for z/OS® Installation and Migration Guide』の手順に従ってください。DSNTEJ2C を正しく実行できたら、Fault Analyzer IDIVPDBB サンプルにリストされている変更を DB2® DSNTEJ2C サンプルに加えるか、または Fault Analyzer IDIVPDBB サンプルを DB2® DSNTEJ2C サンプルと同じように変更してください。

IDISDB2B プログラムに対する DB2® RUN コマンドに指定された TER(TRACE) LE オプションの結果、Fault Analyzer が LE CEEEXTAN 出口 IDIXCEE から呼び出されます

Fault Analyzer に書き込まれた IDIREPRT レポートの概要セクションには、以下が含まれます。

Fault Analyzer の日本語フィーチャーでは、Language(JPN) オプションが有効である場合、これは日本語になっているはずですが。



**注:** ご使用のコンパイラーのバージョン、リリースまたは保守レベルが異なる場合、プログラムのオフセット情報が下記の例とは異なることがあります。

```
A system abend 0C9 occurred in module IDISDB2B program IDISDB2B at offset
X'1EE2'.
```

```
A program-interruption code 0009 (Fixed-Point-Divide Exception) is associated
with this abend and indicates that:
```

```
    The divisor was zero in a signed binary division.
```

```
The cause of the failure was program IDISDB2B in module IDISDB2B. The COBOL
source code that immediately preceded the failure was:
```

```
Source
Line #
-----
001165          DIVIDE NOT-FOUND BY PERCENT-COUNTER
001166          GIVING ERROR-TEXT-LEN.
```

```
The COBOL source code for data fields involved in the failure:
```

```
Source
Line #
-----
000137          77 NOT-FOUND          PIC S9(9) COMP VALUE +100.
000146          77 ERROR-TEXT-LEN    PIC S9(9) COMP VALUE +120.
000207          *
000208          77 PERCENT-COUNTER    PIC S9(4) COMP.
```

```
Data field values at time of abend:
```

```
ERROR-TEXT-LEN = 120
```

```
NOT-FOUND          = 1
PERCENT-COUNTER    = 0  *** Cause of error ***
```

分析には、以下と似た DB2® 情報セクションが組み込まれます。



**注:** インストール先特定の名前や値は、下に示すサンプルのものとは異なる可能性があります。

```
-----
<H3> DB2 Subsystem DB42

DB2 Version . . . . . : V8R1M5
Plan Name . . . . . : DSN8BH81 (Bound 2006/08/25 14:30:52)
Plan Owner. . . . . : NWILKES
Database Request Module Name: DB2V810.DB42.DBRMLIB.DATA(IDISDB2B)
Consistency Token . . . . . : X'17E9C40018AE6A18'
Primary Authorization ID. . : NWILKES
Current SQL ID. . . . . : NWILKES

                                Source
                                Line #
                                -----
Last Executed SQL Statement : 001149          ***** EXEC SQL FETCH TELE1 INTO :PPHONE END-EXEC.

Fault Analyzer Event #. . . : 4 (Program IDISDB2B)
Declare Cursor Stmt No. . . : 200
Declare Cursor Stmt . . . : DECLARE TELE1 CURSOR FOR SELECT * FROM DSN8810 .
                                VPHONE
Open Cursor Stmt No . . . . : 346
Open Cursor Stmt. . . . . : OPEN TELE1

Output Host Variables:
Name and Data Type. . . . : PPHONE.LASTNAME VARCHAR(15)
  At Address. . . . . : 168A83D8
  Data Value. . . . . : HAAS

Name and Data Type. . . . : PPHONE.FIRSTNAME VARCHAR(12)
  At Address. . . . . : 168A83E9
  Data Value. . . . . : CHRISTINE

Name and Data Type. . . . : PPHONE.MIDDLEINITIAL CHARACTER(1)
  At Address. . . . . : 168A83F7
  Data Value. . . . . : I

Name and Data Type. . . . : PPHONE.PHONENUMBER CHARACTER(4)
  At Address. . . . . : 168A83F8
  Data Value. . . . . : 3978

Name and Data Type. . . . : PPHONE.EMPLOYEENUMBER CHARACTER(6)
  At Address. . . . . : 168A83FC
  Data Value. . . . . : 000010
```

```
Name and Data Type. . . . : PPHONE.DEPTNUMBER CHARACTER(3)
  At Address. . . . . : 168A8402
  Data Value. . . . . : A00

Name and Data Type. . . . : PPHONE.DEPTNAME VARCHAR(36)
  At Address. . . . . : 168A8405
  Data Value. . . . . : SPIFFY COMPUTER SERVICE DIV.
```

-----  
<H3> DB2 Control Blocks

SQL Communications Area (SQLCA) for subsystem DB42 not shown as it is identical to the SQLCA in the detail section for event # 4 program IDISDB2B.

## ISPF による Fault Analyzer の使用の検証

ISPF 使用環境で Fault Analyzer を検証するには、[ISPF 環境の変更 ページ 328](#)でセットアップした ISPF オプションから Fault Analyzer ISPF インターフェースを呼び出します。この呼び出しにより、検証プログラムの実行依頼の結果として発生した異常終了の項目を含む、「Fault Entry List」画面が表示されます。

初期状態では、デフォルトのヒストリー・ファイル IDI.HIST、または IDICNFxx parmlib メンバー DataSets オプションで指定されたヒストリー・ファイルが画面に使用されます。

異常終了時に生成された Fault Analyzer レポートを検査するには、参照する項目の横にコマンド V を入力し、Enter を押します。

Fault Analyzer ISPF インターフェースの使用の詳細については、[Fault Analyzer ISPF インターフェース ページ 62](#)を参照してください。

## リカバリー障害記録セットアップの検証

リカバリー障害記録のセットアップを検証するには、以下のような JCL ステートメントを他の IVP ジョブ (例: IDIVPCOB IVP ジョブ) の異常終了ステップに挿入します ([COBOL による Fault Analyzer の使用の検証 ページ 428](#)を参照してください)。

```
//GO.IDIRFRON DD DUMMY
```

IDIRFRON によって分析される異常終了ステップに Fault Analyzer DDname が割り振られている場合は、故意の異常終了 U0777 が発行されます。この異常終了は、に記載されているセットアップに応じて[リカバリー障害記録 ページ 57](#) TDUMP または SDUMP を書き込むため、またリカバリー障害記録障害項目を書き込むために、リカバリー障害記録機能を活動化します。

ジョブを実行依頼する前に、IDIS サブシステムが始動していることを確認してください。

以下のようなメッセージが表示されるはずです。

```
+IDI0001I Fault Analyzer V15R1M00 (HADQF10 2022/09/27)invoked by IDIXCEE using SYS1.PARMLIB.F1.USER(IDICNF00)
+IDI0157I Fault Analyzer about to deliberately abend U0777 and take RFR dump due to IDIRFRON DDname
+IDI0047S IBM Fault Analyzer internal abend. U0777
```

```
+IDI0126I Recovery fault recording fault ID BAT15874 assigned in history file DA.DCAT
IGD101I SMS ALLOCATED TO DDNAME (SYS00038) 524
      DSN (IDIRFRHQ.TDUMP.F1.D181215.T013821.S00072)
      STORCLAS (SCDIRFR) MGMTCLAS (PRIMARY) DATACLAS (DEFAULT)
      VOL SER NOS= E$RF01
IGD104I IDIRFRHQ.TDUMP.F1.D181215.T013821.S00072 RETAINED, DDNAME=SYS00038
IEA822I COMPLETE TRANSACTION DUMP WRITTEN TO IDIRFRHQ.TDUMP.F1.D181215.T013821.S00072
```

メッセージ [IDI0126I ページ 724](#) に示されたりカバリー障害記録障害項目の再分析の結果としてレポートが出力されます。このレポートは、IDIRFRON DD ステートメントなしで実行された同じ IVP のレポートと同一のもので、

## ダンプ登録の検証 (IEAVTSEL)

MVS ポスト・ダンプ出口 IDIXTSEL をインストールして ([MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#) を参照)、DumpRegistrationExits オプション ([DumpRegistrationExits ページ 588](#) を参照) でユーザー出口を指定し、IDIS サブシステムを起動すると ([IDIS サブシステムの始動 ページ 321](#) を参照)、以下を使用してダンプ登録をテストできるようになります:

- [SLIP ダンプの使用 ページ 439](#)
- [コンソール・ダンプの使用 ページ 439](#)

DumpRegistrationExits オプションで分析制御ユーザー出口または通知ダンプ登録ユーザー出口が指定されている場合は、WTO メッセージ (または使用するプログラミング言語に応じた同様のメッセージ) をロード・モジュール・ユーザー出口に追加するか、IDIWTO コマンドを REXX ユーザー出口に追加すると便利な場合があります。[IDIWTO コマンド ページ 537](#) を参照してください。

例えば、共通した通常の分析制御ユーザー出口とダンプ登録分析制御ユーザー出口を使用する場合、2つの出口タイプの呼び出しを区別する方法については [共通のユーザー出口を呼び出している出口タイプの判別 ページ 468](#) を参照してください。

## SLIP ダンプの使用

1. IDIS サブシステムが SLIP パラメーターで起動されたことを確認してください。
2. MVS オペレーター・コマンドを使用して SLIP トラップを設定します:

```
SL SET, ID=xxxx, COMP=U0777, A=SVCD, END
```

ここで、xxxx はご自身で選択した ID です。

3. 以下の JCL ステートメントを異常終了するジョブ (例えば、Fault Analyzer アセンブラー IVP ジョブ、IDIVPASM) に追加します:

```
//G.IDIRFRON DD DUMMY
```

4. ジョブを実行依頼します。

## コンソール・ダンプの使用

1. 次の MVS オペレーター・コマンドを発行します。

```
DUMP TITLE='xxx'
```

ここで、xxx はご自身で選択したタイトルです。

2. MVS が WTOR <n>IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND を発したら、以下のように応答します:

```
R <n>,ASID=1,END
```

ASID=1 の場合は、任意の有効な ASID 番号を置き換えたり、(例えば) `JOBNAME=name` や `TSONAME=name` に変更することができます。DUMP コマンドの詳細については、MVS システム・コマンドに関する資料を参照してください。



## 第 26 章. ヒストリー・ファイルの管理 (IDIUTIL ユーティリティー)

IDIUTIL バッチ・ユーティリティーは、ヒストリー・ファイル障害項目のリストや削除などのヒストリー・ファイル管理機能を実行するために使用できます。

日付やその他の基準に基づいて一連の項目を削除する機能により、管理可能なレベルでヒストリー・ファイルの項目の数を保持することができます。



**注:** Fault Analyzer ISPF インターフェースまたは IDIUTIL バッチ・ユーティリティーを使用せずに (例えば、ISPF データ・セット・メンバー・リストからの直接処理で)、ヒストリー・ファイル PDS または PDSE からメンバーを削除しないでください。そのようにすると、リアルタイム分析時に次の障害が記録されるまで、またはヒストリー・ファイルに対して IDIUTIL バッチ・ユーティリティーが実行されるまで、そのヒストリー・ファイル索引が同期しなくなります。

ヒストリー・ファイルに項目をリストできることで、問題を把握することができます。

IDIUTIL の保守機能は、SYSIN DD JCL データ・セットから読み取る一連の制御ステートメントにより呼び出されます。制御ステートメントは、SYSIN レコードの 1 列目から始まります。継続ステートメントの 1 列目はブランクです。この制御ステートメント・ストリームにコメントを書き込むには、コメントにする行の 1 列目にアスタリスクを挿入します。

SYSIN ストリームでは、制御ステートメントが 1 回に 1 つずつ、順次に処理されます。制御ステートメントのターゲット・ヒストリー・ファイルは、その制御ステートメントに応じて暗黙的または明示的に指定されます。ターゲット・ヒストリー・ファイルのデータ・セット名がセットアップされている制御ステートメントは、それまで有効だったターゲット・ヒストリー・ファイル名を上書きします。FILES 制御ステートメントは、それ以降の制御ステートメント用にターゲット・ヒストリー・ファイルのデータ・セット名を設定する目的に限り使用されます。この目的は、LISTHF および DELETE 制御ステートメント構文にターゲット・ヒストリー・ファイル・キーワードが含まれていない場合に意味を持ちます。これらの制御ステートメントは現行のターゲット・ヒストリー・ファイル・セットを処理します。

IMPORT、SETFAULTPREFIX などのこの他の制御ステートメントの場合は、構文内に 1 つのヒストリー・ファイルを指定します。この構文は、アクションを実行する前に、現行のヒストリー・ファイル・セットをこのデータ・セット名にリセットします。

SYSIN ストリームの代わりとして、IDIUTIL バッチ・ユーティリティー用の制御ステートメントを、EXEC JCL ステートメントの PARM フィールドを介して渡すことができます。この方法で渡される制御ステートメントには組み込みブランク・スペースを含めてはなりません。他の制御ステートメントからは 1 つ以上のブランクで分離する必要があります。

IDIUTIL バッチ・ユーティリティーは、日付やジョブ名の基準に基づいて障害項目を選択してリストまたは削除するなどの一連の基本的な機能を提供します。3 つのユーザー出口点により、より詳細な選択と、ユーザーによりコーディングされる記録が可能になります。これらの出口は、DELETE、LISTHF、および IMPORT 制御ステートメントで使用できます。これらは、Fault Analyzer リアルタイムおよび再分析機能に提供されるユーザー出口と同じ構造です。REXX、アセンブラー、または高水準言語で作成されます。

## IDIUTIL 制御ステートメント

ここでは、制御ステートメントが、使用について効率的な順序で説明されますが、実行に関する順序の要件として、LISTHF および DELETE 制御ステートメントは、ヒストリー・ファイルのデータ・セット名セットを移植するその他の制御ステートメントの後に実行される必要があります。ヒストリー・ファイルのデータ・セット名のセットに複数のデータ・セット名を含めることができるのは、FILES 制御ステートメントのみです。この他の制御ステートメント (LISTHF、DELETE、および Exits は除く) はすべて、現行のデータ・セット名を1つのターゲット・データ・セット名にリセットします。

ブランク、引用符 ( ' または " )、大なり記号 (>)、小なり記号 (<)、等号 (=)、アンパーサンド (&)、縦線 (|) を含む値を単一引用符または二重引用符で囲む必要があります。引用符で囲まれた値の中に、その値を囲むものと同じ種類の引用符がある場合は、当該の引用符を2つ重ねる必要があります。

次の構文図では、サブオプションまたは値を繰り返し指定する場合に、コンマまたはブランク文字のいずれかを区切り文字として使用できます。

## FILES 制御ステートメント

FILES 制御ステートメントは、PDS または PDSE ヒストリー・ファイル・データ・セット名のリストを指定します。

図 210. Syntax



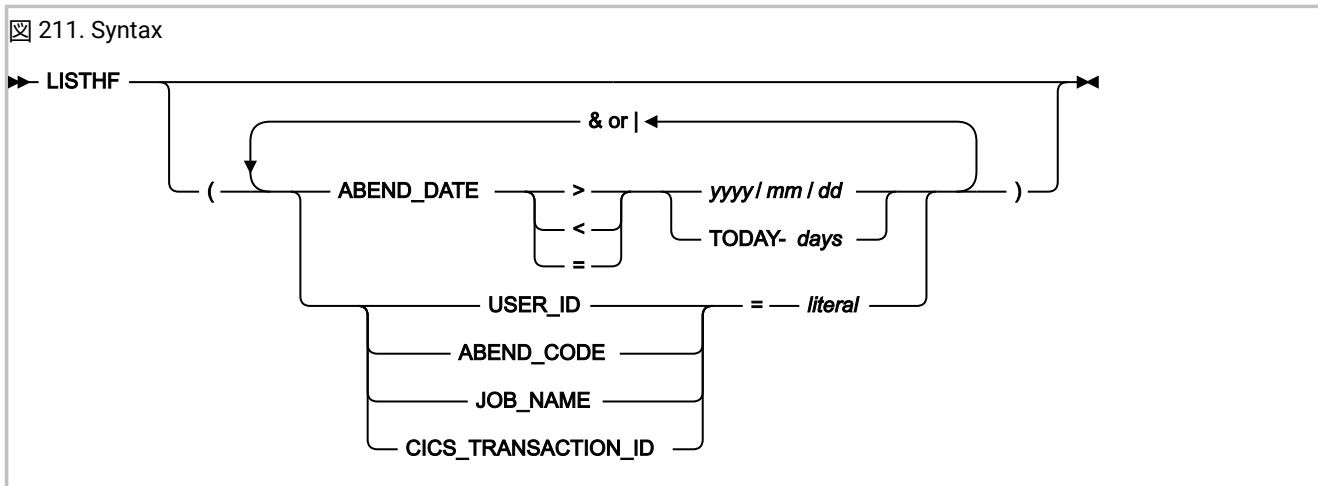
### 説明

FILES 制御ステートメントのすぐ後に続く LISTHF および DELETE 制御ステートメントは、FILES ステートメント内のすべてのヒストリー・ファイルを操作します。

FILES 制御ステートメントの使用法を示す例が、[例 ページ 451](#) に記載されています。

## LISTHF 制御ステートメント

LISTHF (LIST ヒストリー・ファイル) 制御ステートメントは、リストする障害項目を選択するためのオプションの修飾子のセットを指定します。



## 説明

修飾子の基本的な比較機能により、障害項目 ABEND\_DATE、USER\_ID、ABEND\_CODE、JOB\_NAME、または CICS\_TRANSACTION\_ID (すべて ENV データ域のフィールド名) が LISTHF 制御ステートメントのリテラルより大きいか、小さいか、または等しいかが示されます。比較は、and または (&) 演算子で結合できます。より複雑な比較を行う必要がある場合は、この単純構文機能の結果をユーザー出口に渡すことができます。

2つの特殊なリテラル比較修飾子が認識されます。リテラルのアスタリスクは、次のように、ワイルドカード機能により比較部分を省略できます。

```
JOB_NAME = AB*
```

他の1つの特殊なリテラルは TODAY-days です。これは、今日の日付から days で指定した数の日数を引いた数に変換され、比較の前に 2001/02/23 形式のストリングに変換されます。必然的に、TODAY-days リテラルは、次のように、ABEND\_DATE で使用される場合にのみ意味があります。

```
ABEND_DATE < TODAY-30
```

days に指定する値は、0 から 2147483647 までの範囲でなければなりません。

ABEND\_CODE を比較するときのフォーマットは、ユーザー異常終了コードの場合は4桁の数で、システム異常終了の場合はSで始まる3桁の16進数です。例えば、システム 0C4 の場合は S0C4、ユーザー 4038 異常終了の場合は 4038 です。CICS® 異常終了コードは、ASRA などの4文字の英字です。

IDIUTIL ListHF ユーザー出口 (IDIUTIL ListHF ユーザー出口 ページ 512 を参照) は、LISTHF 制御ステートメントとともに使用して、リストするヒストリー・ファイル項目に追加の選択基準を適用できます。

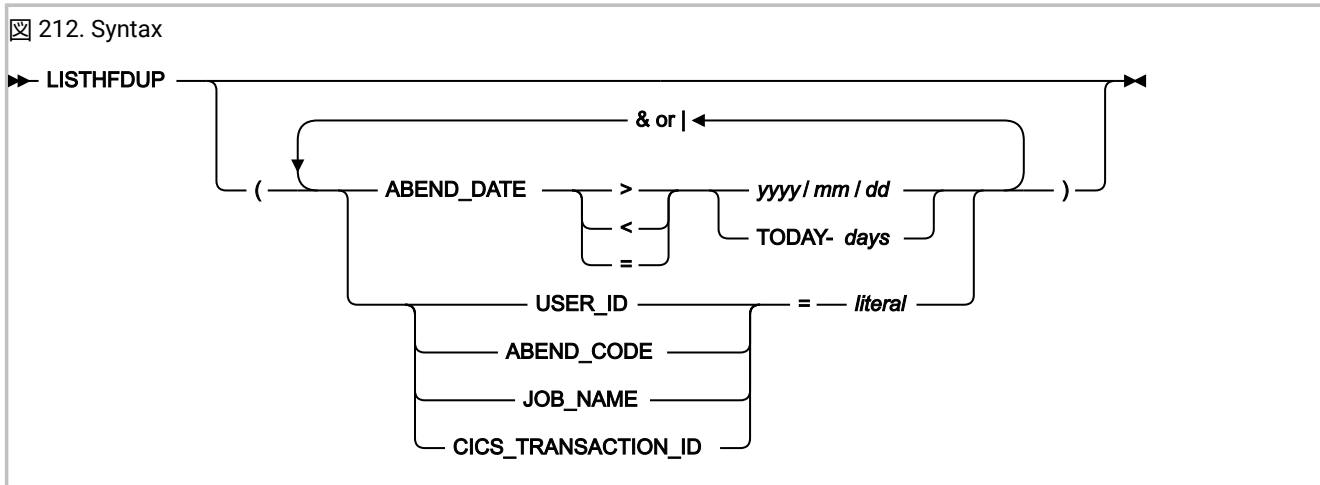
LISTHF 制御ステートメントの使用法を示す例が、例 1.ヒストリー・ファイル項目のリスト ページ 451 に記載されています。

## LISTHFDUP 制御ステートメント

LISTHFDUP (重複のあるリスト・ヒストリー・ファイル) は、障害のすべての重複インスタンスをリストできます。

LISTHFDUP 制御ステートメントは、障害項目だけでなく、障害のすべての重複インスタンスをリストできるため、LISTHF 制御ステートメントとは異なります。例えば、特定の障害項目に 10 個の重複が記録されている場合、LISTHFDUP はデフォルトで障害の 11 個すべてのインスタンスを個々のタイム・スタンプで個別にリストしますが、LISTHF は合計の重複数を持つ元の障害項目のみをリストします。

LISTHFDUP 制御ステートメントは、リストする障害インスタンスを選択するためのオプションの修飾子のセットを指定します。



### 説明

修飾子の基本的な比較機能により、障害項目 ABEND\_DATE、USER\_ID、ABEND\_CODE、JOB\_NAME、または CICS\_TRANSACTION\_ID (すべて ENV データ域のフィールド名) が LISTHFDUP 制御ステートメントのリテラルより大きい、小さいか、または等しいかが示されます。比較は、and または (&) 演算子で結合できます。より複雑な比較を行う必要がある場合は、この単純構文機能の結果をユーザー出口に渡すことができます。

2 つの特殊なリテラル比較修飾子が認識されます。リテラルのアスタリスクは、次のように、ワイルドカード機能により比較部分を省略できます。

```
JOB_NAME = AB*
```

他の 1 つの特殊なリテラルは TODAY-days です。これは、今日の日付から days で指定した数の日数を引いた数に変換され、比較の前に 2001/02/23 形式のストリングに変換されます。必然的に、TODAY-days リテラルは、次のように、ABEND\_DATE で使用される場合にのみ意味があります。

```
ABEND_DATE < TODAY-30
```

days に指定する値は、0 から 2147483647 までの範囲でなければなりません。

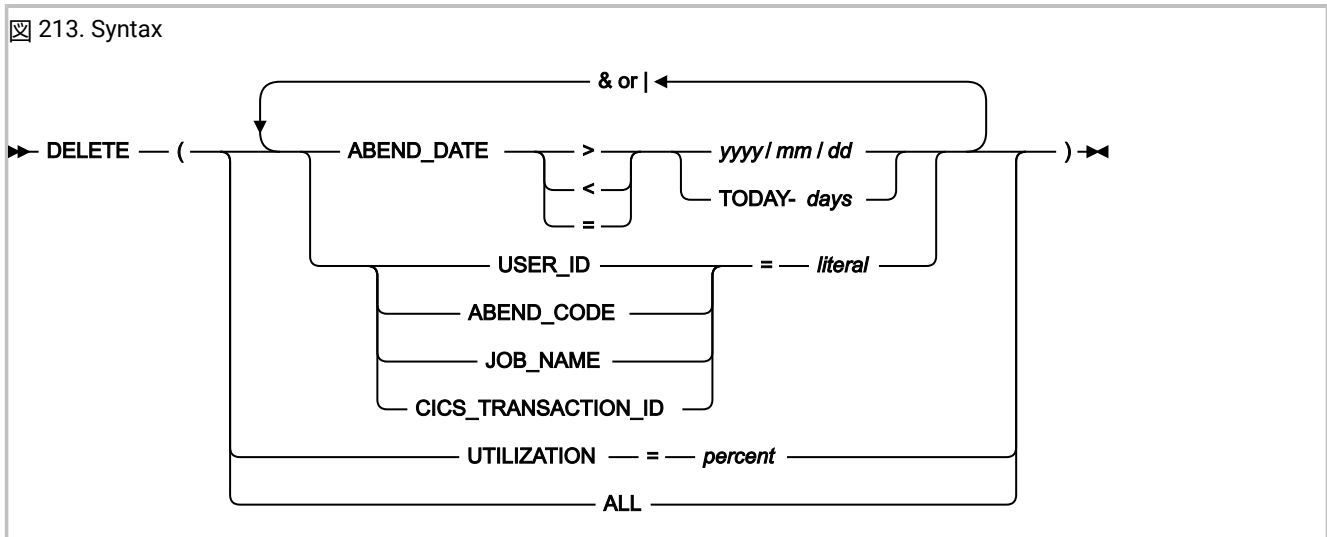
ABEND\_CODE を比較するときのフォーマットは、ユーザー異常終了コードの場合は 4 桁の数で、システム異常終了の場合は S で始まる 3 桁の 16 進数です。例えば、システム 0C4 の場合は S0C4、ユーザー 4038 異常終了の場合は 4038 です。CICS® 異常終了コードは、ASRA などの 4 文字の英字です。

IDIUTIL ListHFDUP ユーザー出口 ([IDIUTIL ListHFDUP ユーザー出口 ページ 517](#) を参照) は、ListHFDUP 制御ステートメントとともに使用して、リストする異常終了インスタンスに追加の選択基準を適用できます。

LI\$STHFDUP 制御ステートメントの使用法を示す例が、[例 2.ヒストリー・ファイル異常終了インスタンスのリスト ページ 452](#)に記載されています。

## DELETE 制御ステートメント

DELETE 制御ステートメントは、削除する障害項目を選択するための修飾子を指定します。



### 説明: DELETE(ALL)

DELETE(ALL) 制御ステートメントは、以前の FILES 制御ステートメントで無条件に指定されたヒストリー・ファイル内のすべての障害項目を削除するために使用します。障害項目はどのロック・フラグ設定とも関係なく削除され、ユーザー出口は呼び出されません。

関連するすべてのダンプ・データ・セット (RFR または XDUMP など) も削除されていることを確認するために、ヒストリー・ファイル自体を削除する前に、ヒストリー・ファイルに対して IDIUTIL を、DELETE(ALL) を指定して実行します。これが実行されないと、ヒストリー・ファイルが削除された場合でも、オーファンのダンプ・データ・セットが割り振られたままになる可能性があります。

**!** **重要:** この制御ステートメントを使用する場合、IDIUTIL を実行するユーザー ID には、指定されたヒストリー・ファイルに対する ALTER データ・セット・セキュリティー・アクセス権限が必要です。XFACILIT アクセス権では不十分です。

### 説明: DELETE(ALL) 以外

DELETE(ALL) を除き、IDIUTIL DELETE 関数は PDS ヒストリー・ファイルに対してのみ使用してください。PDSE ヒストリー・ファイルは、自動スペース管理を使用する必要があります。自動スペース管理については、[AUTO で管理される PDSE ヒストリー・ファイル ページ 347](#)を参照してください。

**🔑 注:** ロックされた障害項目の削除は、IDIUTIL 削除ユーザー出口を使用してデフォルトのアクションを指定変更した場合にのみ可能です。IDIUTIL 削除ユーザー出口の指定および使用方法については、[EXITS 制御ステートメント](#)



ページ 450を参照してください。ロック・フラグに関する一般情報、および IDIUTIL バッチ・ユーティリティを実行する前にロック・フラグの値を変更する方法に関する一般情報については、[障害項目情報の表示 ページ 134](#)を参照してください。

UTILIZATION 修飾子は、項目の削除に使用できます。古い項目から削除を開始し、指定したパーセントの使用率に達するまで実行されます。この修飾子は、PDSE ヒストリー・ファイルにのみ使用可能です。UTILIZATION を使用する場合、DELETE ステートメントの唯一の演算子である必要があり、& または | 演算子を追加して修飾することはできません。他の DELETE ステートメントの前に指定されるか、後に続く場合があります。



**注:** UTILIZATION 修飾子を使用した場合、IDIUTIL は、IDIUTIL の実行の時点で、指定されたパーセントまで使用率を削減しようと試みます。これは、新しい障害項目が作成されたときに、指定されたパーセントで使用率を維持するよう Fault Analyzer に指示する手段ではありません。PDSE ヒストリー・ファイルは、デフォルトで AUTO で管理され、可能な限り、追加のエクステントを割り振らずに、既存のデータ・セット・エクステントを使用します。

残りの修飾子は、上記の LISTHF 修飾子と同じ規則に従っています。

IDIUTIL 削除ユーザー出口を DELETE 制御ステートメントとともに使用すれば、削除するヒストリー・ファイル項目をさらに選択できます。[IDIUTIL 削除ユーザー出口 ページ 511](#)を参照してください。

DELETE 制御ステートメントの使用例を[例 3.日付によるヒストリー・ファイル項目の削除 ページ 452](#)と[例 4.使用率によるヒストリー・ファイル項目の削除 ページ 452](#)に示します。

## SETFAULTPREFIX 制御ステートメント

SETFAULTPREFIX 制御ステートメントは、PDS または PDSE ヒストリー・ファイル・データ・セット名とそれに続いて新しい接頭部を指定します。

☒ 214. Syntax

```
▶▶ SETFAULTPREFIX — ( — data-set-name — , — prefix — ) ▶▶
```

### 説明

*data-set-name* 障害項目の接頭部文字が変更された履歴ファイル。*prefix* は 3 文字まで指定できます。接頭部文字には英字のみを使用できます。

指定されたヒストリー・ファイル内の既存のすべての障害 ID の接頭部は、指定された接頭部に変更されます。このヒストリー・ファイル内で後で作成されたすべての障害 ID は、自動的にこの接頭部を受け取ります。



**注:** デフォルトでは、この機能は、ヒストリー・ファイルに対して UPDATE アクセス権を持つすべてのユーザーが使用できます。ただし、その使用は、[ヒストリー・ファイル設定の変更の制限 ページ 308](#)で説明されているとおりに、制限することができます。

SETFAULTPREFIX 制御ステートメントの使用法を示す例が、[例 5.ヒストリー・ファイル障害接頭部文字の変更 ページ 452](#)に記載されています。

## SETMAXFAULTENTRIES 制御ステートメント

SetMaxFaultEntries 制御ステートメントは、PDS ヒストリー・ファイル・データ・セット名および最大障害項目数を設定します。

図 215. Syntax

```
▶▶ SETMAXFAULTENTRIES — ( — data-set-name — , — max-number — ) ▶▶
```

### 説明

*max-number* は、ヒストリー・ファイルに維持できる障害項目の最大数を指定します。この障害項目数を達成するために、必要に応じて追加のデータ・セット・エクステン트가割り振られます。ヒストリー・ファイルに *max-number* の障害項目が入る前にデータ・セット内に使用可能なスペースがなくなった場合 (つまり、データ・セット・エクステン트의最大数に達するか、ボリュームが満杯になった場合)、スペース不足状態になります。

*max-number* は 1 から 2147483647 までのいずれかの数にする必要があります。



**注:** ヒストリー・ファイルにある障害項目の現在の数が、指定した障害項目の最大数を超過している場合、ロックされていない最も古い障害項目が削除され、指定した障害項目数のみが含まれます。

このオプションは、PDS ヒストリー・ファイルに対してのみ使用できます。PDSE ヒストリー・ファイルには SetMinFaultEntries を使用してください。



**注:** デフォルトでは、この機能は、ヒストリー・ファイルに対して UPDATE アクセス権限を持つすべてのユーザーが使用できます。ただし、その使用は、[ヒストリー・ファイル設定の変更の制限 ページ 308](#) で説明されているとおりに、制限することができます。

## SETMINFAULTENTRIES 制御ステートメント

SetMinFaultEntries 制御ステートメントは、PDSE ヒストリー・ファイル・データ・セット名および最小障害項目数を設定します。

図 216. Syntax

```
▶▶ SETMINFAULTENTRIES — ( — data-set-name — , — min-number — ) ▶▶
```

### 説明

このヒストリー・ファイル内の障害項目が最小数 *min-number* に達すると、このヒストリー・ファイルは自動的に維持されます (割り振られたデータ・セット・エクステン트의数に関係なく自動的に維持されます)。ヒストリー・ファイルが自動的に維持されるようになると、障害項目の数は、現在使用可能なデータ・セット・スペースによってのみ制限されます。追加のデータ・セット・エクステン트는通常は割り振られず、スペース不足状態は発生しないものと予想されます。

このオプションは、PDSE ヒストリー・ファイルに対してのみ使用できます。PDS ヒストリー・ファイルには SetMaxFaultEntries を使用してください。

AUTO で管理される PDSE ヒストリー・ファイルについて詳しくは、[AUTO で管理される PDSE ヒストリー・ファイル ページ 347](#) を参照してください。

min-number は 25 から 2147483647 までのいずれかの数にする必要があります。



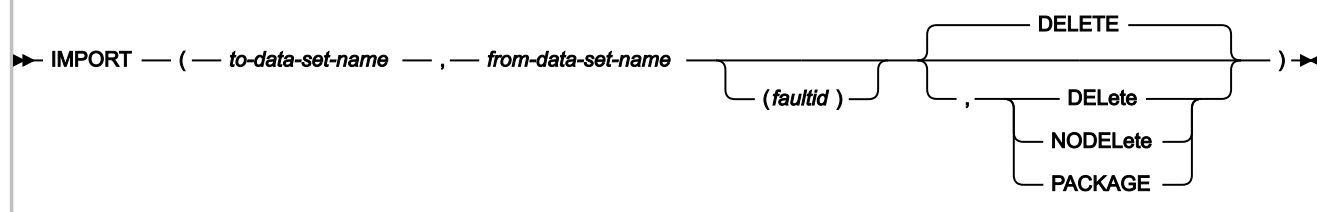
**注:** デフォルトでは、この機能は、ヒストリー・ファイルに対して UPDATE アクセス権を持つすべてのユーザーが使用できます。ただし、その使用は、[ヒストリー・ファイル設定の変更の制限 ページ 308](#) で説明されているとおりに、制限することができます。

SetMinFaultEntries 制御ステートメントの使用法を示す例が、[例 6.自己保持ヒストリー・ファイルの作成 ページ 453](#) に記載されています。

## IMPORT 制御ステートメント

IMPORT 制御ステートメントは、「インポート先」 PDS または PDSE ヒストリー・ファイル・データ・セット名とそれに続いて「インポート元」データ・セット名を指定します。

図 217. Syntax



## 説明

インポート元のデータ・セット内の障害項目がコピーされ、その接頭部文字がインポート先のヒストリー・ファイルの接頭部に設定されます。インポートされた障害番号は可能な限り同じままですが、既存のインポート先ヒストリー・ファイル障害項目が上書きされないようにする必要がある場合、障害番号は変更されます。DELETE オプションが有効な場合、正常にコピーされた後で、障害メンバーは「インポート元」データ・セットで削除されます。DELETE はデフォルトの設定です。

関連する密結合ダンプ・データ・セットには、自動的にコピーおよび新しい障害項目へのリンクが試みられます。アクセス制御と、障害項目が削除されるときにダンプ・データ・セットの自動削除を可能にする目的で、Fault Analyzer は障害項目とその関連する密結合ダンプ・データ・セットとの間の一意のリンクを保持するため、これが必要です。関連するダンプ・データ・セットの詳細は、[関連付けられたダンプ・データ・セット ページ 26](#) を参照してください。

コピーされたダンプ・データ・セットの名前は、以下のように決定されます。



- ダンプ・データ・セットのタイプに応じて、該当する IDIOPTLM 構成オプションのロード・モジュール・オプション RFRDSN、XDUMPDSN、SDUMPDSN を指定します。これは、デフォルトです。詳しくは、[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#)を参照してください。
- IDIUTIL インポート・ユーザー出口の使用。詳しくは、[IDIUTIL インポート・ユーザー出口 ページ 508](#)を参照してください。

インポート元データ・セット指定に単一の障害 ID (例: TEMP.HIST(F00234)) が含まれている場合は、その障害 ID のみがインポートされて削除されます。

この機能により、システムから TSO XMIT コマンドを使用して送信された障害メンバーは保留データ・セットに取り込まれ、次に、要求されたターゲット・ヒストリー・ファイルにインポートされます。この例については、[共用 DASD のない MVS システム全体でのヒストリー・ファイルの管理 ページ 353](#)を参照してください。この機能とともに IMPORT ユーザー出口を使用すると、項目のインポートが完了したことをユーザーに通知できます。

PACKAGE オプションは次の状況でのみ使用されます。

- IDIROBOT exec による ([IDIROBOT: 障害項目を受信するためのサンプル REXX EXEC ページ 356](#)を参照してください)。
- 「障害エンタリー・リスト」画面から P 行コマンドを使用して作成された障害項目パッケージをインポートする場合 ([障害項目のパッケージ化 ページ 146](#)を参照してください)。
- IDIXMIT フォーマット・ユーザー出口を使用して作成された障害項目パッケージをインポートする場合 ([オンデマンドの実装 ページ 361](#)を参照してください)。

障害項目パッケージの作成に使用した方法が何であっても、IMPORT 入力データ・セットは IDIUTIL の実行前に terse 解除される必要があります。

PACKAGE オプションを使用する場合、Fault Analyzer は IMPORT 入力データ・セットおよびパラメーターで追加のチェックを実行し、以下のことを確認します。

- *faultid* が指定されている場合、入力データ・セットに障害項目が 1 つだけ含まれているか、または *from-data-set-name* がシーケンシャルである。
- *from-data-set-name* に \$\$INDEX メンバーが含まれず、IDIS サブシステムから管理されない。

IDIUTIL インポート・ユーザー出口 ([IDIUTIL インポート・ユーザー出口 ページ 508](#)を参照) を IMPORT 制御ステートメントとともに使用すれば、インポートするヒストリー・ファイル項目をさらに選択できます。

IMPORT 制御ステートメントの使用法を示す例が、[例 7.ヒストリー・ファイル項目のインポート ページ 453](#)に記載されています。

## EXPORT 制御ステートメント

EXPORT 制御ステートメントを使用して、障害項目のポータブル・コピーを作成できます。これには障害項目そのものと、関連するすべてのダンプ・データ・セットの両方が含まれます。

図 218. Syntax

EXPORT — ( — *hist\_dsn* — ( *fault\_id* ) — , — *output\_dd* — ) →

### 説明

EXPORT 制御ステートメントは、主として IDIROBOT exec および Fault Analyzer ISPF インターフェースの P 行コマンドによって生成されたジョブによって使用されることを意図しています。

ターゲットの障害項目は、*hist\_dsn* および *fault\_id* を使用して指定されます。障害項目のポータブル・コピーは、*output\_dd* を使用して指定された DDname に書き込まれます。*output\_dd* に割り振られたデータ・セットは、RECFM VB および LRECL 10000 が適用された PDS または PDSE 一時シーケンシャル・データ・セットになると想定されています。既存のデータが破棄される可能性があるため、データ・セットは空にする必要があります。

障害項目への READ アクセス権は、通常データ・セット・アクセスまたは Fault Analyzer XFACILIT アクセスにより提供されますが、関連する密結合ダンプ・データ・セットへの READ アクセスは、通常データ・セット・アクセスにより提供される必要があります。

エクスポートされた障害項目は、IMPORT 制御ステートメントを使用して、別のヒストリー・ファイルに再度インポートする必要があります。

### EXITS 制御ステートメント

EXITS 制御ステートメントは、リアルタイム分析および再分析の Exits キーワードと同じフォーマットに従っています。Exits 制御ステートメントは、リアルタイム分析および再分析の Exits キーワードと同じフォーマットに従っています。出口点が LISTHF、LISTHFDUP、DELETE、および IMPORT 用であるという点で異なります。

図 219. Syntax

EXITS — ( — LISTHF — ( — *exit name specification* — ) — ) →

LISTHFDUP —

DELETE —

IMPORT —

exit name specification

→ *exit\_name* →

REXX — ( — *exit\_name* — ) →

## 説明

出口は、指定された選択修飾子と一致する LISTHF または DELETE ターゲット・データ・セットの各障害項目、および IMPORT のインポート元データ・セットにあるメンバーに呼び出されます。LISTHFDUP の場合、異常終了 (初期異常終了、通常重複、または個別の CICSFast または ImageFast 重複)、または異常終了のグループ (CICSFast または ImageFast) のインスタンスごとに出口が呼び出されます。重複のタイプが UTL.DUP\_TYPE に指定されています。いずれの場合も、UTL.PERFORM\_ACTION フラグはデフォルトで「Y」に設定されます (ただし、IDIUTIL ユーザー出口がロックされた障害項目に呼び出される場合を除きます)。この場合、ENV.LOCK\_FLAG がブランクでないと、ユーザー出口に制御が渡される前に、UTL.PERFORM\_ACTION フラグに「N」が設定されます。

EXITS 制御ステートメントは、後に続く LISTHF、LISTHFDUP、DELETE、または IMPORT 制御ステートメントに対して有効です。またはユーティリティの実行で新しい EXITS 制御ステートメントが検出されるまで有効です。複数の EXITS 制御ステートメントが累積して有効になることはありません。新しい EXITS 制御ステートメントが検出されたときに、それ以前の出口はクリアされます。IDIUTIL 実行の始めにアクティブな初期ユーザー出口はありません。LISTHF、LISTHFDUP、DELETE、および IMPORT 出口点は、Fault Analyzer リアルタイム分析および再分析によって使用される構成ファイルで認識されることも、この構成ファイルから読み取られることもありません。

非推奨オプション ACCOUNTING および NOACCOUNTING は引き続き指定できますが、無視されます。

各出口タイプの詳細記述は、[ユーザー出口を使用した Fault Analyzer のカスタマイズ ページ 464](#) に示されています。

EXITS 制御ステートメントの使用法を示す例が、[例 7. ヒストリー・ファイル項目のインポート ページ 453](#) に記載されています。

## 例

IDIUTIL バッチ・ユーティリティの使用法を示す例は、次のとおりです。

### 例 1. ヒストリー・ファイル項目のリスト

この例に示している IDIUTIL バッチ・ユーティリティ・ジョブは、ヒストリー・ファイル MY.HIST1 に含まれているすべてのヒストリー・ファイル項目をリストし、さらに、TEMP で始まるジョブ名用であり、異常終了コード S0C1 を含み、ユーザー ID P0001 用である、ヒストリー・ファイル MY.HIST2 内のすべての項目をリストします。

```
//UTILJOB1 JOB ...
//RUNUTIL EXEC PGM=IDIUTIL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
* In the first history file, list all entries
FILES(MY.HIST1)
LISTHF
* In the second history file, only list those entries that match a specific criteria
FILES(MY.HIST2)
LISTHF(USER_ID=P0001 & ABEND_CODE=S0C1 &
JOB_NAME=TEMP*)
/*
```

## 例 2. ヒストリー・ファイル異常終了インスタンスのリスト

この例は、昨日発生したヒストリー・ファイル MY.HIST1 に記録された異常終了インスタンスを、重複するかどうかにかかわらずすべてリストする IDIUTIL バッチ・ユーティリティ・ジョブを示しています。

```
//UTILJOB1 JOB ...
//RUNUTIL EXEC PGM=IDIUTIL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
FILES(MY.HIST1)
LISTHFDUP(ABEND_DATE = TODAY-1)
/*
```

## 例 3. 日付によるヒストリー・ファイル項目の削除

この例に示している IDIUTIL バッチ・ユーティリティ・ジョブは、ヒストリー・ファイル MY.HIST1 および MY.HIST2 内にある、2 週間より多く経過したすべてのヒストリー・ファイル項目を削除します。

```
//UTILJOB2 JOB ...
//RUNUTIL EXEC PGM=IDIUTIL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
FILES(MY.HIST1,MY.HIST2)
DELETE(ABEND_DATE < TODAY-14)
/*
```

## 例 4. 使用率によるヒストリー・ファイル項目の削除

この例に示している IDIUTIL バッチ・ユーティリティ・ジョブは、ヒストリー・ファイル MY.HIST1 および MY.HIST2 内にあるヒストリー・ファイル項目を、ヒストリー・ファイル使用率が 80 パーセント未満になるまで削除します。まず古い項目から削除されます。

```
//UTILJOB2 JOB ...
//RUNUTIL EXEC PGM=IDIUTIL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
FILES(MY.HIST1,MY.HIST2)
DELETE(UTILIZATION = 80)
/*
```

この機能を実行するには、ヒストリー・ファイルは PDSE である必要があります。

## 例 5. ヒストリー・ファイル障害接頭部文字の変更

この例に示している IDIUTIL バッチ・ユーティリティ・ジョブは、ヒストリー・ファイル MY.HIST の障害接頭部文字を ABC に変更します。

```
//UTILJOB3 JOB ...
//RUNUTIL EXEC PGM=IDIUTIL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
SETFAULTPREFIX(MY.HIST,ABC)
/*
```

## 例 6. 自己保持ヒストリー・ファイルの作成

この例に示しているジョブは、MY.HIST という名前の PDSE ヒストリー・ファイルを割り振り、IDIUTIL バッチ・ユーティリティを使用して、障害折り返し番号を 100 に設定すると同時に、既存の障害番号の今後の再使用を許可します。

```
//UTILJOB4 JOB ...
//IDCAM5 EXEC PGM=IDCAM5
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  ALLOC DSNAME('MY.HIST')      -
        NEW                    -
        SPACE(10 10)           -
        CYLINDERS              -
        RECFM(V B)             -
        LRECL(10000)           -
        DIR(10)                -
        DSNTYPE(LIBRARY)
/*
//RUNUTIL EXEC PGM=IDIUTIL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
SetMinFaultEntries(MY.HIST,100)
/*
```

このコードにより、最大自動フリー・スペース再利用機能を備えたヒストリー・ファイルが作成され、スペース不足が生じなくなります (ただし、割り振られたデータ・セットに、SetMinFaultEntries で指定した最小数の障害を保持できる場合に限りです)。

また、新しい各ヒストリー・ファイルの障害 ID を固有のものにするために、ご使用のヒストリー・ファイル作成ジョブに SETFAULTPREFIX 制御ステートメントを追加することもできます ([SETFAULTPREFIX 制御ステートメント ページ 446](#)を参照)。

## 例 7. ヒストリー・ファイル項目のインポート

この例に示している IDIUTIL バッチ・ユーティリティ・ジョブは、システム名 CICS04 で発生したすべてのヒストリー・ファイル項目を MY.TEMP.HIST から MY.HIST にインポートします。この例ではシステム名をテストする必要があるため、IDIUTIL インポート・ユーザー出口が必要です。

MY.TEMP.HIST に次の障害が含まれていると仮定します。

Fault ID	Job/Tran	User	System	Abend	Date	Time
F00101	IDIVPCOB	NWILKES	MVS2	S0C7	2001/10/18	08:57:08
F00098	FRED	CICSUSER	CICS02	AEIL	2001/10/15	14:33:30
F00097	WILMA	CICSUSER	CICS04	AEIL	2001/10/15	13:00:57
F00096	BARNEY	CICSUSER	CICS02	AEIL	2001/10/15	12:56:32
F00095	BUSHBY2N	NWILKES	MVS2	U4038	2001/10/14	10:41:29
F00093	BETTY	CICSUSER	CICS04	ASRA	2001/10/12	21:16:37
F00092	DACBB045	NWILKES	MVS2	U4038	2001/10/10	10:38:22

MY.HIST に次の障害が含まれていると仮定します。

Fault ID	Job/Tran	User	System	Abend	Date	Time
F00030	BUSHBY2A	BUSHBYD	MVS2	U4038	2001/09/18	13:02:02
F00060	IMSLE4	NWILKES	MVS1	S0C9	2001/09/12	12:39:27
F00059	IMSLE3	NWILKES	MVS2	U4036	2001/09/12	12:38:31

この場合、次の JCL を

```
//UTILJOB5 JOB ...
//RUNUTIL EXEC PGM=IDIUTIL
//IDIEXEC DD DISP=SHR,DSN=MY.REXX.EXECS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
Exits(IMPORT(REXX(IMPXMP)))
IMPORT(MY.HIST,MY.TEMP.HIST)
/*
```

データ・セット MY.REXX.Exits のメンバー IMPXMP 内の IDIUTIL インポート・ユーザー出口とともに実行します。

```
/* REXX */
If ENV.SYSTEM_NAME = 'CICS04' then UTL.PERFORM_ACTION = 'N'
```

この結果、MY.TEMP.HIST 内の障害項目は次のようになります。

Fault ID	Job/Tran	User	System	Abend	Date	Time
F00101	IDIVPCOB	NWILKES	MVS2	S0C7	2001/10/18	08:57:08
F00098	FRED	CICSUSER	CICS02	AEIL	2001/10/15	14:33:30
F00096	BARNEY	CICSUSER	CICS02	AEIL	2001/10/15	12:56:32
F00095	BUSHBY2N	NWILKES	MVS2	U4038	2001/10/14	10:41:29
F00092	DACBB045	NWILKES	MVS2	U4038	2001/10/10	10:38:22

MY.HIST 内の障害項目は次のようになります。

Fault ID	Job/Tran	User	System	Abend	Date	Time
F00030	BUSHBY2A	BUSHBYD	MVS2	U4038	2001/09/18	13:02:02
F00031	WILMA	CICSUSER	CICS04	AEIL	2001/10/15	13:00:57
F00032	BETTY	CICSUSER	CICS04	ASRA	2001/10/12	21:16:37
F00060	IMSLE4	NWILKES	MVS1	S0C9	2001/09/12	12:39:27
F00059	IMSLE3	NWILKES	MVS2	U4036	2001/09/12	12:38:31

MY.HIST にインポートされた項目が、MY.TEMP.HIST から削除されたことに注意してください。

IDIUTIL バッチ・ユーティリティー・インポート機能の使用法を示す追加の例が、[共用 DASD のない MVS システム全体でのヒストリー・ファイルの管理 ページ 353](#)に示されています。

## IDIUTIL バッチ・ユーティリティー・ユーザー出口の例

IDIUTIL バッチ・ユーティリティーのユーザー出口の例は、[IDIUTIL バッチ・ユーティリティー・ユーザー出口タイプの説明 ページ 508](#)にあります。

特に、[例 2: IDIUTIL ListHF ユーザー出口を用いた障害項目のカスタム・レポートおよび CSV ファイルの作成 ページ 514](#)の例は、カスタマイズ・レポートの作成方法やスプレッドシート・アプリケーションへの入力として使用できるコマンド区切りファイルをどのように生成するかを示しています。

## 第 27 章. アプリケーション固有の説明の指定

IDIHUSRM データ・セットのメンバー IDI.SIDIDOC1 は、ユーザー定義のメッセージおよび異常終了コードの説明に加えて、プログラム名の説明も保持します。

IDIHUSRM の構造は簡潔です。各項目 (メッセージの説明、異常終了コードの説明、またはプログラムの説明) はヘッダー・レコードで識別され、その後その項目に対して表示されるテキストが続きます。本文は、複数のレコードに含まれる場合があります。項目は、新規のヘッダー・レコードまたはファイルの終わりによって完結します。各レコードはストリングです。項目の保守にはテキスト・エディターを使用します。



**注:** Fault Analyzer IDIS サブシステムを使用する場合は、新規の説明または変更された説明をストレージ内キャッシュに組み込むために IDIS サブシステムを停止してから再始動する必要があります。再始動を行わないと、新規の説明や変更された説明は検出されません。

Fault Analyzer は、`.*` で始まる IDIHUSRM 内の行をコメントとして扱い、無視します。これは分析レポートには含まれません。

### ユーザー定義メッセージの説明

何かしらのアプリケーションが SYSLOG にメッセージを書き込んだ後に、そのアプリケーションが異常終了した場合、Fault Analyzer は IDIHUSRM を調べて、発行されたメッセージの ID に関連するメッセージを見つけることができるかどうかを確認します。適合するものが検出された場合、Fault Analyzer は分析レポートにメッセージの説明を組み込むことができます。

IDIHUSRM 内のメッセージ説明ヘッダー・レコードのフォーマットは以下のとおりです。

図 220. Syntax

```
►► :msg.message_identifier ◀◀
```

それぞれの意味は以下のとおりです。

#### **message\_identifier**

メッセージを識別するコード。このコードは、任意の数字と文字の組み合わせを使用できます。大文字小文字の区別はありません。例:

```
MYMESSAGE  
m103a  
dbg303
```

これらはすべて有効なメッセージ ID です。

メッセージの説明を提供する目的で、Fault Analyzer は、すべてのメッセージの開始位置から最初の空白文字までの間のすべての文字を、検索するためのメッセージ ID を構成するものとして扱います。

メッセージが表示されるときに MVS™ がメッセージの開始位置に追加することがある正符号 (+) は、メッセージ ID の一部とは見なされません。これらが IDIHUSRM メンバーに組み込まれていると、無視されます。

“msg.” は 1 桁目から開始する必要があることに注意してください。

ヘッダー・レコードに続くメッセージの行は、入力したとおりに直接転送されます。そのため、説明の先頭に空白を含めると、Fault Analyzer がメッセージを印刷するときに、先頭の空白が含まれます。

次の例では、2つのメッセージが定義されています。

```
:msg.payrollmsg1
Processing of accumulated leave about to commence.
- If processing fails after this point,
  check for negative accumulations.
:msg.payrollmsg2
Processing of accumulated leave completed.
```

LOOKUP コマンドを使用すると、ユーザー定義メッセージが「Messages」メイン・カテゴリのサブカテゴリ名(メッセージ ID の最初の 3 文字で構成される)の下に表示されます。

## ユーザー定義の異常終了コードの説明

IDIHUSRM 内の異常終了コード説明ヘッダー・レコードのフォーマットは以下のとおりです。

```
☒ 221. Syntax
▶▶ :abend.abend_code ▶▶
```

それぞれの意味は以下のとおりです。

### **abend\_code**

CICS または非 CICS 異常終了コード指定のいずれかです。

CICS 異常終了コード指定は、「CICS」という文字の後に 4 桁の CICS 異常終了コードが続きます。これには、大/小文字の区別がありません。例:

```
CICSFLT1
cicsa123
Cics0100
```

これらはすべて有効な CICS 異常終了コード指定です。

非 CICS ユーザー異常終了コード指定は、3 文字のプレフィックスの後に 4 桁のユーザー異常終了コードが続きます。大文字小文字の区別はありません。例:

"

```
XYZ0001
prd1234
zzz0999
```

これらはすべて有効な異常終了コード指定です。

3 文字の異常終了コード接頭部は、ユーザー異常終了が発行されたロード・モジュール名を識別するために使用されます。つまり、異常終了が発生したロード・モジュール名の最初の 3 文字は、指定された異常終了コード接頭部と一致しなければなりません。この一致により、異なるロード・モジュールから同じユーザー異常終了コードを発行でき、それぞれに固有の説明を付けることができます。



“:abend.”は1桁目から開始する必要があることに注意してください。

以下の例では、2つの非 CICS ユーザー異常終了が定義されています。U1888 異常終了は、名前が MD1 という3文字で始まるロード・モジュール (例: MD100P) によって発行される可能性があります。U0016 異常終了は、名前が EXT という3文字で始まるロード・モジュール (例: EXTMAIN) によって発行される可能性があります。IDIHUSRM は以下のように指定されます。

```
:abend.MD11888
An error occurred when attempting to write the invoice
to DDname SYSPRINT.
:abend.EXT0016
Incorrect EXTRACT option specified.
The problem might be one of the following:
- Invalid range
- Typo
Respecify and try again.
```

LOOKUP コマンドを使用すると、ユーザー定義の異常終了コード説明が「Abend Codes」メイン・カテゴリーのサブカテゴリー「Other Abend Codes」の下で、3文字のロード・モジュール名接頭部に従ってさらに複数のカテゴリーに分類されて表示されます。

## ユーザー定義プログラムの説明

IDIHUSRM 内のプログラム説明ヘッダー・レコードのフォーマットは以下のとおりです。

図 222. Syntax

```
►► :misc.PROGDESC: program_name ◄◄
```

それぞれの意味は以下のとおりです。

### *program\_name*

ユーザー・アプリケーション・エン트리・ポイント、プログラム、またはロード・モジュールの名前。大文字小文字の区別はありません。例:

```
XYZMAIN
prd01
zzzpayr1
```

これらはすべて有効な指定です。

最初に、一致するエン트리・ポイント名が検索されます。一致するエン트리・ポイント名が見つからない場合は、次にプログラム名が検索されます。一致するプログラム名も見つからない場合は、次にロード・モジュール名が検索されます。

一致するエン트리・ポイント、プログラム、またはロード・モジュール名が見つかった場合は、その説明が Fault Analyzer レポート Event Summary の Description 見出しの下、およびイベントの詳細セクションに表示されます。

“:misc.”は1桁目から開始する必要があることに注意してください。

以下の例では、2つのプログラム説明が指定されています。

```
:misc.PROGDESC:accrevbl  
Accounts receivable main module  
:misc.PROGDESC:errrtn  
Common error routine
```

LOOKUP コマンドを使用すると、ユーザー定義プログラムの説明が「Miscellaneous Information」メイン・カテゴリーのサブカテゴリー「Program Descriptions」の下に表示されます。

## 第 28 章. Fault Analyzer の保守

Fault Analyzer は標準 IBM® APAR/PTF サービスを使用して保守されます。



**注:** ADFzCC の保守の詳細は、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*を参照してください。

最新の Fault Analyzer for z/OS サービスの情報については、<https://www.ibm.com/support/pages/latest-fault-analyzer-zos-service-information>を参照してください。

++APAR フィックス・テストは、PTF で置き換えられるか、後続の ++APAR フィックス・テストで後の SMP/E REWORK 日付が指定されるため、復元する必要はありません。しかし、保守で競合が発生する場合は、SMP/E 復元と USERMOD の再適用を行う必要がある場合があります。

SMP/E APPLY によって Fault Analyzer に保守が適用されるか、SMP/E RESTORE によって Fault Analyzer から保守が削除されるたびに、以下の手順を実行する必要があります。

### ステップ 1: LLA および VLF 制御から LINKLIST 内のライブラリーを除外する

以下のライブラリーが LINKLIST にある場合、SMP/E APPLY を実行する前に、それらを LLA および VLF 制御から除外してください。

- Fault Analyzer SMP/E ターゲット・ライブラリー
- その他の製品のライブラリーを更新する Fault Analyzer によって提供される USERMODs のターゲット・ライブラリー

ライブラリーを削除すると、SMP/E がライブラリーを圧縮したり、ライブラリーにエクステントを追加したりした場合に、モジュールの読み込み中に発生するエラーを防ぐことができます。

### ステップ 2: IPL または更新を動的に実行する

CLPA で IPL 代替方法として、以下のような動的更新の実行。

1. Fault Analyzer モジュール IDIDA が SETPROG コマンドを使用して LPA に配置されている場合は、以下を行います。
  - a. 次のコマンドを発行します。

```
SETPROG LPA,DELETE,MOD=(IDIDA),FORCE=YES
```

(SETPROG コマンドの使用の詳細は、「MVS™ システム・コマンド」を参照してください。)

- b. 次のコマンドを発行します。

```
F LLA,REFRESH
```

オプションとして、LPA にモジュール IDIDA を再度追加して領域サイズ・スペースの利点を取り戻すには、次のコマンドを発行します。

```
SETPROG LPA,ADD,MOD=(IDIDA),DSN=LNKLST
```

2. CICS® を使用している場合は、インストールされたすべての CICS® 出口をリフレッシュします。このリフレッシュは、[CICS トランザクション異常終了分析の制御 ページ 405](#)に記載されている CFA トランザクションを使用して、出口をまずアンインストールし、次に再インストールすることによって行うことができます。CICS® 領域を再始動する必要はありません。
3. [Fault Analyzer IDIS サブシステムの使用 ページ 319](#)の記載どおりに、Fault Analyzer IDIS サブシステムを停止して再始動します。
4. Fault Analyzer ISPF インターフェイスを使用している場合は、更新内容を反映させるために一度終了してから再度 ISPF を開始します。
5. 次のオペレーター・コマンドを発行して IEAVTABX\_EXIT をリフレッシュします。

```
SET PROG=xx
```

xx は、次を含む PARMLIB PROGxx メンバーの接尾部に一致します。

```
EXIT REPLACE EXITNAME(IEAVTABX_EXIT) MODNAME(IDIXDCAP) STATE(ACTIVE)
```

### ステップ 3: サービス・レベルを検証する (オプション)

Fault Analyzer のインストール状況および適用される保守に関する情報は、次の方法で使用できます。

- Fault Analyzer ISPF インターフェイスの **Services->Service Information** プルダウン・メニュー。
- IDICHKI ユーティリティーで以下のようなジョブを実行依頼する:

```
//jobname JOB
//EXEC PGM=IDICHKI
//SYSPRINT DD SYSOUT=*
```

どちらの方法でも提供される情報は類似しており、以下が含まれています。

#### IEAVTABX 終了状況

動的な IEAVTABX 出口をテストし、IDIXDCAP が IEAVTABX に静的に定義されている場合には、z/OS V2.2 でエラーを報告します。Z/OS V2.2 より前のシステムについての情報は表示されません。

#### IEAVTSEL 終了状況

Fault Analyzer ポスト・ダンプ出口が定義されているかどうかを報告します。

#### Fault Analyzer ロード・モジュール

重要なロード・モジュールのサービス情報が示されています。

#### CEEEXTAN および CEECX TAN 出口に関する情報 (言語環境)

さまざまな Fault Analyzer LE 出口が定義されているかどうかを検査します。

#### IDIOPTLM 情報

IDIOPTLM が定義されている場合は、その内容を報告します。

## 第 29 章. Fault Analyzer の無効化

Fault Analyzer のリアルタイム起動をシステム・レベルまたはジョブ・レベルで使用不可にするためのオプションが各種あります。それらについて、以下のセクションで説明します。

使用する方法にかかわらず、SMF タイプ 89 レコードの書き込みもできなくなります。

Fault Analyzer CICS 出口は、CFA トランザクションを使用して、指定された CICS 領域からアンインストールすることもできる点に注意してください。これは Fault Analyzer を特定の CICS 領域からのみアンインストールするのに使用でき、他には影響を与えません。詳しくは、[CICS トランザクション異常終了分析の制御 ページ 405](#)を参照してください。

### Fault Analyzer を一時的にアンインストール

何らかの理由で一時的に Fault Analyzer をアンインストールする必要がある場合、以下の方法が最も簡単です。

1. IDI.SIDIAUTH データ・セット内の、次の呼び出し出口ロード・モジュールの名前を変更します (例えば最初の文字を「I」から「O」に変更する)。

```
IDIXCCEE
IDIXCEE
IDIXCX53
IDIXDCAP
IDIXTSEL
```

2. 次の MVS™ オペレーター・コマンドを発行します。

```
F LLA,REFRESH
```

3. 次の MVS™ オペレーター・コマンドを発行します。

```
SETPROG EXIT,MODIFY,EXITNAME=IEAVTABX_EXIT,MODNAME=IDIXDCAP,STATE=INACTIVE
```

4. CICS® を使用している場合は、すべての CICS® システムをバウンスするか、Fault Analyzer が提供する CFA トランザクションを使用して、すべてのインストール済み CICS® 呼び出し出口をアンインストールします。

上記を実行すると、出口での処理は必要なロード・モジュールを検出できず、それら抜きで処理が続行されます。



**注:** 保守を適用する前に、必ずロード・モジュールの名前を元に戻してください。

**再インストール:** Fault Analyzer を再インストールするには、以下を実行します。

1. 上記の手順のステップ [1 ページ 461](#) で名前変更したロード・モジュールの名前を元に戻します。
2. 次の MVS™ オペレーター・コマンドを発行します。

```
F LLA,REFRESH
```

3. 次の MVS™ オペレーター・コマンドを発行します。

```
SETPROG EXIT,MODIFY,EXITNAME=IEAVTABX_EXIT,MODNAME=IDIXDCAP,STATE=ACTIVE
```

4. CICS® を使用している場合は、すべての CICS® システムをバウンスするか、Fault Analyzer が提供する CFA トランザクションを使用して、すべての必要な CICS® 呼び出し出口をインストールします。

## IFAPRDxx parmlib メンバーを使用して Fault Analyzer をオフにする

特定の z/OS® イメージで Fault Analyzer を実行できないようにする必要がある場合は、IFAPRDxx parmlib メンバーに項目を追加することによってこれを行います。

Fault Analyzer を製品コード 5755-A01 IBM Application Delivery Foundation for z/OS の一部として購入した場合は、製品コード 5755-A01 の項目を次のように変更します。

```
PRODUCT OWNER('IBM CORP')
  NAME('IBM APP DLIV FND')
  ID(5755-A01)
  VERSION(*) RELEASE(*) MOD(*)
  FEATURENAME('FAULT-ANALYZER')
  STATE(DISABLED)
```

Fault Analyzer を製品コード 5655-PDS、IBM Problem Determination Solution Pack for z/OS の一部として購入した場合は、製品コード 5655-PDS の項目を次のように変更します。

```
PRODUCT OWNER('IBM CORP')
  NAME('IBM PD SOLTN PAC')
  ID(5655-PDS)
  VERSION(*) RELEASE(*) MOD(*)
  FEATURENAME('PROB-DET-SOL-PAC')
  STATE(DISABLED)
```

Fault Analyzer を製品コード 5655-DMS、IBM® Problem Determination Modernization Solution Pack for z/OS® の一部として購入した場合は、製品コード 5655-DMS の項目を次のように変更します。

```
PRODUCT OWNER('IBM CORP')
  NAME('IBM PD MD SO PAC')
  ID(5655-DMS)
  VERSION(*) RELEASE(*) MOD(*)
  FEATURENAME('PR-DE-MD-SOL-PAC')
  STATE(DISABLED)
```

Fault Analyzer を製品コード 5697-CDT、IBM® Enterprise COBOL Suite for z/OS® V1.1, の一部として購入した場合は、製品コード 5697-CDT の項目を次のように変更します。

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

Fault Analyzer を製品コード 5755-A02 の一部として購入した場合は、製品コード 5755-A02 の項目を次のように変更します。

```
PRODUCT OWNER('IBM CORP')
  NAME('FAULT ANALYZER')
  ID(5755-A02)
```

```
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('FAULT_ANALYZER')
STATE(DISABLED)
```

VERSION、RELEASE および MOD を除くすべてのパラメーターは、示されている通り正しく指定する必要があります。構文上有効な値を VERSION、RELEASE および MOD に指定できます。

IFAPRDxx parmlib メンバーに関する一般情報については、『MVS™ 初期設定およびチューニング解説書』を参照してください。

後の段階で、Fault Analyzer を再度使用可能にすることが必要になった場合には、上記の項目を削除するか、または STATE を ENABLED に変更します。

## JCL スイッチ (Fault Analyzer) を使用した IDIOFF の停止

Fault Analyzer 呼び出し出口は、ジョブ・ステップ内で次の JCL ステートメントをコーディングすることにより、ジョブ・ステップ・レベルでオフにすることができます。

```
//IDIOFF DD DUMMY
```

JCL スイッチの使用は、

```
//IDIOPTS DD *
  Exclude
/*
```

をコーディングするよりも効率的です。これは、JCL スイッチが Exclude オプションよりもかなり前に Fault Analyzer により処理されるためです (詳しくは、[リアルタイム除外処理 ページ 54](#)を参照してください)。

## 環境変数 (Fault Analyzer) を使用した \_IDI\_OFF の停止

z/OS® UNIX システム・サービス環境では、名前 Language Environment® の環境変数を作成し、文字「Y」を含めることによって、Fault Analyzer の異常終了出口 IDIXCEE による `_IDI_OFF` の呼び出しをオフにすることができます。

C プログラムでのこの環境変数の設定例は、次のとおりです。

```
setenv("_IDI_OFF","Y",1); /* disable IDIXCEE invocation*/
```

Fault Analyzer 出口による IDIXCEE の呼び出しを再度有効にするには、`_IDI_OFF` 環境変数に「Y」以外の値 (例えば、「N」) を設定します。

```
setenv("_IDI_OFF","N",1); /* re-enable IDIXCEE invocation */
```

IDIXCEE 出口使用時の詳細については、[Fault Analyzer を呼び出すための出口 ページ 299](#)を参照してください。`_IDI_OFF` 環境変数は、他の出口による Fault Analyzer の呼び出しに影響しません。

## 第 30 章. ユーザー出口を使用した Fault Analyzer のカスタマイズ

Fault Analyzer をより柔軟に使用可能にするために、Fault Analyzer 操作中にユーザー出口が制御を取得できる場所に、ユーザー出口点のセットが作成されました。ユーザー出口は REXX、アセンブラ、または高水準言語で作成できます。通常これらには 2 つのデータ構造が渡されます。1 つは、現在処理されている障害についての一般情報フィールドを提供する、すべてのユーザー出口に渡される共通環境構造です。もう 1 つの構造は、通常、呼び出されている特定の出口のフィールドです。フィールドには、出口に情報を渡すために使用されるものと、ユーザー出口が Fault Analyzer にデータまたは必須アクションを渡すためのものがあります。REXX で作成された出口の場合、データは構造ではなくステム変数で渡されます。その方法のほうが REXX で処理しやすいからです。



**注:** REXX は、フォーマット・ユーザー出口でサポートされる唯一のプログラミング言語です。

出口を使用して、履歴・ファイルのデータ・セットまたはコンパイル・リストのデータ・セットの動的な選択などの機能を実行できます。また、障害が発生したことをメッセージまたは E メールによってユーザーに通知するために出口を使用することもできます。出口には、この他にも多くの使用法があります。



**注:** Fault Analyzer リカバリー障害記録 (RFR) が実行される条件が生じた場合、この障害分析中にユーザー出口は呼び出されません。

ユーザー出口で行われるオプション設定や選択は、現行の分析にのみ効果があります。

Fault Analyzer のユーザーには、次のユーザー出口が使用可能です。

- [分析制御ユーザー出口 ページ 475](#)
- [分析制御ユーザー出口 \(MVS SVC ダンプ登録\) ページ 479](#)
- [コンパイラ・リスト読み取りユーザー出口 ページ 480](#)
- [メッセージおよび異常終了コード説明ユーザー出口 ページ 485](#)
- [フォーマット・ユーザー出口 ページ 491](#)
- [終了処理ユーザー出口 ページ 494](#)
- [終了処理ユーザー出口 \(障害項目リフレッシュ\) ページ 498](#)
- [通知ユーザー出口 ページ 499](#)
- [通知ユーザー出口 \(MVS SVC ダンプ登録\) ページ 507](#)
- [IDIUTIL インポート・ユーザー出口 ページ 508 \(IDIUTIL バッチ・ユーティリティーでのみ使用されます\)](#)
- [IDIUTIL 削除ユーザー出口 ページ 511 \(IDIUTIL バッチ・ユーティリティーでのみ使用されます\)](#)
- [IDIUTIL ListHF ユーザー出口 ページ 512 \(IDIUTIL バッチ・ユーティリティーでのみ使用されます\)](#)

図 223 : Fault Analyzer 分析の出口点 (IDIDA) ページ 465 リアルタイム分析、バッチ再分析、および対話式再分析に提供される出口点を示しています。図 224 : IDIUTIL バッチ・ユーティリティー出口点 ページ 466 では、IDIUTIL バッチ・ユーティリティーに提供される出口点を示しています。



図 223. Fault Analyzer 分析の出口点 (IDIDA)

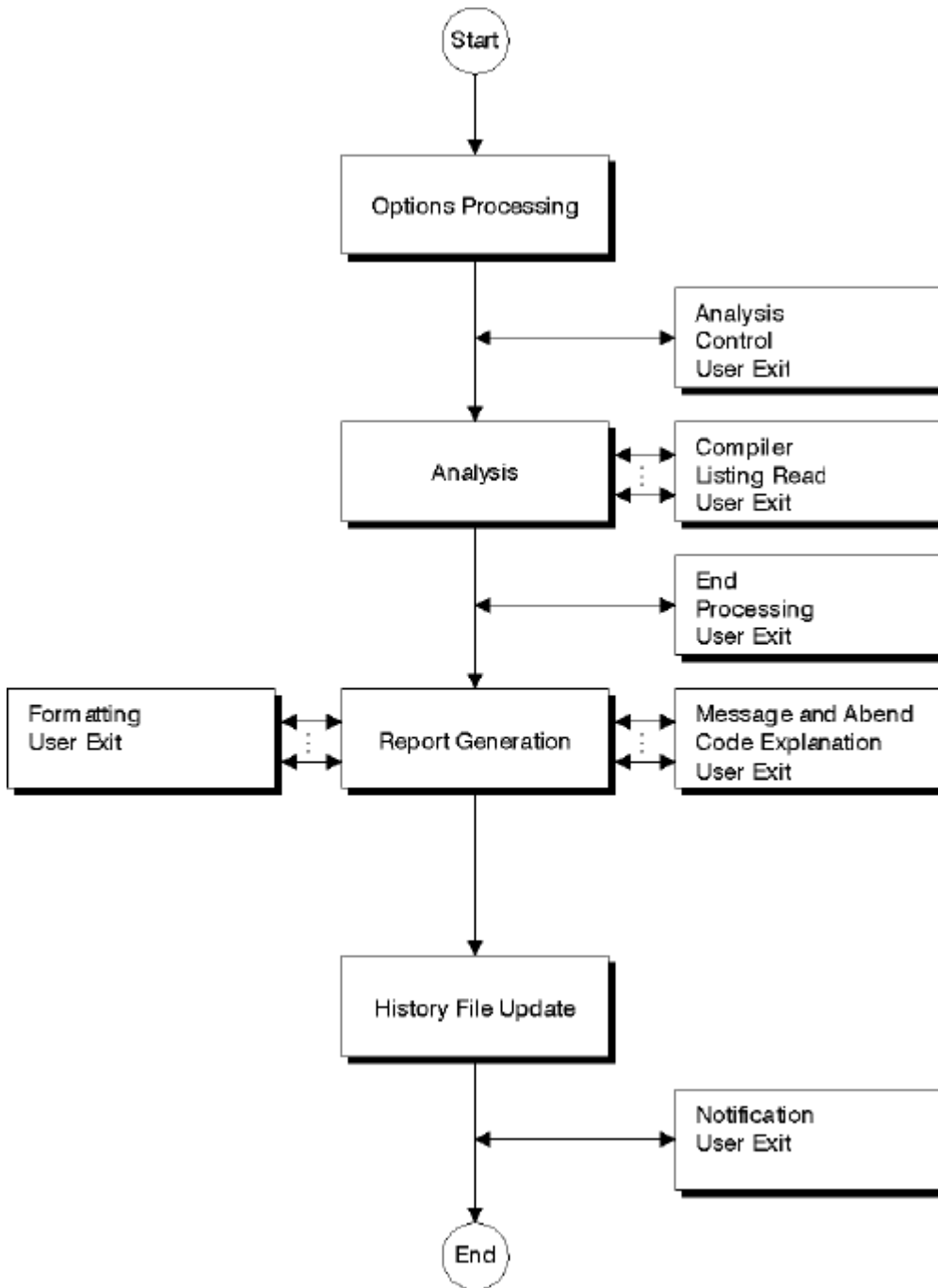
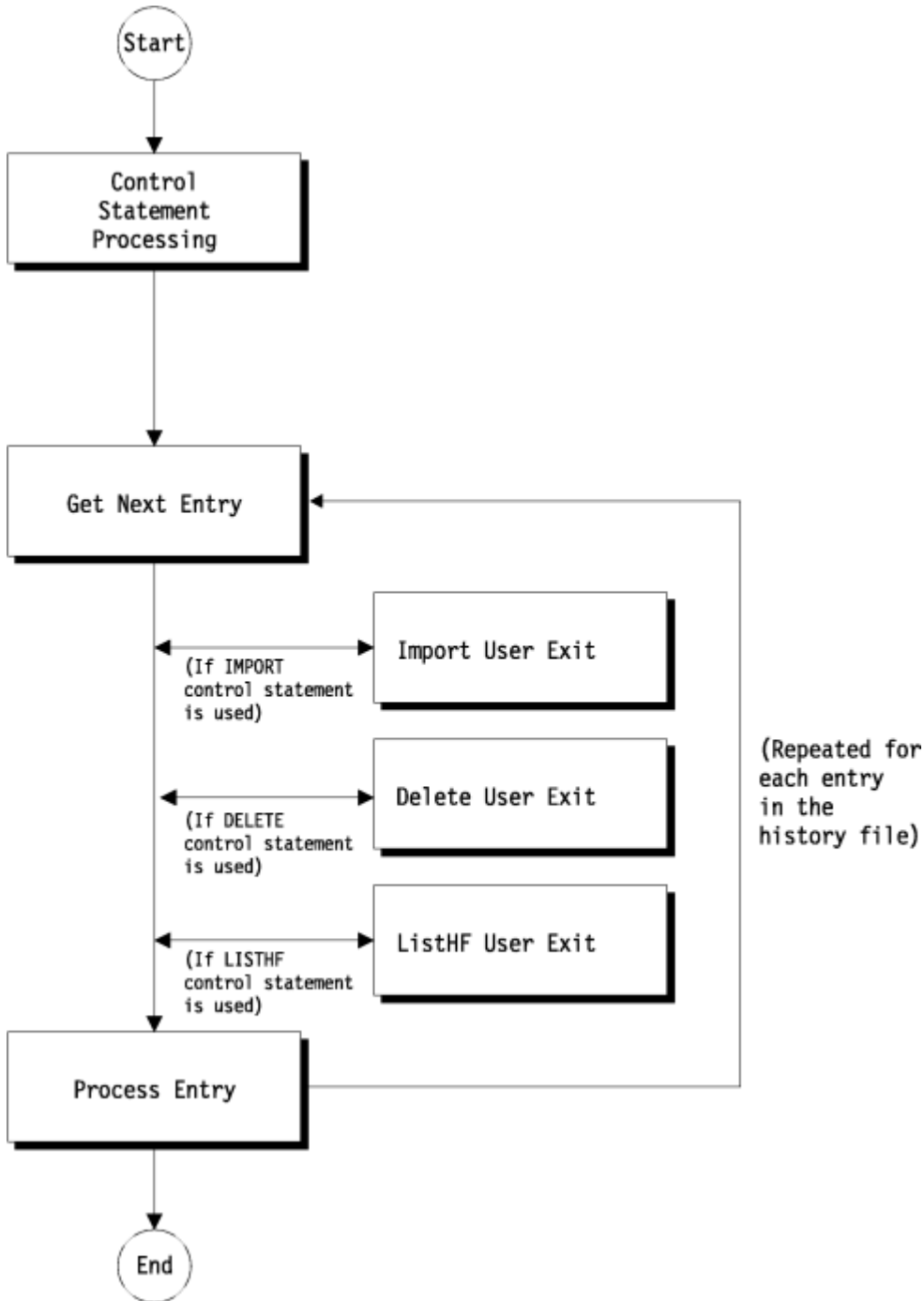


図 224. IDIUTIL バッチ・ユーティリティー出口点



ユーザー出口は、Exits オプション ([Exits ページ 595](#)を参照)、DumpRegistrationExits オプション ([DumpRegistrationExits ページ 588](#)を参照)、または RefreshExits オプション ([RefreshExits ページ 625](#)を参照) を介して Fault Analyzer に指定されます。

IDIUTIL インポート、IDIUTIL ListHF、IDIUTIL 削除の各ユーザー出口は、IDIUTIL バッチ・ユーティリティーでのみ機能します。これらは、Exits 制御ステートメントを使用します。詳細については、[EXITS 制御ステートメント ページ 450](#)を参照してください。

## 起動パラメーター

ユーザー出口には必ず 2 つのデータ域が渡されます。

- グローバル環境データ域 (ENV)。
- ユーザー出口のタイプに固有のデータ域。

## グローバル環境データ域 (ENV)

ENV データ域は、すべての出口タイプに共通の情報を提供します。

このデータ域には 2 つの特殊フィールドがあります: USER\_1 および USER\_2。まず Fault Analyzer はこれらのフィールドをブランクに初期化しますが、最初のユーザー出口の呼び出しと最後のユーザー出口の呼び出しの間に再初期化は行われません。これらのフィールドを使用して、情報 (データ域のアドレスなど) を出口間で渡すことができます。

ENV データ域の詳細については、[ENV - 共通出口環境情報 ページ 650](#)を参照してください。

## ユーザー出口タイプに固有のデータ域

出口タイプに固有のデータ域については、[障害分析ユーザー出口タイプの説明 ページ 474](#)または [IDIUTIL バッチ・ユーティリティー・ユーザー出口タイプの説明 ページ 508](#)の個々の出口タイプを参照してください。

## サポートされる出口プログラム言語

ユーザー出口は、REXX、または標準 OS パラメーター・リスト (フルワードの配列を指す R1) をロード・モジュールへのエントリーで受信できる、任意の言語で作成できます。

### REXX ユーザー出口

REXX ユーザー出口は IDIEXEC DDname 経由で使用可能である必要があります。この ddname のデータ・セットは、DataSets オプションに指定するか ([DataSets ページ 576](#) を参照)、またはジョブの JCL DD ステートメントから指定できます。

ユーザー出口から戻るときに、変更可能パラメーター・リスト値は定義されたフィールド幅に合わせて必ず切り捨てられます。これより短い値の場合は、ヌル文字 (X'00') が追加されて区切られます。

診断トレース (詳細については、[診断トレース ページ 469](#)を参照) がアクティブでない限り、REXX メッセージなど REXX EXEC からの SYSPRINT タイプ出力、および SAY または TRACE 命令からの出力は、Fault Analyzer により抑止されます。診断トレースがアクティブかどうかにかかわらず REXX EXEC がメッセージを示すことを要求する場合、代わりに IDIWTO コマンド (詳細については、[IDIWTO コマンド ページ 537](#)を参照) を発行してください。

新規の REXX 出口を作成する場合、別の方法では外部から参照できない問題の検出に IDITRACE を使用することをお勧めします。

REXX ユーザー出口は TSO/E REXX 環境を使用できないため、「Address TSO」コマンドの使用はサポートされていません。何がサポートされているかについては、「z/OS@TSO/E REXX 解説書」の『異なるアドレス・スペースでの REXX の使用』の章の『非 TSO/E アドレス・スペースで実行される EXEC の作成』のセクションを参照してください。

## ロード・モジュール出口

ロード・モジュール出口は、標準 MVS™ 検索パス経由で使用可能でなければなりません。また、IDIDA サブタスクにより既存の LE エンクレーブ下で実行されるため、LE メインルーチン (C `main()` 関数または PL/I `PROC OPTIONS(MAIN)`) を含むことはできません。

ロード・モジュール出口を作成しやすくするために、Fault Analyzer はサンプル・データ・セットで以下のパラメーター・リスト・データ域マッピング・メンバーを提供します。

### 名前

#### 言語

#### IDISXPLA

アセンブラー

#### IDISXPLC

C

#### IDISXPLB

COBOL



**注:** COBOL 言語の制限により、パラメーター・リスト・フィールド名内のすべての下線 (.) が ダッシュ (-) に置換されています。

#### IDISXPLP

PL/I

ロード・モジュール出口は再入可能または再入不可のどちらにすることもでき、RMODE(ANY) としてリンク・エディットされている必要があります。これらは AMODE(31) で呼び出されます。

## 共通のユーザー出口を呼び出している出口タイプの判別

ユーザー出口は、Fault Analyzer オプション (Exits または DumpRegistrationExits) や IDIUTIL Exits 制御ステートメントによって呼び出すことができます。Fault Analyzer オプションと IDIUTIL 制御ステートメントでは、どのユーザー出口がどのような条件の下で呼び出されるのかを決定します:

- Exits オプション (Control、Listing、Msgxpl、Format、End、Notify)
- DumpRegistrationExits オプション (Control、Notify)
- IDIUTIL Exits 制御ステートメント (ListHF、Delete、Import)

ユーザー出口タイプごとに異なる REXX ユーザー出口またはロード・モジュール・ユーザー出口を使用したり、さまざまなユーザー出口タイプで同一の REXX ユーザー出口またはロード・モジュール・ユーザー出口を使用したりすることが可能です。

さまざまなユーザー出口タイプで同一の REXX ユーザー出口またはロード・モジュール・ユーザー出口を使用する場合、当該のユーザー出口がどの出口タイプに呼び出されているのかを確認することができます。

例えば、単一の REXX ユーザー出口は、通常の分析制御ユーザー出口、終了処理ユーザー出口、オプションを使用したダンプ登録分析制御ユーザー出口として指定されます：

```
Exits(Control(REXX(myexit)),End(REXX(myexit)))
DumpRegistrationExits(Control(REXX(myexit)))
```

この場合、ユーザー出口がどの出口タイプに呼び出されているのかを確認する際に使用するコードには、次が含まれます：

```
If ENV.EXIT_CALL_TYPE = 'C' then do
  "IDIWTO 'Normal Analysis Control user exit called'"
  /* Perform normal Analysis Control user exit processing... */
else if ENV.EXIT_CALL_TYPE = 'E' then do
  "IDIWTO 'End Processing user exit called'"
  /* Perform End Processing user exit processing... */
else if ENV.EXIT_CALL_TYPE = 'X' then do
  "IDIWTO 'Dump registration Analysis Control user exit called'"
  /* Perform dump registration Analysis Control user exit processing... */
else do
  "IDIWTO 'User exit unexpectedly called'"
end
```

## データ域のバージョンの検査

すべてのユーザー出口に、期待されるバージョンのデータ域が使用されているかどうかの検査を組み込むことをお勧めします。この検査は、データ域の変更の結果、旧バージョンとの非互換を起こすような誤った処理が発生しないようにするためのものです。

出口を作成したときの現行値である VERSION フィールドを含むすべてのユーザー出口データ域は、[データ域 ページ 635](#)のデータ域の説明から取得することができます。

これらの検査の例は、示されたすべてのサンプル出口に提供されています。

## 診断トレース

Exits オプションで呼び出されたユーザー出口のデバッグ情報を利用するには、JCL に IDITRACE DDname を追加します。

例:

```
//IDITRACE DD SYSOUT=*
```

(CICS® 環境でこのトレースを有効にする別の方法については、[IDITRACE CICS](#) において [ページ 407](#)を参照してください。)

## ユーザー出口パラメーター・リスト値のトレース

IDITRACE DDname がジョブ・ステップにより割り振られると、ユーザー出口の呼び出しの前および出口から戻るときに、すべてのパラメーター・リストの内容がこの DDname に書き込まれます。

更新されたパラメーター・リスト・フィールドの妥当性検査中のエラーは、トレース出力の警告メッセージによって示されます。

Exits オプションを使用して指定された出口のすべての出口タイプ、および Fault Analyzer の実行モードで呼び出すことができるすべての出口タイプについて、トレース情報が提供されます。最初に出口を作成せずに、出口の呼び出しの前にフィールドの値を示す情報をトレースするには、特定の出口名「NONE」を使用します。

呼び出されたすべての出口について、秒単位で経過時間が提供されます。

以下は、SAMPCTLX という名前の出口に対する 1 つの分析制御ユーザー出口呼び出しを含む出口トレースの例です。

```

ANALYSIS CONTROL User Exit:
Parameter values prior to exit invocation:
ENV.VERSION . . . . . : 0005
EXIT_CALL_TYPE. . . . . : C
FAULT_ID. . . . . :
ABEND_DATE. . . . . : 2001/03/23
ABEND_TIME. . . . . : 10:02:03
DUP_DATE. . . . . :
DUP_TIME. . . . . :
ORIGINAL_DATE . . . . . :
ORIGINAL_TIME . . . . . :
REALTIME. . . . . : Y
SYSTEM_NAME . . . . . : MVSA
JOB_NAME. . . . . : CI03DA
EXEC_PGM_NAME . . . . . : DFHSIP
USER_ID . . . . . : CICSUSER
GROUP_ID. . . . . : APC
ABEND_CODE. . . . . : AEIL
ABEND_REASON_CODE . . . . . :
INVOCATION_ABEND_CODE . . . . . : AEIL
ABEND_MODULE_NAME . . . . . : CICFRED
CICS_TRANSACTION_ID . . . . . : FRED
CICS_TASK_NUMBER. . . . . : 00026
JOB_TYPE. . . . . : C
JOB_CLASS . . . . . : A
ACCOUNTING_FIELDS . . . . . :
ACCOUNTING_INFO . . . . . :
USER_1. . . . . :
USER_2. . . . . :
LOCK_FLAG . . . . . :
LOCK_USERID . . . . . :
LOOPPROTECTION_OPT. . . . . : Y
WRITE_ROUTINE_EP. . . . . : X'00000000'
INVOCATION_EXIT . . . . . : L
STEP_NAME . . . . . : CICS
JOB_ID. . . . . : JOB30073
IMS_PROGRAM_NAME. . . . . :
USER_NAME . . . . . :
USER_TITLE. . . . . :
APPLID. . . . . : QXPM2C53
NETNAME . . . . . :
TERMID. . . . . : SAMA

```

```

TCB_ADDRESS . . . . . :
CSA_ADDRESS . . . . . :
TCA_ADDRESS . . . . . :
IDIHIST . . . . . : FRED.DCAT
CPU_HSECONDS . . . . . :
CICS_VRM . . . . . :
DB2_VRM . . . . . :
IMS_VRM . . . . . :
ZOS_VRM . . . . . :
DUPLICATE_COUNT . . . . . : 00000
POF_MODULE_NAME . . . . . :
POF_MODULE_LKED_DATE . . . . . :
POF_MODULE_LKED_TIME . . . . . :
POF_CSECT_NAME . . . . . :
POF_CSECT_OFFSET . . . . . : 0000000000
POF_LOADED_FROM . . . . . :
EXEC_LOADED_FROM . . . . . :
MINIDUMP_PAGES . . . . . : 0000000000
CTL.VERSION . . . . . : 0002
Exclude . . . . . :
DETAIL_OPT . . . . . : M
DEFERREDREPORT_OPT . . . . . : N
RETAINDUMP_OPT . . . . . : AUTO
SOURCE_OPT . . . . . : Y
IDIADATA_PRE . . . . . :
IDIADATA_JOB . . . . . :
IDIADATA_CFG . . . . . : FRED.SYSADATA
IDILC_PRE . . . . . :
IDILC_JOB . . . . . :
IDILC_CFG . . . . . : FRED.LISTING.C
IDILCOB_PRE . . . . . :
IDILCOB_JOB . . . . . :
IDILCOB_CFG . . . . . : FRED.LISTING.COBOL
IDILCOBO_PRE . . . . . :
IDILCOBO_JOB . . . . . :
IDILCOBO_CFG . . . . . :
IDILANGX_PRE . . . . . :
IDILANGX_JOB . . . . . :
IDILANGX_CFG . . . . . : FRED.WDBLANGX
IDILPLI_PRE . . . . . :
IDILPLI_JOB . . . . . : USERA.JCLLIB          USERA.TEXT
                                USERA.LOAD
IDILPLI_CFG . . . . . : FRED.LISTING.PLI
IDILPLIE_PRE . . . . . :
IDILPLIE_JOB . . . . . :
IDILPLIE_CFG . . . . . :
LOCALE . . . . . : S370
FADATE . . . . . : N
IDISYSDB_PRE . . . . . :
IDISYSDB_JOB . . . . . :
IDISYSDB_CFG . . . . . :
IDIJAVA_PRE . . . . . :
IDIJAVA_JOB . . . . . :
IDIJAVA_CFG . . . . . :
STEPLIB . . . . . :
Parameter values after return from exit SAMPCTLX (elapsed 0.03 seconds):
ENV.VERSION . . . . . : 0005
EXIT_CALL_TYPE . . . . . : C
FAULT_ID . . . . . :
ABEND_DATE . . . . . : 2001/03/23
ABEND_TIME . . . . . : 10:02:03
DUP_DATE . . . . . :
DUP_TIME . . . . . :
ORIGINAL_DATE . . . . . :
ORIGINAL_TIME . . . . . :
REALTIME . . . . . : Y

```

```

SYSTEM_NAME . . . . . : MVSA
JOB_NAME. . . . . : CI03DA
EXEC_PGM_NAME . . . . . : DFHSIP
USER_ID . . . . . : CICSUSER
GROUP_ID. . . . . : APC
ABEND_CODE. . . . . : AEIL
ABEND_REASON_CODE . . . . . :
INVOCATION_ABEND_CODE . . . . . : AEIL
ABEND_MODULE_NAME . . . . . : CICFRED
CICS_TRANSACTION_ID . . . . . : FRED
CICS_TASK_NUMBER. . . . . : 00026
JOB_TYPE. . . . . : C
JOB_CLASS . . . . . : A
ACCOUNTING_FIELDS . . . . . :
ACCOUNTING_INFO . . . . . :
USER_1. . . . . : ABCD
USER_2. . . . . : 0123
LOCK_FLAG . . . . . :
LOCK_USERID . . . . . :
LOOPPROTECTION_OPT. . . . . : Y
WRITE_ROUTINE_EP. . . . . : X'00000000'
INVOCATION_EXIT . . . . . : L
STEP_NAME . . . . . : CICS
JOB_ID. . . . . : JOB30073
IMS_PROGRAM_NAME. . . . . :
USER_NAME . . . . . :
USER_TITLE. . . . . :
APPLID. . . . . : QXPM2C53
NETNAME . . . . . :
TERMID. . . . . : SAMA
TCB_ADDRESS . . . . . :
CSA_ADDRESS . . . . . :
TCA_ADDRESS . . . . . :
IDIHIST . . . . . : USERA.DCAT
CPU_HSECONDS. . . . . :
CICS_VRM. . . . . :
DB2_VRM . . . . . :
IMS_VRM . . . . . :
ZOS_VRM . . . . . :
DUPLICATE_COUNT . . . . . : 00000
POF_MODULE_NAME . . . . . :
POF_MODULE_LKED_DATE. . . . . :
POF_MODULE_LKED_TIME. . . . . :
POF_CSECT_NAME. . . . . :
POF_CSECT_OFFSET. . . . . : 0000000000
POF_LOADED_FROM . . . . . :
EXEC_LOADED_FROM. . . . . :
MINIDUMP_PAGES. . . . . : 0000000000
CTL.VERSION . . . . . : 0002
Exclude . . . . . :
DETAIL_OPT. . . . . : S
DEFERREDREPORT_OPT. . . . . : N
RETAINDUMP_OPT. . . . . : AUTO
SOURCE_OPT. . . . . : Y
IDIADATA_PRE. . . . . :
IDIADATA_JOB. . . . . :
IDIADATA_CFG. . . . . : FRED.SYSADATA
IDILC_PRE . . . . . :
IDILC_JOB . . . . . :
IDILC_CFG . . . . . : FRED.LISTING.C
IDILCOB_PRE . . . . . :
IDILCOB_JOB . . . . . :
IDILCOB_CFG . . . . . : FRED.LISTING.COBOL
IDILCOBO_PRE. . . . . :
IDILCOBO_JOB. . . . . :
IDILCOBO_CFG. . . . . :

```



```

IDILANGX_PRE. . . . . :
IDILANGX_JOB. . . . . :
IDILANGX_CFG. . . . . : FRED.WDBLANGX
IDILPLI_PRE . . . . . :
IDILPLI_JOB . . . . . :
IDILPLI_CFG . . . . . : FRED.LISTING.PLI
IDILPLIE_PRE. . . . . :
IDILPLIE_JOB. . . . . : *** Data from 9945 byte buffer at address 17FF08F8 follows:
                                FRED.IDILPLIE.T001                FRED.IDILPLIE.T002
                                FRED.IDILPLIE.T003                FRED.IDILPLIE.T004
                                FRED.IDILPLIE.T005                FRED.IDILPLIE.T006
                                FRED.IDILPLIE.T007                FRED.IDILPLIE.T008
                                FRED.IDILPLIE.T009                FRED.IDILPLIE.T010
                                FRED.IDILPLIE.T011                FRED.IDILPLIE.T012
                                FRED.IDILPLIE.T013                FRED.IDILPLIE.T014
                                FRED.IDILPLIE.T015                FRED.IDILPLIE.T016
                                FRED.IDILPLIE.T017                FRED.IDILPLIE.T018
                                FRED.IDILPLIE.T019                FRED.IDILPLIE.T020
                                FRED.IDILPLIE.T021                FRED.IDILPLIE.T022
                                FRED.IDILPLIE.T023                FRED.IDILPLIE.T024
                                FRED.IDILPLIE.T025                FRED.IDILPLIE.T026
                                FRED.IDILPLIE.T027                FRED.IDILPLIE.T028
                                FRED.IDILPLIE.T029                FRED.IDILPLIE.T030
                                FRED.IDILPLIE.T031                FRED.IDILPLIE.T032
                                FRED.IDILPLIE.T033                FRED.IDILPLIE.T034
                                FRED.IDILPLIE.T035                FRED.IDILPLIE.T036
                                FRED.IDILPLIE.T037                FRED.IDILPLIE.T038
                                FRED.IDILPLIE.T039                FRED.IDILPLIE.T040
                                FRED.IDILPLIE.T041                FRED.IDILPLIE.T042
                                FRED.IDILPLIE.T043                FRED.IDILPLIE.T044
                                FRED.IDILPLIE.T045                FRED.IDILPLIE.T046
                                FRED.IDILPLIE.T047                FRED.IDILPLIE.T048
                                FRED.IDILPLIE.T049                FRED.IDILPLIE.T050
                                FRED.IDILPLIE.T051                FRED.IDILPLIE.T052
                                FRED.IDILPLIE.T053                FRED.IDILPLIE.T054
                                FRED.IDILPLIE.T055                FRED.IDILPLIE.T056
                                FRED.IDILPLIE.T057                FRED.IDILPLIE.T058
                                FRED.IDILPLIE.T059                FRED.IDILPLIE.T060
                                FRED.IDILPLIE.T061                FRED.IDILPLIE.T062
                                FRED.IDILPLIE.T063                FRED.IDILPLIE.T064
                                FRED.IDILPLIE.T065                FRED.IDILPLIE.T066
                                FRED.IDILPLIE.T067                FRED.IDILPLIE.T068
                                FRED.IDILPLIE.T069                FRED.IDILPLIE.T070
                                FRED.IDILPLIE.T071                FRED.IDILPLIE.T072
                                FRED.IDILPLIE.T073                FRED.IDILPLIE.T074
                                FRED.IDILPLIE.T075                FRED.IDILPLIE.T076
                                FRED.IDILPLIE.T077                FRED.IDILPLIE.T078
                                FRED.IDILPLIE.T079                FRED.IDILPLIE.T080
                                FRED.IDILPLIE.T081                FRED.IDILPLIE.T082
                                FRED.IDILPLIE.T083                FRED.IDILPLIE.T084
                                FRED.IDILPLIE.T085                FRED.IDILPLIE.T086
                                FRED.IDILPLIE.T087                FRED.IDILPLIE.T088
                                FRED.IDILPLIE.T089                FRED.IDILPLIE.T090
                                FRED.IDILPLIE.T091                FRED.IDILPLIE.T092
                                FRED.IDILPLIE.T095                FRED.IDILPLIE.T096
                                FRED.IDILPLIE.T097                FRED.IDILPLIE.T098
                                FRED.IDILPLIE.T099                FRED.IDILPLIE.T100
                                FRED.IDILPLIE.T101                FRED.IDILPLIE.T102
                                FRED.IDILPLIE.T103                FRED.IDILPLIE.T104
                                FRED.IDILPLIE.T105                FRED.IDILPLIE.T106
                                FRED.IDILPLIE.T107                FRED.IDILPLIE.T108
                                FRED.IDILPLIE.T109                FRED.IDILPLIE.T110
                                FRED.IDILPLIE.T111                FRED.IDILPLIE.T112
                                FRED.IDILPLIE.T113                FRED.IDILPLIE.T114
                                FRED.IDILPLIE.T115                FRED.IDILPLIE.T116
                                FRED.IDILPLIE.T117                FRED.IDILPLIE.T118

```

```

                                FRED.IDILPLIE.T119
                                FRED.IDILPLIE.T121
                                FRED.IDILPLIE.T120
                                FRED.IDILPLIE.T121

IDILPLIE_CFG. . . . . :
LOCALE. . . . . : S370
FADATE. . . . . : N
IDISYSDB_PRE. . . . . :
IDISYSDB_JOB. . . . . :
IDISYSDB_CFG. . . . . :
IDIJAVA_PRE . . . . . :
IDIJAVA_JOB . . . . . : /u/temp/payroll/directory171/DEPT64directory/accountingDIR1/very/long/path/name/
                                that/wraps/to/the/next/line
                                '/u/temp/payroll/directory171/DEPT64directory/accountingDIR1''temp'

IDIJAVA_CFG . . . . . :
STEPLIB . . . . . :
    
```

## REXX EXEC のトレース

ユーザー出口を REXX EXEC として作成する場合、SAY または TRACE 命令を使用して作成された情報は Fault Analyzer データ・セット内の IDITRACE 出口パラメーター・リスト・トレース・レコードにマージされます。

## 障害分析ユーザー出口タイプの説明

リアルタイムまたは障害再分析の実行モードで、Fault Analyzer で使用可能な各ユーザー出口タイプについて、以下に説明します。表 13: 実行モードに適用可能な障害分析ユーザー出口タイプ ページ 474 各実行モードに適用可能な出口タイプを示しています。

表 13. 実行モードに適用可能な障害分析ユーザー出口タイプ

ユーザー出口タイプ	リアルタイム		再分析		
	通常	ダンプの登録	バッチ		対話式
			通常	更新	
分析制御	可 (1)	可 (2)	可 (1)	可 (1)	可 (1)
Compiler Listing Read	可 (1)	なし	可 (1)	可 (1)	可 (1)
Message and Abend Code Explanation	可 (1)	なし	可 (1)	可 (1)	可 (1)
フォーマット	可 (1)	なし	可 (1)	可 (1)	可 (1)
End Processing	可 (1)	なし	なし	可 (3)	なし
通知	可 (1)	可 (2)	なし	なし	なし

注:

(1)

Exits オプションにより指定

(2)

DumpRegistrationExits オプションにより指定

(3)

RefreshExits オプションにより指定

## 分析制御ユーザー出口

分析制御ユーザー出口について、以下で説明します。

### 目的

この出口を使用して、以下のオプションの現行設定を確認し、オーバーライドできます。

#### DataSets

以下の ddname のデータ・セットが指定されます。

- IDIADATA
- IDIHIST
- IDILANGX
- IDILC
- IDILCOB
- IDILCOBO
- IDILPLI
- IDILPLIE
- IDISYSDB

IDIADATA、IDILANGX、IDILC、IDILCOB、IDILCOBO、IDILPLI、IDILPLIE、および IDISYSDB DDname の場合、現行データ・セットが以下のように指定されます。

#### 事前割り振りデータ・セット

これは、JCL DD ステートメントなどから Fault Analyzer を起動する前に割り振られたデータ・セットです。各 DDname のデータ・セットは、CTL.ddname\_PRE フィールドに提供されます。

事前割り振りデータ・セット名フィールドは読み取り専用で、変更は Fault Analyzer により無視されます。

#### ジョブ割り振りデータ・セット

これは、ユーザー・オプション・ファイル (IDIOPTS) からの DataSets オプション指定です。各 DDname のデータ・セットは、CTL.ddname\_JOB フィールドに提供されます。

ジョブ割り振りデータ・セット名フィールドへの変更は、Fault Analyzer がデータ・セットのこのリストを割り振るときに受け入れられます。

## 構成データ・セット

これは、IDICNFxx 構成メンバーからの DataSets オプション指定です。各 DDname のデータ・セットは、CTL.ddname\_CFG フィールドに提供されます。

構成データ・セット名フィールドへの変更は Fault Analyzer により受け入れられます。

データ・セットの最終連結順序は次のとおりです。

1. 事前割り振りデータ・セット
2. ジョブ割り振りデータ・セット
3. 構成データ・セット

IDIHIST DDname の場合、現行ヒストリー・ファイルは ENV.IDIHIST データ域フィールドに指定されます。ユーザー出口はこのデータ・セット名を変更することを選択できます。この場合、指定されたデータ・セット名は現行の障害のヒストリー・ファイルとして使用されます。

DataSets オプション ([DataSets オプションのデータ・セット名の置換シンボル ページ 580](#)を参照) に適用されるデータ・セット名内の置換シンボルの利用に関する同じ規則が、分析制御ユーザー出口により返されるデータ・セット名にも適用されます。

## DeferredReport

DeferredReport オプションが有効かどうかの状態は、「Y」(DeferredReport が有効) または「N」(DeferredReport が有効でない) のいずれで CTL.DEFERREDREPORT\_OPT データ域フィールドに示されます。このフィールドへの変更が有効な場合、現行オプション設定がオーバーライドされます。

リアルタイム処理にのみ適用できます。

## Detail

有効な Detail オプションの省略フォームは、CTL.DETAIL\_OPT データ域フィールドに指定されます。このフィールドへの変更が有効な場合、現行オプション設定がオーバーライドされます。

対話式再分析には適用されません。

## Exclude

最後に一致する Exclude 基準は、CTL.EXCLUDE\_CRITERION データ域フィールドに指定されます。一致する Exclude 基準に基づき、障害が分析から除外された場合、CTL.EXCLUDE データ域フィールドは「Y」に初期化されます。このフィールドは出口により修正可能です。

リアルタイム処理にのみ適用できます。

## Include

最後に一致する Include 基準は、CTL.INCLUDE\_CRITERION データ域フィールドに指定されます。ブランクの Include 基準は、すべてを組み込むことを示します。これは、暗黙的製品デフォルトです。

リアルタイム処理にのみ適用できます。

## Locale

現行のロケール名は、CTL.LOCALE データ域フィールドに指定されます。このフィールドへの変更が有効な場合、現行オプション設定がオーバーライドされます。

LOCALE オプションのサブオプション FADATE は、CTL.FADATE

データ域フィールドに指定されます。このフィールドへの変更が有効な場合、現行オプション設定がオーバーライドされます。

## RetainDump

有効な RetainDump オプションの値は、CTL.RETAINDUMP\_OPT

データ域フィールドに指定されます (このフィールドに指定される値は、RetainDump

オプションの実際のサブオプションである「AUTO」または「ALL」です)。このフィールドへの変更が有効な場合、現行オプション設定がオーバーライドされます。

このオプションは、後ほど、End Processing ユーザー出口 ([終了処理ユーザー出口 ページ 494](#)を参照) によってオーバーライドされることがあります。

リアルタイム処理にのみ適用できます。

## Source

有効な Source オプションの状態は、「Y」(Source が有効) または「N」(Source が有効でない)

のいずれかで、CTL.SOURCE\_OPT

データ域フィールドに示されます。このフィールドへの変更が有効な場合、現行オプション設定がオーバーライドされます。

リアルタイム処理にのみ適用できます。

さらに、分析制御ユーザー出口は、オプションにより指定されない、または渡されたデータ域で指定される可能性があるその他のデータ・セットの割り振りを実行できます。リアルタイムでは、このような割り振りは IDIREPRT で実行可能です。これにより、インストール環境で SYSOUT クラスなどのレポート宛先属性を制御できます。このタイプの IDIREPRT 割り振りについて、詳細は [Fault Analyzer リアルタイム・レポートの結合 ページ 37](#)、[リアルタイム・レポートの SYSOUT クラスを制御 ページ 37](#)、および [リアルタイム・レポートの抑止 ページ 38](#) を参照してください。

**CTL.IDITRACE data area** フィールドを使用して、IDITRACE 処理を動的に開始または停止することができ、またトレースの出力先をデータ・セットにすることができます。

## 呼び出されるタイミング

この出口は、オプション処理が完了した後で、障害分析の開始前に呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、分析制御ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

## REXX

出口は、以下の2つのシステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- CTL.

CTL データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([CTL - 分析制御ユーザー出口パラメーター・リスト ページ 636](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の2つのフルワードを構成するパラメーター・リストの31ビット・アドレスが含まれます。

- word 1 の31ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の31ビット CTL アドレス。

CTL データ域のアドレス ([CTL - 分析制御ユーザー出口パラメーター・リスト ページ 636](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例

以下は、REXX で書かれた分析制御ユーザー出口の例です。

## 図 225. サンプル REXX 分析制御ユーザー出口

```

/* REXX */
/* Check data areas used */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if CTL.VERSION <> 2 then
  say 'Note: CTL data area version change - field usage review required!'
if ENV.REALTIME = 'Y' then do /* Exclude all MVSA jobs from analysis */
  if ENV.SYSTEM_NAME = 'MVSA' then
    CTL.Exclude = 'Y'
  /* Select a separate history file for DB2, IMS, and other jobs
  based on jobname */
  if SUBSTR(ENV.JOB_NAME,1,3) = 'DB2' then
    ENV.IDIHIST = 'MY.DB2.HIST'
  else if SUBSTR(ENV.JOB_NAME,1,3) = 'IMS' then
    ENV.IDIHIST = 'MY.IMS.HIST'
  else
    ENV.IDIHIST = 'MY.OTHER.HIST'
end
exit 0

```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```

DataSets(IDIEXEC(X.Y.Z))
Exits(CONTROL(REXX(ABC)))

```

## 分析制御ユーザー出口 (MVS SVC ダンプ登録)

ダンプ登録分析制御ユーザー出口について、以下で説明します。

### 目的

この出口を使用して、ダンプ登録処理に関連する以下のオプションの現行設定を確認し、オーバーライドできます。

#### DataSets

ヒストリー・ファイルのデータ・セット名のみ、この出口に関連します。現行のヒストリー・ファイル・データ・セット名は、ENV.IDIHIST データ域フィールドに指定されます。ユーザー出口はこのデータ・セット名を変更することを選択できます。この場合、指定されたデータ・セット名は現行の障害のヒストリー・ファイルとして使用されます。ヒストリー・ファイルが事前割り振りされていた場合は解放されます。

#### Exclude

最後に一致する Exclude 基準は、CTL.EXCLUDE\_CRITERION データ域フィールドに指定されます。一致する Exclude 基準に基づき、障害が分析から除外された場合、CTL.EXCLUDE データ域フィールドは「Y」に初期化されます。このフィールドは出口により修正可能です。

#### Include

最後に一致する Include 基準は、CTL.INCLUDE\_CRITERION データ域フィールドに指定されます。ブランクの Include 基準は、すべてを組み込むことを示します。これは、暗黙的製品デフォルトです。

## 呼び出されるタイミング

この出口は、オプション処理が完了した後で、MVS™ SVC ダンプ登録障害項目の書き込み前に呼び出されます。

## パラメーター

[パラメーター ページ 477](#)を参照してください。

## 例

以下は、REXX で書かれた分析制御ダンプ登録ユーザー出口の例です。

図 226. サンプル REXX 分析制御ダンプ登録ユーザー出口

```
/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if CTL.VERSION <> 2 then
  say 'Note: CTL data area version change - field usage review required!'
/* Exclude all MVSA jobs from analysis */
if ENV.SYSTEM_NAME = 'MVSA' then
  CTL.Exclude = 'Y'
/* Select a separate history file for DB2, IMS, and other jobs
based on jobname */
if SUBSTR(ENV.JOB_NAME,1,3) = 'DB2' then
  ENV.IDIHIST = 'MY.DB2.HIST'
else if SUBSTR(ENV.JOB_NAME,1,3) = 'IMS' then
  ENV.IDIHIST = 'MY.IMS.HIST'
else
  ENV.IDIHIST = 'MY.OTHER.HIST'
exit 0
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合は、IDICNFxx 構成メンバーまたは IDIOPTS サブシステムに割り振られる IDIS ユーザー・オプション・ファイルで次のオプションを指定すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))
DumpRegistrationExits(CONTROL(REXX(ABC)))
```

DumpRegistrationExits オプションを、IDICNFxx parmlib メンバーに指定するか、または IDIOPTS DD ステートメントを介して IDIS サブシステム JCL に指定する必要があります。DumpRegistrationExits オプションは、IDIOPTS DD ステートメントを介してどこか他の場所 (CICS® 領域内やバッチ・ジョブ内など) に指定された場合には無視されます。

## コンパイラー・リスト読み取りユーザー出口

コンパイラー・リスト読み取りユーザー出口について、以下で説明します。

## 目的

この出口は、通常の IDI データ・セット (IDILANGX、IDILCOB、IDILPLI など) にメンバーとして保管されているコンパイラー・リストまたは Fault Analyzer サイド・ファイル以外のソースからソース・コード情報を取得する場合に使用できま



す。例えば、圧縮フォーマットで保管されているか、または所有者独自のアクセス方式でのみアクセス可能なコンパイラー・リストなどです。

Fault Analyzer がソース・コード情報を要求する場合、常に、[コンパイラー・リストまたはサイド・ファイルの検索 ページ 384](#)に示すステップが実行されます。Compiler Listing Readユーザー出口の呼び出しは、すべての指定された使用可能な出口の呼び出しを暗黙指定します。すなわち、最初にすべての出口がサイド・ファイルを準備するために呼び出され、必要な場合は、その後でコンパイラー・リストを準備するために、すべての出口が呼び出されます。出口から最後に提供されたサイド・ファイルまたはコンパイラー・リストのみが使用されます。

Fault Analyzer は必須ソース・コード情報について、以下の情報を Compiler Listing Read ユーザー出口に提供します。

- ロード・モジュール名。この名前は LST.MODULE\_NAME データ域フィールドに示されます。LST データ域の参照については、[パラメーター ページ 484](#)を参照してください。
- LST.LOAD\_MODULE\_DSN データ域フィールド内のロード・モジュール・データ・セット名。
- CSECT 名。この名前は LST.CSECT\_NAME データ域フィールドに示されます。
- エントリー・ポイント名。この名前は LST.EP\_NAME データ域フィールドに示され、最大 256 文字で切り捨てられます。切り捨てられる場合、エントリー・ポイント名の最初の 254 文字の後に、波形記号 (~) および切り捨てられていないエントリー・ポイント名の最後の文字が付けられます。
- コンパイル・データ。この日付は LST.COMPILE\_DATE データ域フィールドに YYYY/MM/DD 形式で示されます。
- コンパイル時間。この時間は LST.COMPILE\_TIME データ域フィールドに HH:MM:SS 形式で示されます。
- リスト・タイプ。このタイプは LST.LISTING\_TYPE データ域フィールドに以下のいずれかで示されます。

#### L

コンパイラー・リスト またはアセンブラー SYSADATA ファイル

#### S

Fault Analyzer サイド・ファイル

Fault Analyzer に渡されるソース・コード情報は、このフィールドに指定されたタイプである必要があります。

- 言語タイプ。このタイプは LST.LANGUAGE\_TYPE データ域フィールドに以下のいずれかで示されます。
  - アセンブラー
  - C/C++
  - COBOL
  - OS/VS COBOL
  - PL/I
- 出口により提供されるリストまたはサイド・ファイルの予期されるレコード・フォーマット。このフォーマットは LST.RECFM データ域フィールドに V、VB、VBA、F、FB、FBA などとして示されます。
- 出口により提供されるリストまたはサイド・ファイルの予期される論理レコード長。この長さは、LST.LRECL データ域フィールドに 10 進数文字フォーマットで示されます。

上記の情報に基づいて、ユーザー出口は要求されたリストまたはサイド・ファイルを提供できます。このプロビジョンは、以下のいずれかの方法で行われます。

- LST データ域を介して 1 回に 1 つのレコードを Fault Analyzer に渡す。
  - リスト・レコードを含む変数の名前が REXX EXEC の IDIWRITE コマンドで渡されない限り、リストまたはサイド・ファイル・データ・レコードを DATA\_BUFFER フィールドに指定する必要があります。
  - 可変長レコードの場合、DATA\_BUFFER のレコードの長さを DATA\_LENGTH フィールドに 10 進文字フォーマットで指定する必要があります。この長さは、可変長レコード記述子語を含まず、LRECL フィールドの値から 4 バイトを引いたものより小さいか等しくなければなりません。
  - 固定長レコードの場合、DATA\_BUFFER のレコードの長さが LRECL フィールドと一致することが予期されます。「DATA\_LENGTH」フィールドで指定された値は無視されます。

または

- 順次データ・セットまたは PDS あるいは PDSE の名前をメンバー指定とともに「DATA\_BUFFER」フィールドに指定し、DATA\_BUFFER\_DSN を「Y」に設定する。データ・セット名は、大文字で、「DATA\_BUFFER」フィールドへのオフセット・ゼロから開始し、完全修飾名でなければならず、引用符で囲むことはできません。有効なデータ・セット名の例は、次のとおりです。

```
MY.SEQ.DS
MY.PDS.DS(MBR)
```

データ・セット名の後に、ブランク以外の文字を指定しないでください (ユーザー出口が呼び出される前に、DATA\_BUFFER フィールドはブランクに初期化されます)。

データ・セット名を指定しないで、代わりに個々のデータ・レコードを返す場合、LST データ域をデータ・レコード情報で更新した後、出口タイプに応じて Fault Analyzer にデータを渡すために 2 つの異なる方法を使用できます。

## REXX

Fault Analyzer 環境 IDIWRITE コマンドが使用されます。このコマンドを使用するには、REXX ステートメントを次のようにコーディングしてください。

```
ADDRESS FAULTA 'IDIWRITE [var-name]'
```

IDIWRITE コマンドの正常終了は、ゼロの戻りコードで示されます。

IDIWRITE コマンドの詳細情報については、[IDIWRITE コマンド ページ 535](#)を参照してください。

## ロード・モジュール

書き込みルーチンのアドレスは、ENV.WRITE\_ROUTINE\_EP データ域フィールドに 16 進数 31 ビット・アドレスで示されます。

このルーチンは、ENV データ域のアドレスを含むフルワードをポイントする R1 で呼び出す必要があります。例えば、さまざまなプログラム言語で作成された以下のユーザー出口のコードを使用して、書き込みルーチンを呼び出すことができます。

### アセンブラー:

```
ASMEXIT CSECT
...
L      R2,0(,R1)
USING ENV,R2
L      R3,4(,R1)
USING LST,R3
```

```

...
L    R15,ENV_WRITE_ROUTINE_EP
LA   R1,ENV_VERSION
ST   R1,++8
BAL  R1,++8
DC   F'0'
BALR R14,R15
...
COPY IDISXPLA
...

```

**C:**

```

#include "SAMPLES(IDISXPLC)"
typedef void WRTN(ENV *pENV);
#pragma linkage(WRTN,OS)
int cexit(ENV *pENV, LST *pLST) {
    ...
    WRTN *write_rtn;
    write_rtn = (WRTN *)pENV->WRITE_ROUTINE_EP;
    write_rtn(pENV);
    ...
}

```

**COBOL:**

```

...
PROGRAM-ID. COBEXIT
...
LINKAGE SECTION.
    COPY IDISXPLB IN LIB.
PROCEDURE DIVISION USING ENV, LST.
MAIN SECTION.
    ...
    CALL WRITE-ROUTINE-EP USING ENV.
    ...
END PROGRAM COBEXIT.

```

**PL/I:**

```

PLIEXIT: PROC (ENVPTR,LSTPTR)  OPTIONS(BYVALUE,FETCHABLE) ;
Dcl (Envptr,Lstptr)    Pointer ;
%include syslib(IDISXPLP) ;
Dcl IDIWRITE          Entry Variable Options(Asm Byaddr) ;
...
    Entryaddr(IDIWRITE) = Envptr->Write_Routine_EP ;
    Call IDIWRITE (Envptr->Env) ;
...
End PLIEXIT ;

```

書き込みルーチンからの戻りコードは、RC=8 (構文エラー) が戻されないことを除いて、IDIWRITE REXX コマンドからの戻りコードと同じです。 [IDIWRITE コマンド ページ 535](#)を参照してください。

ユーザー出口により提供されているリストまたはサイド・ファイルの使用をキャンセルするために使用できるインディケータは、LST データ域フィールド DISREGARD\_EXIT\_LISTING に示されます。このフィールドが「Y」に設定されている場合、ユーザー出口から Fault Analyzer に渡された可能性のあるデータ・レコードが廃棄されます。このインディケータに

よって、ユーザー出口は、データ・レコードの提供中にエラーが発生した場合に Fault Analyzer による部分リストの使用を防ぐことができます。

## 呼び出されるタイミング

この出口は、任意の Fault Analyzer 実行モードでソース・コード情報が要求される場合は常に呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、Compiler Listing Read ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

## REXX

出口は、以下の 2 つのシステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- LST.

LST データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([LST - Compiler Listing Read ユーザー出口パラメーター・リスト ページ 668](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の 2 つのフルワードを構成するパラメーター・リストの 31 ビット・アドレスが含まれます。

- word 1 の 31 ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の 31 ビット LST アドレス。

LST データ域のアドレス ([LST - Compiler Listing Readユーザー出口パラメーター・リスト ページ 668](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例

以下は、REXX で書かれたCompiler Listing Readユーザー出口の例です。区分データ・セット myid.LISTING.TERSE には、IBM® AMATERSE ユーティリティーによって作成された圧縮コンパイラー・リストが含まれます。

Fault Analyzer がコンパイラー・リスト・ファイルを要求するときに、この出口は myid.LISTING.TERSE PDS を検索して、モジュール名と一致するメンバー名を探します。一致するメンバーが検出されると、出口はそのメンバーを作業データ・セットに解凍し、IDIWRITE コマンドを使用して Fault Analyzer に渡します。

図 227. REXX コンパイラー・リスト読み取りユーザー出口の例

```
/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if LST.VERSION <> 1 then
  say 'Note: LST data area version change - field usage review required!'
if LST.LISTING_TYPE = 'L' then do
  modnm = strip(LST.MODULE_NAME)
  "IDIALLOC DD(INFILE) DSN(myid.LISTING.TERSE("||modnm||")) SHR"
  if rc = 0 then do
    recfm = strip(LST.RECFM)
    lrecl = strip(LST.LRECL,L,0)
    "IDIALLOC DD(OUTFILE) DSN(myid.TEMP) NEW CATALOG SPACE(1,1) ",
      "RECFM("||recfm||") LRECL("||lrecl||") UNIT(SYSALLDA)"
    address linkmvs "amaterse unpack"
    address mvs "execio * diskr outfile (finis"
    do while queued() <> 0
      parse pull rec
      LST.DATA_LENGTH = length(rec)
      LST.DATA_BUFFER = rec
      "IDIWRITE"
    end
    "IDIFREE DD(INFILE,OUTFILE)"
  end
end
end
exit 0
```

前述のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかで次のオプションを使用すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))
Exits(LISTING(REXX(ABC)))
```

## メッセージおよび異常終了コード説明ユーザー出口

メッセージおよび異常終了コード説明ユーザー出口について、以下で説明します。

## 目的

この出口は、分析レポートのメッセージおよび異常終了コードの説明を提供できます。Message and Abend Code Explanation がメッセージの説明または異常終了コードの説明のいずれかが使用可能かどうかを判別した後で、分析レポートに説明を表示する前に、Fault Analyzerユーザー出口が制御を受け取ります。Fault Analyzer がソフトコピー・ブックまたはオーバーライド・データ・セットを使用して説明を検出した場合、「XPL.EXPLANATION\_AVAILABLE」データ領域フィールドは「Y」設定されます。

データ域の詳細については、[XPL - Message and Abend Code Explanationユーザー出口パラメーター・リスト ページ 689](#)を参照してください。

Message and Abend Code Explanation ユーザー出口は、以下を行うことができます。

- 何も行わずに使用可能な説明が検出された場合、これを保存する。
- 説明が検出されなかった場合に説明を指定するか、または検出された説明を置換する。

メッセージの説明が Fault Analyzer によりフォーマットされている場合、以下の情報が XPL データ域に示されます。

- 発行されたメッセージの実際のテキスト。

このテキストは MESSAGE\_TEXT1 から MESSAGE\_TEXT10 までのフィールドに示されます。MESSAGE\_TEXT1 はメッセージの最初の行または 1 行のみのメッセージを含むように初期化され、メッセージが複数行から構成される場合、MESSAGE\_TEXT2 から MESSAGE\_TEXT10 までが使用されます。



**注:** 発行されたメッセージの実際のテキストが使用不可の場合、メッセージ ID のみが MESSAGE\_TEXT1 フィールドに置かれます。

異常終了コードの説明が Fault Analyzer によりフォーマット設定されている場合、以下の情報が XPL データ域に示されません。

- 異常終了コード。

このコードはフィールド ABEND\_CODE に示されます。このフィールドのコメントは異常終了タイプによって異なります。詳しくは下記を参照してください。

- 異常終了理由コード。

このコードは ABEND\_REASON\_CODE フィールドに 8 文字の 16 進値として示されます。

- 異常終了モジュール名。

この名前は ABEND\_MODULE\_NAME フィールドに示されます。

- 異常終了タイプ。

このタイプは ABEND\_TYPE フィールドに示されます。ここには、以下のいずれかの値が含まれます。

**C**

CICS® トランザクション異常終了を示します。

フィールド「ABEND\_CODE」には、4文字の CICS® 異常終了コードが含まれます。

**D**

CICS® ダンプ・コードを示します。

フィールド「ABEND\_CODE」には、4文字の CICS® ダンプ・コードが含まれます。

**S**

システム異常終了を示します。

フィールド ABEND\_CODE には3文字の16進左寄せシステム異常終了コードが含まれます。

**U**

ユーザー異常終了を示します。

フィールド「ABEND\_CODE」には、4文字の10進ユーザー異常終了コードが含まれます。

上記の情報に基づいて、ユーザー出口は不足していた説明を提供したり、Fault Analyzer により検出された説明の置換を行います。このプロビジョンは、XPL データ域経由で一度に1つの説明レコードを Fault Analyzer に渡すことによって行われます。

説明レコードを含む変数の名前が REXX EXEC の IDIWRITE コマンドで渡されない限り、説明データ・レコードを DATA\_BUFFER フィールドに指定する必要があります。

XPL データ域をデータ・レコード情報で更新すると、Fault Analyzer にデータを渡すために出口タイプに応じて2つの異なる方法を使用できます。

**REXX**

Fault Analyzer 環境 IDIWRITE コマンドが使用されます。このコマンドを使用するには、REXX ステートメントを次のようにコーディングしてください。

```
ADDRESS FAULTA 'IDIWRITE [var-name]'
```

IDIWRITE コマンドの正常終了は、ゼロの戻りコードで示されます。

IDIWRITE コマンドの詳細情報については、[IDIWRITE コマンド ページ 535](#)を参照してください。

**ロード・モジュール**

書き込みルーチンのアドレスは、ENV.WRITE\_ROUTINE\_EP データ域フィールドに16進数31ビット・アドレスで示されます。

このルーチンは、ENV データ域のアドレスを含むフルワードをポイントする R1 で呼び出す必要があります。例えば、さまざまなプログラム言語で作成された以下のユーザー出口のコードを使用して、書き込みルーチンを呼び出すことができます。

**アセンブラー:**

```

ASMEXIT CSECT
...
L    R2,0(,R1)
USING ENV,R2
L    R3,4(,R1)
USING XPL,R3
...
L    R15,ENV_WRITE_ROUTINE_EP
LA   R1,ENV_VERSION
ST   R1,++8
BAL  R1,++8
DC   F'0'
BALR R14,R15
...
COPY IDISXPLA
...

```

**C:**

```

#include "SAMPLES(IDISXPLC)"
typedef void WRTN(ENV *pENV);
#pragma linkage(WRTN,OS)
int cexit(ENV *pENV, XPL *pXPL) {
...
WRTN *write_rtn;
write_rtn = (WRTN *)pENV->WRITE_ROUTINE_EP;
write_rtn(pENV);
...
}

```

**COBOL:**

```

...
PROGRAM-ID. COBEXIT
...
LINKAGE SECTION.
    COPY IDISXPLB IN LIB.
PROCEDURE DIVISION USING ENV, XPL.
MAIN SECTION.
...
    CALL WRITE-ROUTINE-EP USING ENV.
...
END PROGRAM COBEXIT.

```

**PL/I:**

```

PLIEXIT: PROC (ENVPTR,XPLPTR) OPTIONS(BYVALUE,FETCHABLE) ;
Dcl (Envptr,Xplptr) Pointer ;
%include syslib(IDISXPLP) ;
Dcl IDIWRITE Entry Variable Options(Asm Byaddr) ;
...
Entryaddr(IDIWRITE) = Envptr->Write_Routine_EP ;
Call IDIWRITE (Envptr->Env) ;
...
End PLIEXIT ;

```



書き込みルーチンからの戻りコードは、RC=8 (構文エラー) が戻されないことを除いて、IDIWRITE REXX コマンドからの戻りコードと同じです。IDIWRITE コマンド ページ 535を参照してください。

メッセージまたは異常終了コードの説明を構成する文字の合計数が、次の数式の *max-chars* を超えてはなりません。

```
max-chars = 32752 - (num-recs * 2)
```

*num-recs* はレコードの合計数です。

文字の最大数が *max-chars* を超える原因となるレコードを渡そうとすると、IDIWRITE (またはロード・モジュール出口により使用される書き込みルーチン) によって RC=4 が戻され、レコードは無視されます。

Fault Analyzer ではテキストのフォーマットは行われません。すべてのレコードは、Message and Abend Code Explanationユーザー出口により指定されたとおりに分析レポートに示されます。

表示時にフォーマット幅を超える長さを持つレコードは、現行インデント (各レコード内の先頭ブランク文字数によって決定される) でラップされます。

複数のMessage and Abend Code Explanationユーザー出口が特定のメッセージまたは異常終了コードの説明を提供している場合は、最後の説明のみが使用されます。

## 呼び出されるタイミング

この出口は、Fault Analyzer の実行モードにかかわらず分析レポートのフォーマットで呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、Message and Abend Code Explanation ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

## REXX

出口は、以下の2つのステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- XPL.

XPL データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([XPL - Message and Abend Code Explanationユーザー出口パラメーター・リスト ページ 689](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の 2 つのフルワードを構成するパラメーター・リストの 31 ビット・アドレスが含まれます。

- word 1 の 31 ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の 31 ビット XPL アドレス。

XPL データ域のアドレス ([XPL - Message and Abend Code Explanationユーザー出口パラメーター・リスト ページ 689](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例

以下は、REXX で書かれた Message and Abend Code Explanationユーザー出口の例です。

図 228. REXX メッセージおよび異常終了コード説明ユーザー出口の例

```
/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if XPL.VERSION <> 1 then
  say 'Note: XPL data area version change - field usage review required!'
parse var XPL.MESSAGE_TEXT1 msgid msgtext
if msgid = 'MYMSG01' then do
  rec = 'This message indicates that:'
  'IDIWRITE rec'
  rec = ' - A serious problem has occurred'
  'IDIWRITE rec'
  rec = ' - Any data produced should be ignored'
  'IDIWRITE rec'
end
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))
Exits(MSGXPL(REXX(ABC)))
```

## フォーマット・ユーザー出口



**注:** 通常のロード・モジュール・フォーマット・ユーザー出口は、フォーマット・ユーザー出口としてサポートされていません。ただし、特殊なロード・モジュール・フォーマット・ユーザー出口である IDIXUFMT が使用可能です。詳しくは、[IDIXUFMT ロード・モジュール・フォーマット・ユーザー出口 ページ 552](#)を参照してください。

## 目的

この出口は、分析レポートにユーザー固有のセクションを作成するために使用できます。ユーザー固有のセクションは、例えば分析中のアプリケーション環境に固有のデータ域をフォーマットするために使用できます。

このタイプの出口を呼び出すには、以下のような方法があります。

### 1. Exits オプションを使用する。

このオプションを使用すると、System-Wide Information セクションと Abend Job Information セクションの間にユーザー分析レポート・セクションが挿入されます。Exits オプションを指定する方法については、[Exits ページ 595](#)を参照してください。

リアルタイムまたはバッチ再分析レポートでは、以下に示すように、System-Wide Information セクションと Abend Job Information セクションの間にユーザー分析セクションが挿入されます。

```

:
<H1> S Y S T E M - W I D E   I N F O R M A T I O N
:
<H1> U S E R
:
<H1> A B E N D   J O B   I N F O R M A T I O N
:

```

対話式再分析レポートでは、以下に示すように、Interactive Reanalysis Report 画面からユーザー分析セクションを選択できます。

```

1. Synopsis
2. Event Summary
3. System-Wide Information
4. User
5. Abend Job Information
6. Options in Effect

```

呼び出されたすべてのフォーマット・ユーザー出口からのデータがレポートに組み込まれます。

デフォルトの「User」の見出しは、UFM.USEROPTIONTITLE を別の値に設定することによって変更できます。いずれかのフォーマット・ユーザー出口で設定された最後の値が使用されます。

### 2. EXEC コマンドを使用する。

このオプションは、Interactive Reanalysis Report からのみ使用できます。EXEC コマンドの使用に関する詳細は、[EXEC ページ 108](#)を参照してください。

3. APF 許可ライブラリー内の IDIXUFMT というロード・モジュールを使用可能にする。このタイプのフォーマット・ユーザー出口の詳細については、[IDIXUFMT ロード・モジュール・フォーマット・ユーザー出口 ページ 552](#)を参照してください。

フォーマット出口では、出口固有の UFM データ域の障害点イベントに関する情報が最初に提供されます。(UFM データ域の参照については[パラメーター ページ 493](#)を参照。)しかし、IDIEventInfo コマンドを使用すると、すべてのイベントの情報を入手できます。この手順は [IDIEventInfo コマンド ページ 530](#)で説明されています。

ロード・モジュールの存在に関する情報(そのロード・アドレスおよび長さを含む)を取得するには、IDIModQry コマンドを使用します。この手順は [IDIModQry コマンド ページ 532](#)で説明されています。

分析後の環境からストレージを取得するか、レポートにデータを書き込むには、以下のような追加コマンドが使用できません。

#### Evaluate

このコマンドを使用して、分析後の環境からストレージを検索します。

このコマンドの詳細については、[EVALUATE コマンド ページ 519](#)を参照してください。

#### List

このコマンドを使用して、レポートにストレージを印刷します。

このコマンドの詳細については、[LIST コマンド ページ 538](#)を参照してください。

#### Note

このコマンドを使用して、レポートに 1 行のテキストを印刷します。

このコマンドの詳細については、[NOTE コマンド ページ 540](#)を参照してください。

LIST コマンドと NOTE コマンドのほか、HTML に類似したタグ言語も使用可能で、レポートの情報をさらに柔軟にフォーマットできます。このタグの詳細については、[フォーマット・タグ ページ 542](#)を参照してください。

タグ付きテキストを Fault Analyzer に返すには、以下の 3 つの方法のいずれかで IDIWRITE コマンドを使用します。

1. 引用符付きストリングを使用する。

例:

```
IDIWRITE '<p>Paragraph text.'
```

ストリングを囲むために、単一引用符 (') または二重引用符 (") のいずれかを使用できます。ただし、両方の文字は同じタイプである必要があります。

ストリングに、そのストリングを囲むために使用されている引用符と同じ文字が含まれている場合、その文字を重複して 2 回指定してください。つまり、次のストリングを戻す場合

```
'The TCB's address is not zero'
```

次のように指定します。

```
'The TCB''s address is not zero'
```

## 2. 変数を使用する。

例:

```
data = '<p>Paragraph text.'
IDIWRITE data
```

## 3. UFM データ域を使用する。

例:

```
UFM.DATA_BUFFER = '<p>Paragraph text.'
UFM.DATA_LENGTH = length(UFM.DATA_BUFFER)
IDIWRITE
```

この方法は、主に非 REXX 出口用に提供されています。

LIST コマンドと NOTE コマンドは、フォーマットに副次作用を与えることなく、タグ言語と合わせて使用できません。

## 呼び出されるタイミング

この出口は、Fault Analyzer の実行モードにかかわらず分析レポートのフォーマットで呼び出されます。さらに、このタイプの出口は、Interactive Reanalysis Report から EXEC コマンドを使用して、オンデマンドで呼び出すことができます。詳細については、[ユーザー固有レポートのフォーマット ページ 237](#)を参照してください。

## パラメーター

Fault Analyzer は、Message and Abend Code Explanation ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

出口は、以下の 2 つのステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照)。

- UFM.

UFM データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([UFM - ユーザー出口パラメーター・リストのフォーマット ページ 673](#)を参照)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## サンプル・フォーマット・ユーザー出口

次の REXX サンプルは IDI.SIDISAM1 のデータ・セット・メンバーとして指定されます。

サンプル出口がデータ・セット X.Y.Z のメンバー IDISUFM $n$  として存在する場合、IDICNF $xx$  構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が分析中に呼び出されま

```
DataSets (IDIEXEC(X.Y.Z))
Exits (FORMAT(REXX(IDISUFM $n$ )))
```

または、Interactive Reanalysis Report で EXEC IDISUFM $n$  コマンドを入力して、この出口を呼び出すこともできます。

### IDISUFM1

TCB 情報を表示するサンプル Fault Analyzer フォーマット・ユーザー出口。

### IDISUFM2

CICS CWA 情報を表示するサンプル Fault Analyzer フォーマット・ユーザー出口。

### IDISUFM3

フォーマット・タグの使用法を示すサンプル Fault Analyzer フォーマット・ユーザー出口。

### IDISUFM4

Hogan アプリケーションで使用するサンプル Fault Analyzer フォーマット・ユーザー出口。

### IDISUFM5

MVS ダンプ分析バッチ・ジョブで使用して、指定のヒストリー・ファイル内に障害項目を作成するサンプル Fault Analyzer フォーマット・ユーザー出口。この出口ではオプションとして、作成された障害項目に関する情報を使用して WTO メッセージの書き込みまたは E メール送信を行うことができます。

### IDISUFM6

LE CAA と CIB をフォーマット設定するサンプル Fault Analyzer フォーマット・ユーザー出口。

### IDISUFM7

Name/Token 値を表示するサンプル Fault Analyzer フォーマット・ユーザー出口。

## 終了処理ユーザー出口

終了処理ユーザー出口について、以下で説明します。

### 目的

この出口を使用して、Fault Analyzer により行われる分析後アクションを制御できます。

## • ヒストリー・ファイル選択

分析の後、分析制御ユーザー出口を呼び出した時点では使用可能でなかった、障害に関する詳細情報が使用可能になります。これに基づいて、End Processingユーザー出口は ENV.IDIHIST データ域フィールドを変更することにより、使用するヒストリー・ファイルを変更する選択ができます。

重複障害に関する情報は選択したヒストリー・ファイルに依存するため、ヒストリー・ファイル名が変更されなくなるまで、End Processingユーザー出口が繰り返し呼び出されます。再呼び出しごとに、前の呼び出しで指定されたヒストリー・ファイルで、重複障害情報が更新されます。

ENV.IDIHIST フィールドに無効なヒストリー・ファイルが指定されると、End Processingユーザー出口の再呼び出しは行われず、End Processingユーザー出口が最後に呼び出されたときに現行であったヒストリー・ファイルが使用されることになります。

## • 重複する障害の判別

障害の特性が一致し、互いの NoDup(NORMAL(*hours*)) オプションに対して有効な時間数内に障害が発生した場合、デフォルトで、Fault Analyzer は、これらの障害を同じヒストリー・ファイル内の別の障害の重複であると見なしません (詳細については、[NoDup ページ 611](#) を参照してください)。End Processingユーザー出口は、重複する障害を判別するための別の時間間隔を、NoDup オプションにより有効な時間間隔に適用することを、インストール・システムに許可します。

特定の障害特性のみに基づく重複する障害が検出された場合、次の情報が提供されます。

- フィールド EPC.MINUTES\_SINCE\_LAST\_DUP は、重複する障害を最後に記録してから経過した分数に初期化されます。値は 0 から 99999 の範囲になります (このフィールドの制限を超える値はすべて最大値の 99999 として示されます)。値が示されていない場合は、重複する障害が検出されていません。
- フィールド EPC.DUPLICATE\_COUNT には、障害が重複であると見なされた合計回数が示されます (現行障害は含まれません)。

この合計は、同じヒストリー・ファイル内に記録されている重複障害のすべてのインスタンスを、NoDup(Normal(...)) が有効であった期間にわたって集計することで決定されます。検出されて記録された障害のうち、重複基準が現行障害に適合するものはいずれも、1つのインスタンスとして計上されます。また、障害に対して重複が記録された場合は、その障害の重複カウントも加算されます。

- フィールド ENV.FAULT\_ID には、記録された重複する障害が障害 ID によって示されます。
- フィールド [ENV.DUP\_DATE] および [ENV.DUP\_TIME] により、最新の重複障害の日時が識別されます。
- フィールド [ENV.ORIGINAL\_DATE] および [ENV.ORIGINAL\_TIME] により、当初の障害が記録された日時が識別されます。これらのフィールドによって、ユーザーは必要に応じて、最後の重複障害とは対照的に、当初の障害以降の時間に基づいて重複を判別できます。

両方の障害特性が一致し、経過時間が、NoDup(NORMAL(*hours*)) オプションの有効な時間数を超えなかった場合、EPC.IS\_DUPLICATE フィールドは [Y] に初期化されます。ただし、障害が重複しているかどうかの最終的な判別は End Processingユーザー出口で行われます。フィールドの戻り値が [Y] の場合、最後に記録された重複する障害 (存在する場合) の重複カウントに 1 が加算され、メッセージ [ID10044I ページ 705](#) が発行されます。



**注:** CICS® または IMS™ の高速重複障害抑止はこの出口を使用して制御することはできません。これらのタイプの重複の判別に影響を与えるのは、NoDup(CICSFAST(...)) または NoDup(ImageFast(...)) オプション (NoDup ページ 611 を参照) を使用した場合に限られます。ただし、CICS® または IMS™ 下で発生する障害が高速重複と見なされない場合には、その障害は、ターゲット・ヒストリー・ファイル内の既存の項目、および有効な NoDup(NORMAL(hours)) オプションに基づいて、通常のタイプの重複抑止として考慮されません。

### ・ヒストリー・ファイル更新

現行の障害が同じヒストリー・ファイル内の以前に記録された障害の重複であると判別された場合、デフォルトで Fault Analyzer はミニダンプを含む障害項目全体を抑止します。ただし、ミニダンプのサイズが有効な MaxMinidumpPages オプションにより強制された制限を超える場合にも、ミニダンプが抑止されることがあります。

次のフィールドに、考えられる抑止の原因が示されます。

- ENV.MINIDUMP\_PAGES フィールド。4K ページ数でのミニダンプのサイズが含まれます。
- EPC.IS\_DUPLICATE フィールド (上記の「重複する障害の判別」を参照してください)。

Fault Analyzer が抑止を行うかどうかは、以下のフィールドの初期化により指示されます。これらのフィールドは両方とも End Processing ユーザー出口によりオーバーライドできます。

- EPC.SUPPRESS\_MINIDUMP フィールド。「Y」に設定されている場合、ミニダンプはヒストリー・ファイルに書き込まれません。「N」に設定されている場合、ミニダンプはサイズにかかわらず書き込まれます。
- EPC.SUPPRESS\_FAULT\_ENTRY フィールド。「Y」に設定されている場合、現行の障害はヒストリー・ファイル (ミニダンプを含む) に記録されません。「N」に設定されている場合、障害記録は実行され、EPC.SUPPRESS\_MINIDUMP フィールドに応じてミニダンプを書き込むことができます。

### ・ダンプの抑止

End Processing ユーザー出口は、ダンプを抑止する場合は EPC.SUPPRESS\_DUMP フィールドを「Y」に、ダンプが行われることを許可する場合は「N」に設定できます。

ダンプの抑止に関する一般情報は、[ダンプの抑止 ページ 33](#) を参照してください。

## 呼び出されるタイミング

この出口は、ヒストリー・ファイルの更新前、リアルタイム分析の完了時に呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX または ロード・モジュール) によって異なります。

Fault Analyzer は、End Processing ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。



## REXX

出口は、以下の2つのシステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- EPC.

EPC データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([EPC - End Processingユーザー出口パラメーター・リスト ページ 667](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の2つのフルワードを構成するパラメーター・リストの31ビット・アドレスが含まれます。

- word 1 の31ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の31ビット EPC アドレス。

EPC データ域のアドレス ([EPC - End Processingユーザー出口パラメーター・リスト ページ 667](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例

以下は、REXX で書かれたEnd Processingユーザー出口の例です。

図 229. REXX 終了処理ユーザー出口の例

```
/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if EPC.VERSION <> 1 then
  say 'Note: EPC data area version change - field usage review required!'
if EPC.MINUTES_SINCE_LAST_DUP ^= ' ' & EPC.MINUTES_SINCE_LAST_DUP < 48*60 then do
  /* Use 48 hours as the duplicate fault threshold */
  EPC.IS_DUPLICATE = 'Y'
  EPC.SUPPRESS_FAULT_ENTRY = 'Y'
end
EPC.SUPPRESS_DUMP = 'N' /* Always permit dumps to be taken */
exit 0
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))
Exits(END(REXX(ABC)))
```

## 終了処理ユーザー出口 (障害項目リフレッシュ)

障害項目リフレッシュ終了処理ユーザー出口について、以下で説明します。

### 目的

この出口は履歴・ファイル更新の制御に使用できます。

Fault Analyzer は、ミニダンプのサイズが有効な MaxMinidumpPages オプションにより強制された制限を超える場合にも、デフォルトでミニダンプを抑制します。

ミニダンプのサイズは、ENV.MINIDUMP\_PAGES フィールドに 4K ページの数で示されます。

Fault Analyzer が抑制を実行するかどうかは、EPC.SUPPRESS\_MINIDUMP データ域フィールドの初期化により示されます。このフィールドは End Processing ユーザー出口によりオーバーライドできます。「Y」に設定されている場合、ミニダンプは履歴・ファイルに書き込まれません。「N」に設定されている場合、ミニダンプはサイズにかかわらず書き込まれます。

End Processing ユーザー出口は、EPC.SUPPRESS\_FAULT\_ENTRY データ域フィールドを使用して、障害項目すべてのリフレッシュを抑制する選択をする可能性があります。「Y」に設定されている場合、現行の障害はリフレッシュされません(ミニダンプを含む)。「N」に設定されている場合、リフレッシュは実行され、「SUPPRESS\_MINIDUMP」フィールドに応じてミニダンプを書き込むことができます。

### 呼び出されるタイミング

この出口は、履歴・ファイルの更新前、バッチ再分析の完了時に呼び出されます。

### パラメーター

[パラメーター ページ 496](#)を参照してください。

### 例

以下は、REXX で書かれた障害項目リフレッシュ End Processing ユーザー出口の例です。

図 230. 障害項目リフレッシュ REXX 終了処理ユーザー出口の例

```
/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if EPC.VERSION <> 1 then
  say 'Note: EPC data area version change - field usage review required!'
EPC.SUPPRESS_MINIDUMP = 'Y' /* Always suppress the minidump */
exit 0
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))
RefreshExits(END(REXX(ABC)))
```

## 通知ユーザー出口

通知ユーザー出口について、以下で説明します。

### 目的

この出口を使用して、履歴・ファイルへの障害の記録について、または、以前に発生した障害の重複発生について、インストール・システム固有の通知を行うことができます。例えば、出口を使用して、障害が発生したアプリケーションの担当者に E メールを送信できます。また、ADFzCC イベント処理ユーザー出口を介して障害を記録できます。

出口が呼び出された理由は、[NFY データ域フィールド ページ 687](#) の NFYTYPE フィールドに、以下のいずれかのように示されます。

#### C

障害が作成されました。

障害が記録された履歴・ファイルの名前は ENV.IDIHIST

データ域フィールドに提供され、記録された障害に割り当てられた障害 ID は、ENV.FAULT\_ID

データ域フィールドに示されます。

Fault Analyzer は、割り当てられた障害 ID および履歴・ファイルを示すメッセージ [IDI0003I ページ 696](#) を発行します。

リアルタイム・レポートの概要セクションのコピーは、NFY.SYNOPSIS

データ域フィールドから使用可能です。SYNOPSIS の各行は、改行文字 (X'15')

で区切られます。このフィールドに関する詳細については、NFY データ域を参照してください。

#### R

Recovery fault recording

この値は、Fault Analyzer のリカバリー障害記録機能の結果として作成された障害を示します。(この機能の詳細については、[リカバリー障害記録 ページ 57](#)を参照してください。)

障害が記録された履歴・ファイルの名前は ENV.IDIHIST

データ域フィールドに提供され、記録された障害に割り当てられた障害 ID は、ENV.FAULT\_ID

データ域フィールドに示されます。

Fault Analyzer は、割り当てられた障害 ID および履歴・ファイルを示すメッセージ [IDI0126I ページ 724](#) を発行します。

#### N

通常重複。

この値は、NoDup(NORMAL)

オプションの基準が現行障害に一致したため、履歴・ファイル障害項目が書き込まれないことを示します。(NoDup(NORMAL)の詳細については、[NoDup ページ 611](#)を参照してください。)

オリジナルの履歴・ファイル名と障害 ID は、ENV.IDIHIST および ENV.FAULT\_ID データ域フィールドに示されます。

DUPCOUNT フィールドには 1 が設定されます。

## F

高速重複 (CICS®)。

この値は、NoDup(CICSFAST)

オプションの基準が現行障害に一致したため、分析が実行されなかったことを示します。(NoDup(CICSFAST)の詳細については、[NoDup ページ 611](#)を参照してください。)

オリジナルの履歴・ファイル名と障害 ID が使用可能な場合は、それらは ENV.IDIHIST および ENV.FAULT\_ID データ域フィールドに提供されます。

DUPCOUNT フィールドには、30 秒の記録期間内に発生した重複の数が設定されます。

通知ユーザー出口で IDIWRITE コマンドを使用すると、Fault Analyzer は出口の完了後に ADFzCC イベント処理機能呼び出し、IDIWRITE コマンドで書き込まれたデータを使用して ADFzCC イベント処理ユーザー出口に渡します。ADFzCC イベント処理機能の詳細は、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* を参照してください。

複数の通知ユーザー出口によって IDIWRITE コマンドを使用して書き込まれたデータは、1 回の呼び出しで ADFzCC イベント処理ユーザー出口に送信されます。ただし、独自の通知ユーザー出口では IDIWRITE を使用しないことをお勧めします。代わりに、次のように、EXITS オプションでサンプル通知ユーザー出口 IDISXEPN を指定します。

```
EXITS(NOTIFY(<existing exit names,>REXX(IDISXEPN)))
```

また、IDIEXEC ddname によって位置指定できるようにします。

IDISXEPN サンプル出口は、IDIWRITE コマンドを使用して、使用可能なすべての ENV データと NFY データを ADFzCC イベント処理ユーザー出口に送信します。

通知ユーザー出口から IDIWRITE コマンドを発行すると、データは ADFzCC イベント処理ユーザー出口に渡されます。IDIWRITE コマンドを発行する前に、書き込まれるデータは「NFY.DATA\_BUFFER」フィールドに入れられ、データ長は「NFY.DATA\_LENGTH」フィールドで指定されます。

通知ユーザー出口の完了後に、バッファ内容が ADFzCC イベント処理ユーザー出口に送信されます。バッファの形式については、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* のロード・モジュール IPVEPSND についてのセクションを参照してください。バッファの各セグメントには、IDIWRITE コマンドの個々の呼び出しごとの内容が含まれます。提供されているサンプル出口 IDISXEPN を使用すると、各セグメントには次の形式のデータが含まれます。

```
data_area.field_name=value
```

例:

```
ENV.JOB_NAME=BATCH1
```

`NFY.EPX_DEBUG_OPT` を \* または他の非空白文字に設定すると、通知ユーザー出口は、ADFzCC イベント処理ユーザー出口からの診断をすべて `IDITRACE` に書き込むように要求します。



**注:** `NFY.EPX_DEBUG_OPT` が非空白文字に設定された場合、通知ユーザー出口は ADFzCC イベント処理ユーザー出口でデバッグ情報の受信が完了するまで待たなければならないため、ADFzCC イベント処理ユーザー出口は同期的に実行されます。これはパフォーマンスに影響を与える可能性があります。

`NFY.EPX_DEBUG_OPT` を空白 (デフォルト設定) のままにすると、Fault Analyzer はイベント処理ユーザー出口の完了を待たずに処理を続けますが、診断は `IDITRACE` に書き込まれません。

ADFzCC `EVENTPROCESSINGEXIT` オプションを使用してイベント処理ユーザー出口が1つも指定されない場合、データは無視されます。`EVENTPROCESSINGEXIT` オプションによるイベント処理ユーザー出口の指定についての詳細は、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* を参照してください。

通知ユーザー出口で `IDIWRITE` コマンドを使用すると、Fault Analyzer は出口の完了後に ADFzCC イベント処理機能呼び出し、`IDIWRITE` コマンドで書き込まれたデータを使用して ADFzCC イベント処理ユーザー出口に渡します。ADFzCC イベント処理機能の詳細は、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* を参照してください。

複数の通知ユーザー出口によって `IDIWRITE` コマンドを使用して書き込まれたデータは、1回の呼び出しで ADFzCC イベント処理ユーザー出口に送信されます。ただし、独自の通知ユーザー出口では `IDIWRITE` を使用しないことをお勧めします。代わりに、次のように、`EXITS` オプションでサンプル通知ユーザー出口 `IDISXEPN` を指定します。

```
EXITS(NOTIFY(<existing exit names,>REXX(IDISXEPN)))
```

また、`IDIEXEC` `ddname` によって位置指定できるようにします。

`IDISXEPN` サンプル出口は、`IDIWRITE` コマンドを使用して、使用可能なすべての `ENV` データと `NFY` データを ADFzCC イベント処理ユーザー出口に送信します。

通知ユーザー出口から `IDIWRITE` コマンドを発行すると、データは ADFzCC イベント処理ユーザー出口に渡されます。`IDIWRITE` コマンドを発行する前に、書き込まれるデータは「`NFY.DATA_BUFFER`」フィールドに入れられ、データ長は「`NFY.DATA_LENGTH`」フィールドで指定されます。

通知ユーザー出口の完了後に、バッファー内容が ADFzCC イベント処理ユーザー出口に送信されます。バッファーの形式については、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* のロード・モジュール `IPVEPSND` についてのセクションを参照してください。バッファーの各セグメントには、`IDIWRITE` コマンドの個々の呼び出しごとの内容が含まれます。提供されているサンプル出口 `IDISXEPN` を使用すると、各セグメントには次の形式のデータが含まれます。

```
data_area.field_name=value
```

例:

```
ENV.JOB_NAME=BATCH1
```

NFY.EPX\_DEBUG\_OPT を \* または他の非空白文字に設定すると、通知ユーザー出口は、ADFzCC イベント処理ユーザー出口からの診断をすべて IDITRACE に書き込むように要求します。



**注:** NFY.EPX\_DEBUG\_OPT が非空白文字に設定された場合、通知ユーザー出口は ADFzCC イベント処理ユーザー出口でデバッグ情報の受信が完了するまで待たなければならないため、ADFzCC イベント処理ユーザー出口は同期的に実行されます。これはパフォーマンスに影響を与える可能性があります。

NFY.EPX\_DEBUG\_OPT を空白 (デフォルト設定) のままにすると、Fault Analyzer はイベント処理ユーザー出口の完了を待たずに処理を続けますが、診断は IDITRACE に書き込まれません。

ADFzCC EVENTPROCESSINGEXIT オプションを使用してイベント処理ユーザー出口が 1 つも指定されない場合、データは無視されます。EVENTPROCESSINGEXIT オプションによるイベント処理ユーザー出口の指定についての詳細は、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide* を参照してください。

## 呼び出されるタイミング

この出口は、Fault Analyzer が履歴・ファイルへの障害の記録を完了した後に呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、通知ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

## REXX

出口は、以下の 2 つのシステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- NFY.

NFY データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([NFY - 通知ユーザー出口パラメーター・リスト ページ 672](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の 2 つのフルワードを構成するパラメーター・リストの 31 ビット・アドレスが含まれます。

- word 1 の 31 ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の 31 ビット NFY アドレス。

NFY データ域のアドレス ([NFY - 通知ユーザー出口パラメーター・リスト ページ 672](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## REXX からの非 REXX ロギング・ルーチンの呼び出し

例えば、REXX で書かれていない外部ロギング・ルーチンの呼び出しなどを行う、REXX で書かれた通知ユーザー出口のために、2つのステム変数が新たに追加されました。ステム変数は、すべての使用可能なデータ域値を外部ルーチンに渡すために使用できます。これらのステム変数は以下のとおりです。

ENV.RECORD

NFY.RECORD

これらのそれぞれが、単一 REXX 変数内に ENV または NFY データ域全体を含んでいます。それぞれが表すものはデータ域全体であり、それは既に提供されているため、これらの変数は、非 REXX 出口の対応するデータ域の中に、フィールドとして存在しません。

これらの変数を外部の非 REXX ルーチンに対する引数リストの中に使用することで、REXX 出口では、将来発生する可能性のあるこれらのデータ域の変更について配慮する必要がなくなります。

外部ルーチンは、データ値へのアクセスに、提供される言語依存データ域マッピングを使用する必要があります ([ロード・モジュール出口 ページ 468](#)を参照してください)。ENV.RECORD および NFY.RECORD 変数内の値はすべて読み取り専用と見なされるので、外部ルーチンで更新することはできません。更新が必要な場合は、REXX 出口そのものを使って、適切なデータ・フィールド・ステム変数を更新しなければなりません。

### 例 1

以下は、REXX で書かれた通知ユーザー出口の例です。TSO SEND メッセージを発行します。

---

 図 231. サンプル REXX 通知ユーザー出口 1

```

/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if NFY.VERSION <> 2 then
  say 'Note: NFY data area version change - field usage review required!'
/* Issue SEND command via TSO batch job */
queue "//NOTIFY JOB MSGCLASS=Z"
queue "//TSOBATCH EXEC PGM=IKJEFT01"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSTSIN DD *"
/* Split the TSO SEND command over three data records that must
   each be padded with blanks to 80 bytes */
rec = "SEND 'Fault ID' ENV.FAULT_ID "assigned in history file -"
queue left(rec,80)
rec = strip(ENV.IDIHIST)||" -"
queue left(rec,80)
rec = "USER("||strip(ENV.USER_ID)||") LOGON"
queue left(rec,80)
queue '/'
/* 'Submit' the stacked TSO batch job */
n = queued()
"IDIALLOC DD(DD1) SYSOUT PGM(INTRDR)"
if rc = 0 then do
  address mvs "EXECIO" n "DISKW DD1 (FINIS"
  "IDIFREE DD(DD1)"
end
exit 0

```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```

DataSets(IDIEXEC(X.Y.Z))
Exits(NOTIFY(REXX(ABC)))

```

## 例 2

以下は、REXX で書かれた通知ユーザー出口の例です。この出口は、SMTP 経由で E メール・メッセージを送信します。



## 図 232. サンプル SMTP REXX 通知ユーザー出口 2

```

/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if NFY.VERSION <> 2 then
  say 'Note: NFY data area version change - field usage review required!'
/* Issue SMTP message via SMTP batch interface */
user = strip(ENV.USER_ID)
queue "helo pthmvs8.au.ibm.com"
queue "mail from:<"user"@pthmvs8.au.ibm.com>"
queue "rcpt to:<"user"@au1.ibm.com>"
queue "data"
queue "Date: " date('N') time('C')
queue "From:<"user"@pthmvs8.au.ibm.com>"
queue "To:<"user"@au1.ibm.com>"
queue "Subject: Batch job "strip(ENV.JOB_NAME)" abend "ENV.ABEND_CODE
queue " "
queue "Fault ID "ENV.FAULT_ID" assigned in history file"
queue strip(ENV.IDIHIST)" for job "ENV.JOB_NAME
queue "program "ENV.EXEC_PGM_NAME" module "ENV.ABEND_MODULE_NAME"."
queue ""
n = queued()
"IDIALLOC DD(DD1) SYSOUT(A) DEST(PTHMVS8.SMTP)"
address mvs "EXECIO" n "DISKW DD1 (FINIS"
"IDIFREE DD(DD1)"
exit

```

この出口を機能させるには、システムが MVS™ TCP/IP SMTP 環境を使用できなければなりません。

上に示されているサンプル出口は、IBM® Communications Server 製品に提供されている IBM® SMTP サーバーのバッチ・インターフェースを使用して検査されています。

サンプル・メッセージが正常に送達されるかどうかは、IBM® CS TCP/IP サービスと SMTP サーバーの構成、および出口が実行されているシステムで使用できる適切な TCP/IP ネットワーク・インフラストラクチャーによって決まります。

中間ネットワーク・ノードにファイアウォール、セキュリティー・ソフトウェア、およびメッセージ・フィルター・ソフトウェアが存在するなどの要因が、メッセージの正常な送達に影響を及ぼす可能性があります。

上記のサンプル出口がデータ・セット TEST.EXEC.PDS のメンバー NOTIFY1 として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```

DataSets(IDIEXEC(TEST.EXEC.PDS))
Exits(NOTIFY(REXX(NOTIFY1)))

```

## 例 3

以下は、REXX で書かれた通知ユーザー出口の例です。この出口は、収集されたリアルタイム概要のすべての行を抽出し、表示します。

図 233. サンプル SMTP REXX 通知ユーザー出口 3

```

/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if NFY.VERSION <> 2 then
  say 'Note: NFY data area version change - field usage review required!'
/* Show synopsis */
rest = NFY.SYNOPSIS
do while rest<>' '
  parse var rest nextline '15'x rest
  say nextline
end
exit 0

```

例えばこの例は、上記の例 2 と結合でき、E メール・メッセージに概要を取り入れることができます。

上記のサンプル出口がデータ・セット TEST.EXEC.PDS のメンバー FRED として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されません。

```

DataSets(IDIEXEC(TEST.EXEC.PDS))
Exits(NOTIFY(REXX(FRED)))

```

## 例 4

次の例は、REXX で作成された通知ユーザー出口です。これを使用して、バッチ再分析ジョブを実行依頼し、DeferredReport 障害項目の保管レポートを生成し、オプションで、LangxCapture オプションによって使用されるサイド・ファイルを保管します。

図 234. REXX 通知ユーザー出口 4 の例

```

/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if NFY.VERSION <> 2 then
  say 'Note: NFY data area version change - field usage review required!'
queue "//GENREP JOB MSGCLASS=X"
queue "//FA EXEC PGM=IDIDA,"
queue "// PARM=( '/FAULTID("ENV.FAULT_ID")', "
queue "// 'GenerateSavedReport', "
queue "// )"
queue "//IDIHIST DD DISP=SHR,DSN="ENV.IDIHIST
queue "//SYSPRINT DD SYSOUT=*"
/* 'Submit' the stacked batch reanalysis job */
n = queued()
"IDIALLOC DD(DD1) SYSOUT PGM(INTRDR)"
if rc = 0 then do
  address mvs "EXECIO" n "DISKW DD1 (FINIS"
  "IDIFREE DD(DD1)"
end
exit 0

```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイルのいずれかに次のオプションを指定すると、このサンプル出口が呼び出されます。

```
DataSets(IDIEXEC(X.Y.Z))  
Exits(NOTIFY(REXX(ABC)))
```

## 通知ユーザー出口 (MVS SVC ダンプ登録)

ダンプ登録通知ユーザー出口について、以下で説明します。

### 目的

この出口を使用して、ヒストリー・ファイルへの SVC ダンプ障害項目の記録について、インストール・システム固有の通知を行うことができます。

障害が記録されたヒストリー・ファイルの名前は ENV.IDIHIST データ域フィールドに提供され、記録された障害に割り当てられた障害 ID は、ENV.FAULT\_ID データ域フィールドに示されます。

Fault Analyzer は、割り当てられた障害 ID およびヒストリー・ファイルを示すメッセージ [IDI0003I ページ 696](#) を発行します。

### 呼び出されるタイミング

この出口は、Fault Analyzer がヒストリー・ファイルへの MVS™ SVC ダンプ障害項目の登録を完了した後に呼び出されます。

### パラメーター

[パラメーター ページ 502](#)を参照してください。

### 例

以下は、REXX で書かれたダンプ登録通知ユーザー出口の例です。

## 図 235. サンプル REXX ダンプ登録通知ユーザー出口

```

/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if NFY.VERSION <> 2 then
  say 'Note: NFY data area version change - field usage review required!'
/* Issue SEND command via TSO batch job */
queue "//NOTIFY JOB MSGCLASS=Z"
queue "//TSOBATCH EXEC PGM=IKJEFT01"
queue "//SYSTSPRT DD SYSOUT=*"
queue "//SYSTSIN DD *"
/* Split the TSO SEND command over three data records that must
   each be padded with blanks to 80 bytes */
rec = "SEND 'Fault ID' ENV.FAULT_ID "assigned in history file -"
queue left(rec,80)
rec = strip(ENV.IDIHIST)||" -"
queue left(rec,80)
rec = "USER(FRED) LOGON"
queue left(rec,80)
queue '/'
/* 'Submit' the stacked TSO batch job */
n = queued()
"IDIALLOC DD(DD1) SYSOUT PGM(INTRDR)"
if rc = 0 then do
  address mvs "EXECIO" n "DISKW DD1 (FINIS"
  "IDIFREE DD(DD1)"
end
exit 0

```

通常のNotificationユーザー出口と異なり、ENV データ域でユーザー ID は使用できません。

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合は、IDICNFxx 構成メンバーまたは IDIOPTS サブシステムに割り振られる IDIS ユーザー・オプション・ファイルで次のオプションを指定すると、このサンプル出口が呼び出されます。

```

DataSets(IDIEXEC(X.Y.Z))
DumpRegistrationExits(NOTIFY(REXX(ABC)))

```

DumpRegistrationExits オプションを、IDICNFxx parmlib メンバーに指定するか、または IDIOPTS サブシステム JCL の IDIS DD ステートメントに指定する必要があります。DumpRegistrationExits オプションは、IDIOPTS DD ステートメントでどこか他の場所 (CICS® 領域内やバッチ・ジョブ内など) に指定された場合には無視されます。

## IDIUTIL バッチ・ユーティリティー・ユーザー出口タイプの説明

以下は、IDIUTIL で使用可能な各 Fault Analyzer バッチ・ユーティリティー・ユーザー出口タイプの説明です。

### IDIUTIL インポート・ユーザー出口

IDIUTIL インポート・ユーザー出口について、以下で説明します。

## 目的

IDIUTIL インポート・ユーザー出口を使用して、以下を行うことができます。

- **IMPORT** 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーを実行し、ヒストリー・ファイルの管理で障害項目のインポートを制御します。この制御は、UTL.PERFORM\_ACTION データ域フィールドを「Y」（項目をインポートする場合）または「N」（項目をインポートしない場合）に設定することによって行われます。デフォルトで、フィールド UTL.PERFORM\_ACTION は出口を呼び出す前に必ず「Y」に設定されます。UTL データ域の参照については、[パラメーター ページ 509](#)を参照してください。

項目がターゲット・ヒストリー・ファイルに正常にインポートされた場合、ソース・ヒストリー・ファイルから削除されます。

- UTL.IMPORT\_DUMP\_DSN でエクスポートされた障害項目から作成される、関連するダンプ・データ・セットの名前を指定します。この名前は、IDIOPTLM 構成オプション・モジュールの RFRDSN、XDUMPDSN または SDUMPDSN オプションから取得したデフォルト名をオーバーライドします。詳しくは、[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#)を参照してください。

## 呼び出されるタイミング

IMPORT 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーが実行される場合は常に、この出口はヒストリー・ファイル内の障害項目ごとに 1 回呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、IDIUTIL インポート・ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

## REXX

出口は、以下の 2 つのステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- UTL.

UTL データ域内のすべてのフィールドに対して定義されたシンボルが含まれます (UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト [ページ 687](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の 2 つのフルワードを構成するパラメーター・リストの 31 ビット・アドレスが含まれます。

- word 1 の 31 ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の 31 ビット UTL アドレス。

UTL データ域のアドレス ([UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例

IDIROBEX は、REXX で書かれた IDIUTIL ユーザー出口の例です。

この例は、IDI.SIDISAM1 データ・セット・メンバー IDISROBT として組み込まれています。詳しくは、[図 193 : TSO 受信 REXX exec \(IDIROBOT\) の例、パート 1 ページ 357](#)を参照してください。

図 236. REXX IDIUTIL ユーザー出口の例

```

/* REXX */
/* Dump data set names provided by this exit override the equivalent */
/* option setting in IDIOPTLM. */
/* The relationship is as follows: */
/* ENV.ASSOCIATED_DUMP_TYPE IDIOPTLM option */
/* ----- */
/* 'S' SDUMP (SVC dump) SDUMPDSN */
/* 'T' TDUMP (transaction dump) RFRDSN */
/* 'X' XDUMP (extended dump) XDUMPDSN */
/* To disable the allocation of an associated dump data set, set */
/* UTL.IMPORT_DUMP_DSN to 'NULLFILE'. */
if ENV.VERSION <> 5 then
say 'Note: ENV data area version change - field usage review required!'
if UTL.VERSION <> 2 then
say 'Note: UTL data area version change - field usage review required!'
ddsnhlq = 'IDIDUMP' /* <--- verify/change */
UTL.PERFORM_ACTION = 'Y' /* Import current entry (default) */
if ENV.ASSOCIATED_DUMP_DSN <> "" then do
t_parm = ENV.ASSOCIATED_DUMP_TYPE /* S/T/X */
UTL.IMPORT_DUMP_DSN =,
ddsnhlq"."t_parm"DUMP.&SYSNAME..D&YYMMDD..T&HHMMSS..S&SEQ."
end
exit 0

```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合は、次のステートメントを IDIUTIL バッチ・ユーティリティー・ヒストリー・ファイル管理ジョブに指定すると、出口が呼び出されます。

DD ステートメント:

```
//IDIEXEC DD DISP=SHR,DSN=X.Y.Z
```

IDIUTIL バッチ・ユーティリティー制御ステートメント:

```
Exits(IMPORT(REXX(ABC)))
```

## IDIUTIL 削除ユーザー出口

IDIUTIL 削除ユーザー出口について、以下で説明します。

### 目的

この出口は、DELETE 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーを実行し、ヒストリー・ファイルの管理で障害項目の削除を制御できます (詳しくは、[DELETE 制御ステートメント ページ 445](#)を参照してください)。この制御は、データ域フィールド UTL.PERFORM\_ACTION を「Y」(項目を削除する場合) または「N」(項目を削除しない場合) に設定することによって行われます。フィールド UTL.PERFORM\_ACTION は、出口を呼び出す前に「Y」に設定されます (ただし、障害項目がロックされている場合は除く)。この場合、ENV.LOCK\_FLAG がブランクでないと、UTL.PERFORM\_ACTION フラグに「N」が設定されます。UTL データ域の詳細については、[UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください。

ユーザー出口が呼び出される対象の障害項目は、指定された DELETE 制御ステートメント基準に一致する障害項目です。

### 呼び出されるタイミング

DELETE 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーが実行される場合は常に、この出口はヒストリー・ファイル内の障害項目ごとに 1 回呼び出されます。

### パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、IDIUTIL 削除ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

### REXX

出口は、以下の 2 つのシステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- UTL.

UTL データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の 2 つのフルワードを構成するパラメーター・リストの 31 ビット・アドレスが含まれます。

- word 1 の 31 ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の 31 ビット UTL アドレス。

UTL データ域のアドレス ([UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例

以下は、REXX で書かれた IDIUTIL 削除ユーザー出口の例です。

図 237. REXX IDIUTIL 削除ユーザー出口の例

```
/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if UTL.VERSION <> 2 then
  say 'Note: UTL data area version change - field usage review required!'
UTL.PERFORM_ACTION = 'Y' /* Delete current entry */
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、JCL DD ステートメント

```
//IDIEXEC DD DISP=SHR,DSN=X.Y.Z
```

および IDIUTIL バッチ・ユーティリティー制御ステートメント

```
Exits(DELETE(REXX(ABC)))
```

を IDIUTIL バッチ・ユーティリティー・ヒストリー・ファイル管理ジョブに指定すると、出口が呼び出されます。

## IDIUTIL ListHF ユーザー出口

IDIUTIL ListHF ユーザー出口について、以下で説明します。

### 目的

この出口は、LISTHF 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーを実行し、ヒストリー・ファイルの管理で障害項目のリストを制御できます (詳しくは、[LISTHF 制御ステートメント ページ 442](#)を参照してください)。この制御は、データ域フィールド UTL.PERFORM\_ACTION を「Y」(項目をリストする場合)または「N」(項目をリストしない場



合)に設定することによって行われます。フィールド UTL.PERFORM\_ACTION は、出口を呼び出す前に「Y」に設定されます。UTL データ域の詳細については、[UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください。

ユーザー出口が呼び出される対象の障害項目は、指定された LISTHF 制御ステートメント基準に一致する障害項目です。

## 呼び出されるタイミング

LISTHF 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーが実行される場合は常に、この出口はヒストリー・ファイル内の障害項目ごとに 1 回呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、IDIUTIL ListHF ユーザー出口を呼び出す前に、特定の障害および有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

## REXX

出口は、以下の 2 つのシステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- UTL.

UTL データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の 2 つのフルワードを構成するパラメーター・リストの 31 ビット・アドレスが含まれます。

- word 1 の 31 ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の 31 ビット UTL アドレス。

UTL データ域のアドレス ([UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例 1: IDIUTIL ListHF ユーザー出口の呼び出し

以下は、REXX で書かれた IDIUTIL ListHF ユーザー出口の例です。

図 238. REXX IDIUTIL ListHF ユーザー出口 1 の例

```
/* REXX */
if ENV.VERSION <> 5 then
  say 'Note: ENV data area version change - field usage review required!'
if UTL.VERSION <> 2 then
  say 'Note: UTL data area version change - field usage review required!'
UTL.PERFORM_ACTION = 'Y' /* List current entry */
```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、JCL DD ステートメント

```
//IDIEXEC DD DISP=SHR,DSN=X.Y.Z
```

および IDIUTIL バッチ・ユーティリティー制御ステートメント

```
Exits(LISTHF(REXX(ABC)))
```

を IDIUTIL バッチ・ユーティリティー・ヒストリー・ファイル管理ジョブに指定すると、出口が呼び出されます。

## 例 2: IDIUTIL ListHF ユーザー出口を用いた障害項目のカスタム・レポートおよび CSV ファイルの作成

以下は、REXX で書かれた IDIUTIL ListHF ユーザー出口の例です。

この出口の例は、カスタマイズ・レポートの作成方法や、スプレッドシート・アプリケーションへの入力として使用できるコンマ区切りファイルの作成方法を示しています。

使用されているフィールドに加えて、ENV または UTL データ域で使用可能なフィールドであれば、すべて組み込むことができます。

提供されているサンプルによって作成されたレポートには、以下の列が含まれています。

- Fault ID
- 日付
- 時刻
- ロック
- ユーザー名
- ユーザー・タイトル
- CPU Sec

列データの説明については、[選択可能な列 ページ 73](#)を参照してください。ただし「CPU Sec」を除きます。これは、ENV.CPU\_HSECONDS を基に Fault Analyzer で使用された合計 CPU 時間です。この値の詳細については、[#unique\\_25\\_Connect\\_42\\_CPU\\_HSECONDS ページ 660](#)を参照してください。

カスタマイズされたレポートは、DDname MYREP に書き込まれます。MYREP DD ステートメントは、この出口を呼び出す IDIUTIL ジョブに組み込みます。例えば、以下のようにします。

```
//MYREP DD SYSOUT=*
```

コンマ区切りファイルは DDname COMMA に書き込まれます。COMMA DD ステートメントは、この出口を呼び出す IDIUTIL ジョブに組み込みます。例えば、以下のようにします。

```
//COMMA DD SYSOUT=*
```

永続ユーザー・フィールド ENV.USER\_1 は、レポート・ヘッダーの書き込みが完了したという事実を記録するために使用されます。

---

 図 239. REXX IDIUTIL ListHF ユーザー出口 2 の例

```

/* First ensure that the current data area versions match the      */
/* versions as at the time of coding the exit.                    */
If ENV.VERSION <> 5 Then
  Say 'Note: ENV data area version change - field usage review',
    'required!'
If UTL.VERSION <> 2 then
  Say 'Note: UTL data area version change - field usage review',
    'required!'
If ENV.USER_1='' Then Do
  /* Write report header */
  out.1="Fault ID Date      Time      Lock Username",
        "User Title                CPU Sec"
  out.2="-----",
        "-----"
  ADDRESS MVS "EXECIO 2 DISKW MYREP (STEM out."
  /* Write comma-delimited file header */
  out.1="Fault ID,Date,Time,Lock,Username,User Title,CPU Sec"
  ADDRESS MVS "EXECIO 1 DISKW COMMA (STEM out."
  ENV.USER_1='done' /* Flag header done. */
End
/* The fault ID value is placed right-aligned in a work field.    */
fault_id=COPIES(' ',8-length(ENV.FAULT_ID))||ENV.FAULT_ID
/* The following lines use the REXX INSERT command to ensure that the */
/* work fields for each value are padded with blanks to fit the      */
/* report column width.                                             */
/* For information about the maximum width of any field, refer to the */
/* User's Guide and Reference "Data Areas" chapter.                */
abend_date=INSERT(ENV.ABEND_DATE,',',10)
abend_time=INSERT(ENV.ABEND_TIME,',',8)
lock_flag =INSERT(ENV.LOCK_FLAG,',',4)
user_name =INSERT(ENV.USER_NAME,',',8)
user_title=INSERT(ENV.USER_TITLE,',',40)
/* If available, the CPU time in 1/100s of a second is changed to a */
/* number of seconds with two decimal digits.                       */
if ENV.CPU_HSECONDS='' then cpu_sec=''
else cpu_sec=FORMAT(ENV.CPU_HSECONDS/100,4,2)
/* Write report line for this fault entry.                          */
out.1=fault_id abend_date abend_time lock_flag user_name user_title,
      cpu_sec
ADDRESS MVS "EXECIO 1 DISKW MYREP (STEM out."
/* Write comma-delimited line for this fault entry.                */
out.1=fault_id,"abend_date","abend_time","lock_flag","user_name,
      ","user_title","cpu_sec
ADDRESS MVS "EXECIO 1 DISKW COMMA (STEM out."
UTL.PERFORM_ACTION='N' /* Optionally, suppress the standard report. */
Exit 0

```

上図に示すサンプル出口は、IDI.SIDISAM1 データ・セット内のメンバー IDISUTL1 として提供されます。

サンプルの実行には、以下の JCL を使用できます。

```

//IDIUTIL JOB parms
//RUNUTIL EXEC PGM=IDIUTIL
//SYSPRINT DD SYSOUT=*
//MYREP DD SYSOUT=*
//COMMA DD SYSOUT=*

```

```
//IDITRACE DD SYSOUT=*          (Optional)
//IDIEXEC DD DISP=SHR,DSN=IDI.SIDISAM1
//SYSIN DD *
Exits(LISTHF(REXX(IDISUTL1)))
FILES(my.histfile)
LISTHF
/*
```

### 例 3: IDIUTIL ユーザー出口を使用した今後実行する統計分析向けの異常終了データの収集

ヒストリー・ファイルは動的です。障害項目をヒストリー・ファイルから明示的に削除したり、自動のスペース管理を使用して削除したりすることが可能です。障害項目データを分析アプリケーションへの入力として使用したいと考えている場合、障害項目データをヒストリー・ファイルの外部で収集して保持することが必要になる可能性があります。IDI.SIDISAM1 データ・セットの IDISUTL2 メンバーは、この目的のために定期的に行える REXX プログラムのサンプルです。

IDISUTL2 メンバーの最後のサンプル JCL では RUNUTIL ステップと MERGE ステップを実行します:

- RUNUTIL ステップでは、IDISUTL2 を IDIUTIL LISTHF ユーザー出口として実行し、1 つ以上のヒストリー・ファイル・データ・セットから障害項目の CSV 形式ファイルを生成します。(このステップについては、[例 2 ページ 514](#) で説明しています。)
- MERGE ステップでは、ALLDATA DD ステートメントによって指定された累積ファイルに、RUNUTIL ステップで収集した CSV データを付け足します。

## IDIUTIL ListHFDUP ユーザー出口

IDIUTIL ListHFDUP ユーザー出口について、以下で説明します。

### 目的

この出口は、LISTHFDUP 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーを実行し、ヒストリー・ファイルの管理で異常終了インスタンスのリストを制御できます (詳しくは、[LISTHFDUP 制御ステートメント ページ 443](#) を参照してください)。この制御は、データ域フィールド UTL.PERFORM\_ACTION を「Y」(項目をリストする場合) または「N」(項目をリストしない場合) に設定することによって行われます。フィールド UTL.PERFORM\_ACTION は、出口を呼び出す前に「Y」に設定されます。さらに、UTL.DUP\_TYPE が提供され、現在の異常終了インスタンス・タイプの重複に関する情報が提供されます。UTL データ域の詳細については、[UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト ページ 687](#) を参照してください。

ユーザー出口が呼び出される対象の異常終了インスタンスは、指定された LISTHFDUP 制御ステートメント基準に一致する異常終了インスタンスです。

### 呼び出されるタイミング

LISTHFDUP 制御ステートメントを使用して IDIUTIL バッチ・ユーティリティーが実行されるとき、この出口はヒストリー・ファイル内の異常終了インスタンスごとに 1 回呼び出されます。

## パラメーター

パラメーターがどのように出口に渡されるかは、出口タイプ (REXX またはロード・モジュール) によって異なります。

Fault Analyzer は、IDIUTIL ListHFDup ユーザー出口を呼び出す前に、特定の異常終了インスタンスおよび有効な処理オプションの現行値を使用してパラメーター・リストを初期化します。

## REXX

出口は、以下の 2 つのシステムを使用できます。

- ENV.

ENV データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- UTL.

UTL データ域内のすべてのフィールドに対して定義されたシンボルが含まれます ([UTL - IDIUTIL バッチ・ユーティリティ・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください)。

定義された変数名は、フィールド名と同じです。例えば、ENV データ域のフィールド VERSION にアクセスするには、REXX 変数 ENV.VERSION を使用します。

## ロード・モジュール

この出口へのエントリーで、R1 には以下の 2 つのフルワードを構成するパラメーター・リストの 31 ビット・アドレスが含まれます。

- word 1 の 31 ビット ENV アドレス。

ENV データ域のアドレス ([ENV - 共通出口環境情報 ページ 650](#)を参照してください)。

- word 2 の 31 ビット UTL アドレス。

UTL データ域のアドレス ([UTL - IDIUTIL バッチ・ユーティリティ・ユーザー出口パラメーター・リスト ページ 687](#)を参照してください)。



**注:** このパラメーターが渡される最後のパラメーターあることを示すため、高位ビットがオンになります。

## 例

以下は、REXX で書かれた IDIUTIL ListHFDUP ユーザー出口の例です。

#### 図 240. REXX IDIUTIL ListHFDup ユーザー出口の例

```

/* REXX */
if ENV.VERSION <> 5 then
say 'Note: ENV data area version change - field usage review required!'
if UTL.VERSION <> 2 then
say 'Note: UTL data area version change - field usage review required!'
if UTL.DUP_TYPE = 'F'      /* If it is a CICSFast duplicate */
  UTL.PERFORM_ACTION = 'Y' /* list current entry */
else
  UTL.PERFORM_ACTION = 'N' /* otherwise don't */

```

上記のサンプル出口がデータ・セット X.Y.Z のメンバー ABC として存在する場合、次の JCL DD ステートメント

```
//IDIEXEC DD DISP=SHR,DSN=X.Y.Z
```

および次の IDIUTIL バッチ・ユーティリティー制御ステートメント

```
Exits(LISTHFDUP(REXX(ABC)))
```

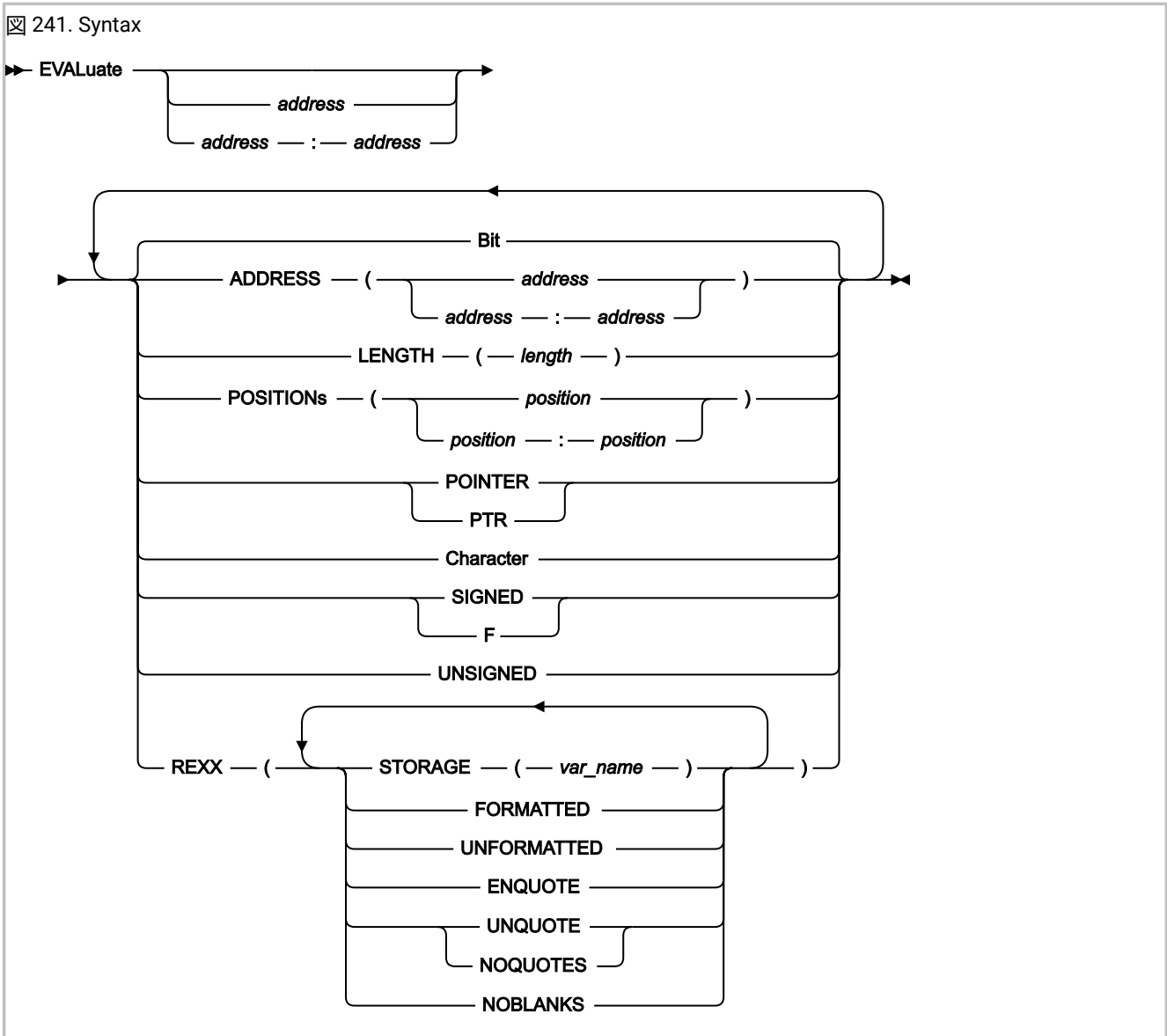
を IDIUTIL バッチ・ユーティリティー・ヒストリー・ファイル管理ジョブに指定すると、出口が呼び出されます。

## ユーザー出口 REXX コマンド

Fault Analyzer は、REXX で作成されたユーザー出口に使用するための以下の特殊な REXX コマンドを提供します。これらのコマンドはすべて、デフォルトの REXX 環境 FAULTA で使用可能です。REXX ADDRESS 命令を使用して別の環境を現行の環境に変更した場合、通常は Fault Analyzer REXX コマンドの前に ADDRESS FAULTA を指定する必要があります。ただし、IPCS 環境の REXX 出口との互換性のために、EVALUATE、LIST、および NOTE コマンドが、ADDRESS FAULTA および ADDRESS IPCS の両方で使用可能です。

## EVALUATE コマンド

EVALUATE コマンドは、フォーマット・ユーザー出口で分析後の障害環境からストレージを取得するために使用できます。



## パラメーター

### アドレス 開始:終了

定位置パラメーターとして、開始アドレスまたはアドレス範囲のいずれかを指定します。このアドレスは、ADDRESS キーワードの使用に代わる方法です。有効な構文の詳細については、次の *ADDRESS* を参照してください。

### ADDRESS(アドレス) ADDRESS(アドレス:アドレス)

戻されるストレージのアドレスを指定します。単一のアドレスまたはアドレス範囲のいずれかで指定します。

address パラメーターは 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。アドレスには、オプションでその後にピリオドを追加できます (例えば「000176C0.」)。



**LENGTH(長さ)**

戻されるバイト数を指定します。16 進数の値 (インディケータを 'X...' で付加) または 10 進数の値 (インディケータは不要) のいずれかで指定します。アドレスの範囲が指定されている場合、LENGTH の指定は無視されます。

**POSITIONs(位置) POSITIONs(位置:位置)**

開始アドレスから、戻されるストレージの最初のバイトまでのオフセットを指定します。単一のオフセットまたはオフセット範囲のいずれかで指定します。POSITION と POSITIONs の両方が有効です。

オフセットはすべて、16 進数 (インディケータを 'X...' で付加) または符号付 10 進数 (オプションで、インディケータを 'F...' で付加) のいずれかで指定する必要があります。

**Bit**

データを 16 進フォーマットで戻すように指定します。

このパラメータは、FORMATTED が有効な場合に限り有効です。

**POINTER PTR**

データを 16 進フォーマットで戻すように指定します。

有効な LENGTH は 1 バイトから 4 バイトです。LENGTH が 4 バイトより大きい場合、4 バイトに変更されます。

このパラメータは、FORMATTED が有効な場合に限り有効です。

**Character**

データを、非印字文字をピリオドに置き換えた文字ストリングのフォーマットで戻すように指定します。ENQUOTE、UNQUOTE、NOQUOTES、または NOBLANKS パラメータを指定して、その他の編集も指定できます。

このパラメータは、FORMATTED が有効な場合に限り有効です。

**SIGNED F**

データを符号付 10 進フォーマットで戻すように指定します。先行ゼロが除去され、負の整数に対しては負符号 (-) が付加されます。

有効な LENGTH は 2 バイトまたは 4 バイトのいずれかです。LENGTH が 1 バイトまたは 3 バイトの場合、2 バイトに変更されます。LENGTH が 4 バイトより大きい場合、4 バイトに変更されます。

このパラメータは、FORMATTED が有効な場合に限り有効です。

**UNSIGNED**

データを符号なし 10 進フォーマットで戻すように指定します。先行ゼロは除去されます。

有効な LENGTH は 1 バイトから 4 バイトです。LENGTH が 4 バイトより大きい場合、4 バイトに変更されます。

このパラメーターは、FORMATTED が有効な場合に限り有効です。

#### **REXX(...)**

ストレージを REXX 変数で戻すように指定します。このパラメーターは必須です。

#### **STORAGE(var\_name)**

ストレージを受け取る REXX 変数の名前を指定します。このパラメーターは必須です。

#### **FORMATTED**

ストレージをフォーマットして戻すように指定します。この値はデフォルトです。

#### **UNFORMATTED**

ストレージを 16 進のロー・フォーマットで戻すように指定します (ストレージ 1 バイトにつき 1 バイトが戻されます)。

#### **ENQUOTE**

戻される文字ストリングの先頭と末尾に 1 つずつ引用符を追加し、ストリング中のアポストロフィが対になるように指定します。

このパラメーターは、文字データが戻される場合に限り有効です。

#### **UNQUOTE NOQUOTES**

戻される文字ストリング中のアポストロフィ (X'7D') をすべてピリオドで置き換えるように指定します。

このパラメーターは、文字データが戻される場合に限り有効です。

#### **NOBLANKS**

戻される文字ストリング中のブランクをすべてピリオドで置き換えるように指定します。

このパラメーターは、文字データが戻される場合に限り有効です。

あるパラメーターが複数回指定された場合、最後に指定されたパラメーターのみが有効になります。

## 戻りコード

EVALUATE コマンドでは以下の戻りコードが使用されます。

**0**

ストレージが正常に取得されました。

**4**

要求されたデータ長が変更されました。説明が IDITRACE DDname に書き込まれます。

**12**

コマンド構文エラーまたはストレージが利用不可です。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 242. EVALUATE コマンドの例

```

/* REXX */

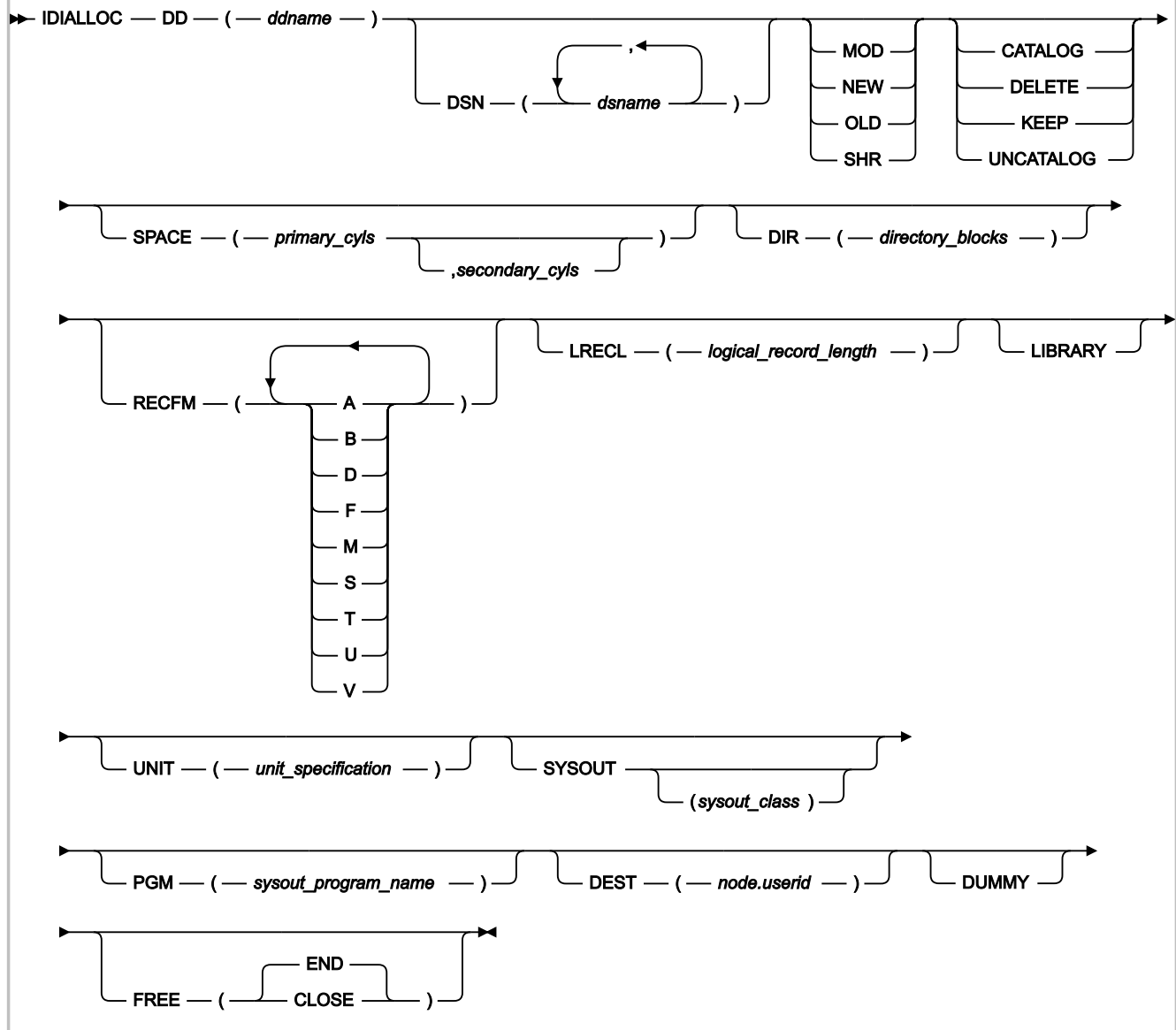
"EVALUATE 0 LENGTH(128) REXX(STORAGE(x))"
if RC = 0 then say 'Storage at address 0 =' x

```

## IDIALLOC コマンド

IDIALLOC コマンドは、DDname へのデータ・セットの動的割り振りと連結を行うために使用できます。

図 243. Syntax





**注:** コンマまたはブランク文字で反復値を区切ることはできません。例えば、固定ブロック・レコード・フォーマットを指定するには、RECFM(F B)ではなく、RECFM(FB)を使用してください。

オペランド:

**DD**

データ・セットに関連付けられる DDname。この ddname は常に必須です。

**DSN**

割り振られるデータ・セットの名前。複数のデータ・セット名を指定すると、最初のデータ・セットよりも後のデータ・セットはすべて既に存在すると見なされ、指定する DDname に連結されます。HFS パス名は許可されません。

**MOD**

データ・セットへの追加が行われます。

**NEW**

データ・セットが作成されます。

**OLD**

データ・セットが存在し、排他制御が必要です。

**SHR**

データ・セットが存在しますが、排他制御は必要ありません。

**CATALOG**

データ・セットがカタログされます。

**DELETE**

データ・セットが解放時に削除されます。

**KEEP**

データ・セットが解放時に保持されます。

**UNCATALOG**

データ・セットがアンカタログされます。

**SPACE**

シリンダー数としての 1 次スペースと増分。

**DIR**

必須ディレクトリー・ブロックの数。

**RECFM**

レコード・フォーマット:

**A**

ASA プリンター文字

**B**

ブロック化

**D**

可変長 ASCII レコード

**F**

固定

**M**

機械制御文字

**S**

標準ブロックまたはスパン

**T**

トラック・オーバーフロー

**U**

未定義

**V**

可変

**LRECL**

論理レコード長 (0 から 32760 までの値)。

**LIBRARY**

拡張区分データ・セット (partitioned data set extended: PDSE) を割り振ります。

**UNIT**

ファイルまたはデータ・セットが割り振られる装置タイプ。

**SYSOUT**

データ・セットがシステム出力データ・セットとなる。クラスはオプションで単一文字として指定できます。

**PGM**

SYSOUT プログラム名。

**DEST**

DEST ノードおよびユーザー ID。

ローカル宛先にはユーザー ID のみを使用できます。

## DUMMY

ダミー・データ・セットを割り振る。

## FREE

割り振り解除の指定:

### CLOSE

システムに対して、クローズ時にデータ・セットの割り振り解除を要求します。

### END

システムに対して、データ・セットを参照する最後のステップの終わりでデータ・セットの割り振り解除を要求します。この値はデフォルトです。

以下の構文規則が適用されます。

- DSN は以下のパラメーターとは一緒に使用できません。
  - PGM
  - DEST
- DEST は PGM パラメーターとは一緒に使用できません。
- SYSOUT は以下のパラメーターとは一緒に使用できません。
  - OLD
  - MOD
  - SHR
  - NEW
- 以下のパラメーターを指定する場合は、DSN パラメーターも指定する必要があります。
  - SPACE
  - DIR
  - UNIT
- 以下のパラメーターを指定する場合は、SYSOUT パラメーターも指定する必要があります。
  - PGM
  - DEST



**注:** このコマンドの場合、Fault Analyzer はデータ・セット名へのユーザー ID の自動プレフィックス変換を行いません。すべてのデータ・セット名を完全に修飾し、引用符なしで指定する必要があります。

## 戻りコード

IDIALLOC コマンドでは、以下の戻りコードが使用されます。

0

割り振りが成功しました。PDS または PDSE のメンバーが割り振られた場合、メンバーが存在し、読み取りのためにオープンできます。

1

割り振りが成功しました。ただし、読み取り用にオープンできない、PDS または PDSE の非既存メンバーが割り振られました。メンバーを読み取りのためにオープンした場合、システム異常終了 S013 が発生します。

4

割り振りが失敗しました。エラーの説明が IDITRACE DDname に書き込まれます。データ・セット (指定された最初のデータ・セット名以外) に関連する割り振りエラーまたは連結エラーによって IDIALLOC コマンドが停止することはありません。処理は続行します。

8

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 244. IDIALLOC コマンド例

```

/* REXX */

/* Allocate an existing data set to DDname DD1 */
"IDIALLOC DD(DD1) DSN(FRED.LISTING) SHR"
if RC = 0 then say 'Success!'

/* Allocate a temporary sequential work data set to DDname DD2 */
"IDIALLOC DD(DD2) DSN(FRED.SEQ) NEW DELETE SPACE(2,3) UNIT(SYSALLDA) ",
"RECFM(FB) LRECL(80)"

/* Allocate a new partitioned data set to DDname DD3 */
"IDIALLOC DD(DD3) DSN(FRED.PDS) NEW CATALOG SPACE(2) UNIT(SYSALLDA) ",
"DIR(5) RECFM(VBA) LRECL(137)"

/* Allocate default JES spool data set to DDname DD4 */
"IDIALLOC DD(DD4) SYSOUT"

/* Allocate internal reader for submission of job to DDname DD5 */
"IDIALLOC DD(DD5) SYSOUT PGM(INTRDR)"
if rc = 0 then do /* Check rdr allocation */
----processing----
end
else "IDIWTO INTRDR FAILED ALLOCATION"

/* Allocate a list of load libraries to the IDIRLOAD DDname */
"IDIALLOC DD(IDIRLOAD) DSN(MY.LOADLIB1 MY.LOADLIB2 MY.LOADLIB3) SHR"

/* Allocate a new PDSE history file data set named MY.HIST */
"IDIALLOC DD(DD6) DSN(MY.HIST) NEW CATALOG LIBRARY,
RECFM(VB) LRECL(10000) SPACE(20,20)"

```

## IDIDDTTEST コマンド

IDIDDTTEST コマンドは、DDname が割り振られているかどうかを検査するために使用できます。

図 245. Syntax

```
▶ IDIDDTEST — DD — ( — ddname — ) ▶
```

## 戻りコード

IDIDDTEST コマンドでは以下の戻りコードが使用されます。

0

指定された DDname が割り振られています。

4

指定された DDname が割り振られていません。

8

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 246. IDIDDTEST コマンド例

```
/* REXX */

/* Test if the DDname DD1 is allocated */
"IDIDDTEST DD(DD1)"
if RC = 0 then say 'DD1 is allocated'
```

## IDIDSECTdsn コマンド

IDIDSECTdsn コマンドを使用すると、IDIDSECTデータ・セット連結 (この DDname の詳細については、[DataSets ページ 576](#)を参照) に対する照会または変更を行い、製品 (CICS® など) の所定のバージョンに正しい DSECT マッピングを確実に使用することができます。

図 247. Syntax

```
▶ IDIDSECTDSN — GET — ( — var_name — ) ▶
                   SET — ( — var_name — ) ▶
```

各コマンドの説明。

### GET(*var\_name*)

現行の IDIDSECT データ・セット連結を REXX 変数 *var\_name* に戻すように指定します。戻されたリスト内のデータ・セット名は、空白で区切られます。

### SET(*var\_name*)

現行 IDIDSECT 連結を置き換えるデータ・セット名リストを含む REXX 変数の名前を指定します。複数のデータ・セット名は、1 つ以上の空白によって区切られている必要があります。



## 戻りコード

IDIDSECTdsn コマンドでは、以下の戻りコードが使用されます。

0

正常終了。

4

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 248. IDIDSECTdsn コマンド例

```
/* REXX */

/* Place MY.DSECTS data set first in the IDIDSECT concatenation */
"IDIDSECTdsn GET(dsnlist)"
if RC = 0 then say 'Current IDIDSECT concatenation:' dsnlist
else exit 4
dsnlist = 'MY.DSECTS' dsnlist
"IDIDSECTdsn SET(dsnlist)"
if RC = 0 then say 'IDIDSECT concatenation changed to:' dsnlist
```

## IDIDSNTTEST コマンド

IDIDSNTTEST コマンドは、データ・セットが存在するかどうかを検査するために使用できます。

図 249. Syntax

```
►► IDIDSNTTEST — DSN — ( — dsname — ) —◄◄
```

### パラメーター

**dsname**

存在の確認テストの対象となるデータ・セットの完全修飾名を指定します。データ・セット名は一重引用符で囲まないようにしてください。

### 戻りコード

IDIDSNTTEST コマンドでは、以下の戻りコードが使用されます。

0

指定されたデータ・セットは存在します。

4

指定されたデータ・セットは存在しません。

8

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

**例****例**

図 250. IDIDSNTTEST コマンド例

```

/* REXX */

/* Test if the data set MY.HIST is allocated */
"IDIDSNTTEST DSN(MY.HIST)"
if RC = 0 then say 'MY.HIST exists'

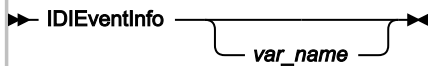
```

**IDIEventInfo コマンド**

IDIEventInfo コマンドは、フォーマット・ユーザー出口で、現在の障害にあるすべてのイベントに関する情報を取得するために使用できます。

UFM データ域には PSW やレジスターなどの 1 セットのフィールドのみが含まれるため、IDIEventInfo コマンドを使用して、UFM データ域の値を、あるイベントを表す値から別のイベントを表す値に変更する必要があります。

図 251. Syntax

**パラメーター****var\_name**

情報を検索するイベントのイベント番号を含む REXX 変数の名前を指定します。

必要なイベント番号を含む変数が *var\_name* に指定されていない場合、代わりに UFM.EVENT\_NO が使用されます。

検索された情報は、UFM データ域に配置されます。

**戻りコード**

IDIEventInfo コマンドでは、以下の戻りコードが使用されます。

**0**

情報が正常に取得されました。

**4**

指定されたイベント番号に該当する情報は存在しません。エラーの説明が IDITRACE DDname に書き込まれます。

**8**

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 252. IDIEventInfo コマンド例

```
/* REXX */

/* Process all events */
do event=1 to UFM.NUM_EVENTS
  "IDIEventInfo event"
  if RC<>0 then iterate
  :
end
```

## IDIFREE コマンド

IDIFREE コマンドは、IDIALLOC を使用して割り振られた可能性のある DDname の割り振りを解除 (解放) するために使用できます。

図 253. Syntax

```
→ IDIFREE — DD — ( — ddname — ) →
```

## 戻りコード

IDIFREE コマンドでは以下の戻りコードが使用されます。

0

指定されたすべての DDname の割り振り解除が成功しました。

4

1 つ以上の指定の DDname の割り振り解除が失敗しました。エラーの説明が IDITRACE DDname に書き込まれます。

8

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 254. IDIFREE コマンド例

```
/* REXX */

/* Free the DDnames DD1 and DD2 */
"IDIFREE DD(DD1,DD2)"
if RC = 0 then say 'Success!'
```

## IDIGET コマンド

再分析処理中に、IDIGET コマンドを使用して、REXX ステム変数の内容を現在の障害項目から取得できます。これは、例えばリアルタイム処理中に IDIPUT コマンドを使用してフォーマット・ユーザー出口が情報を保存した場合などに便利です。

図 255. IDIGET command syntax

▶ IDIGET — *stem\_name* ▶

### パラメーター

#### *stem\_name*

REXX ステム変数の名前を指定します。ステム変数の名前は、IDIPUT で使用されている名前と同じでなければなりません。

### 戻りコード

IDIGET コマンドでは、以下の戻りコードが使用されます。

0

正常終了。

4

エラーが発生しました。

### 例

```
/* REXX */

IDIGET xyz
if rc = 0 then do i = 1 to xyz.0
    say xyz.i
end
exit 0
```

## IDIModQry コマンド

IDIModQry コマンドは、指定されたロード・モジュールに関する情報を Fault Analyzer REXX ユーザー出口から取得する場合に使用します。このコマンドはフォーマット・ユーザー出口でのみ使用できます。

図 256. Syntax

▶ IDIMODQRY — NAME — ( — *module\_name* — ) — LP — ( — *var\_name* — ) — MODLEN — ( — *var\_name* — ) ▶

各コマンドの説明。

#### NAME(*module\_name*)

ロード・ポイントと、オプションでモジュール長を要求する対象となるモジュールの名前。

**LP(*var\_name*)**

文字フォーマットのモジュール・ロード・ポイントを保持する REXX 変数の名前。

**MODLEN(*var\_name*)**

文字フォーマットの 16 進モジュール長を保持する REXX 変数の名前。この値はオプションです。

**戻りコード**

IDIModQry コマンドでは、以下の戻りコードが使用されます。

0

要求されたモジュールは検出されました。

4

要求されたモジュールは検出されませんでした。

12

構文エラーです。

**例**

図 257. IDIModQry コマンド例

```
/* REXX */
"IDIMODQRY NAME("IDISCBL1") LP(addr) MODLEN(len)"
if rc = 0 then "note 'Module IDISCBL1 at address" addr "length" len'"
```

**IDIPUT コマンド**

IDIPUT コマンドを使用して、現行の障害項目に REXX ステム変数の内容を保存することができます。これは、例えば、Formatting ユーザー出口が、デューティー・システム・オペレーターなど、他の方法では Fault Analyzer によってキャプチャーされない異常終了に関連する情報を取得する場合に役立つ可能性があります。

図 258. IDIPUT command syntax

▶ IDIPUT — *stem\_name* ▶

**パラメーター*****stem\_name***

REXX ステム変数の名前を指定します。

**戻りコード**

IDIGET コマンドでは、以下の戻りコードが使用されます。

0

正常終了。

## 4

エラーが発生しました。

## 例

```
/* REXX */
mydata.0 = 1
mydata.1 = "Duty operator: Fred"
IDIPUT mydata
exit 0
```

## IDRegisterFaultEntry コマンド

IDRegisterFaultEntry コマンドを使用すると、MVS™ ダンプ・データ・セット (CICS® システム・ダンプなど) の分析のときに、いつでも障害項目を登録できます。このコマンドにより、ヒストリー・ファイル内の障害項目を早く作成でき、最初に対話式再分析を終了する必要がありません。インストール済み環境で、例えば分析制御ユーザー出口などから IDRegisterFaultEntry コマンドを発行することにより、登録障害項目の作成を自動化するように選択することも、対話式再分析のときに、ユーザーがオンデマンドでフォーマット・ユーザー出口を呼び出して、このコマンドを発行するように選択することもできます。



## 注:

1. 不要な障害項目がさらに作成されるおそれがあるため、このコマンドは、ダンプ登録の分析制御ユーザー出口または通知ユーザー出口で障害項目を登録するために使用しないでください。IDIXTSEL ダンプ登録処理では、ユーザー出口を使用しなくても自動的に障害項目を作成します。
2. IDRegisterFaultEntry コマンドを使用する代わりに GenerateSavedReport オプションを使用すると、MVS™ ダンプをバッチで分析するために、障害項目を現行のヒストリー・ファイルに作成できます。詳しくは、[GenerateSavedReport ページ 606](#)を参照してください。

図 259. Syntax

```
→ IDRegisterFaultEntry — ( — history_file_dsn — ) →
```

## 説明:

**IDIHIST(history\_file\_dsn)**

登録障害項目の作成場所となるヒストリー・ファイルを指定します。対話式再分析においては、このパラメーターを指定しなかった場合、またはユーザーが指定されたヒストリー・ファイルに対する UPDATE アクセス権を持っていない場合は、使用するヒストリー・ファイルを指定できるプロンプトが表示されます。バッチ再分析の場合、このパラメーターの指定は必須です。指定しない場合は、RC=4 が出されます。

## 戻りコード

IDIRegisterFaultEntry コマンドでは、以下の戻りコードが使用されます。

0

正常終了。

4

障害項目が既に存在するか、ユーザーが対話式プロンプトを介して要求をキャンセルしました。

12

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 260. IDIRegisterFaultEntry コマンド例

```
/* REXX */

/* Create registration fault entry in history file MY.HIST */
ENV.USER_TITLE = 'My fault!'
ENV.USER_NAME   = UserID()
ENV.LOCK_FLAG   = '/'
dsn = 'my.hist'
"IDIRegisterFaultEntry IDIHIST("dsn")"
if rc <> 0 then
  "IDIWTO IDIRegisterFaultEntry failed, rc="rc
exit 0
```

IDIRegisterFaultEntry コマンドの別の使用例については、[サンプル・フォーマット・ユーザー出口 ページ 494](#)を参照してください。

## IDIWRITE コマンド

IDIWRITE コマンドは、ユーザー出口から Fault Analyzer にデータ・レコードを渡すために使用します。これは、コンパイラー・リスト読み取り、メッセージ、異常終了コードの説明、フォーマット、または通知ユーザー出口でのみ使用できます。IDIWRITE コマンドによって提供可能なデータのタイプは、このコマンドの使用元であるユーザー出口のタイプによって異なります。詳細については、それぞれのユーザー出口タイプに関する一般セクションを参照してください。

図 261. Syntax

```
IDIWRITE var_name 'text'
```

各コマンドの説明。

### 'text'

印刷するテキストをアポストロフィで囲んで指定します。ストリングを囲むには、単一引用符 (')

または二重引用符 (")

のいずれかを使用できます。このとき最初の引用符と最後の引用符には、同じ文字を使用してください。テキ

スト中の引用符がストリングを囲むために使用された引用符と同じ場合、いずれの引用符も 2 回指定する必要があります。例えば、*It isn't there* を出力するには、テキストを `"It isn't there"` または `'It isn't there'` で指定します。

引用符付きストリングを使用すると、通知ユーザー出口またはフォーマット・ユーザー出口でのみサポートされます。

#### **var\_name**

データ・レコードを含む変数の名前。

*var\_name* を指定せずに IDIWRITE コマンドを使用する場合、関連する出口タイプで説明したように出口固有のデータ域を使用してデータ・レコードを渡す必要があります。

## 戻りコード

IDIWRITE コマンドでは、以下の戻りコードが使用されます。

0

レコードが正常に書き込まれました。

2

直前のエラーによりレコードの書き込みが使用不可になりました。

4

エラーのためレコードが書き込まれませんでした。エラーの説明が IDITRACE DDname に書き込まれます。

8

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。



## 例

図 262. IDIWRITE コマンド例

```

/* REXX */
/* Pass records to Fault Analyzer */

/* Method 1 - plain text */
"IDIWRITE 'This is record 1'"
if RC = 0 then say 'Method 1 success!'

/* Method 2 - using LST data area (Compiler Listing Read user exit) */
rec = 'This is record 2'
LST.DATA_LENGTH = length(rec)
LST.DATA_BUFFER = rec
"IDIWRITE"
if RC = 0 then say 'Method 2 success!'

/* Method 3 - letting REXX resolve data record variable */
rec = 'This is record 3'
"IDIWRITE '"rec'"
if RC = 0 then say 'Method 3 success!'

/* Method 4 - letting Fault Analyzer resolve data record variable */
rec = 'This is record 4'
"IDIWRITE rec"
if RC = 0 then say 'Method 4 success!'

```

## IDIWTO コマンド

IDIWTO コマンドは、メッセージを MVS™ コンソールに書き込む場合に使用します。このコマンドは、例えば、トレースがアクティブでない (IDITRACE DDname が割り振られていない) ときには必ず、REXX SAY コマンドの代わりに使用できません。

図 263. Syntax

```
►► IDIWTO — message_text ◄◄
```

メッセージ・テキスト内で改行文字 (X'15') を使用して、長メッセージを複数の WTO に分割することができます。それ以外のすべての印刷不能文字は、ピリオドに変更されます。

## 戻りコード

IDIWTO コマンドは常に RC=0 で完了します。

## 例

図 264. IDIWTO コマンド例

```
/* REXX */

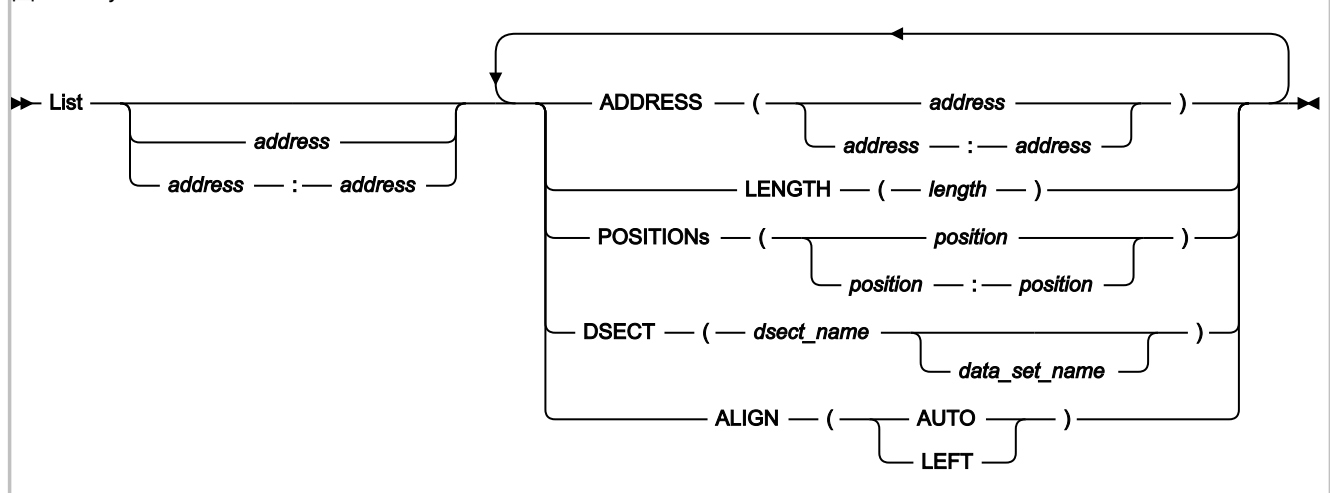
/* Write a message to the MVS console */
"IDIWTO Minutes since last duplicate fault =" EPC.MINUTES_SINCE_LAST_DUP
```

混乱を避けるために、発行される WTO メッセージには、Fault Analyzer によって発行される正式なメッセージのいずれか間違える可能性があるメッセージ ID を組み込まないことをお勧めします。メッセージ ID に接頭部 “IDI” を付ける場合は、Fault Analyzer メッセージと一致する可能性を排除するために、追加文字を付加できます。例えば、数字の前に “IDIX” を付けることができます。

## LIST コマンド

LIST コマンドは、フォーマット・ユーザー出口で、分析後の障害環境からストレージ域を印刷するために使用できます。

図 265. Syntax



## パラメーター

### アドレス 開始:終了

定位置パラメーターとして、開始アドレスまたはアドレス範囲のいずれかを指定します。このアドレスは、ADDRESS キーワードの使用に代わる方法です。有効な構文の詳細については、次の ADDRESS を参照してください。

### ADDRESS(アドレス) ADDRESS(アドレス:アドレス)

印刷されるストレージのアドレスを指定します。単一のアドレスまたはアドレス範囲のいずれかを指定します。

address パラメーターは 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。アドレスには、オプションでその後にはピリオドを追加できます (例えば [000176C0.] )。

**LENGTH(長さ)**

印刷されるバイト数を指定します。16 進数の値 (インディケータを 'X...' で付加) または 10 進数の値 (インディケータは不要) のいずれかで指定します。アドレスの範囲が指定されている場合、LENGTH の指定は無視されます。

**POSITIONs(位置) POSITIONs(位置:位置)**

開始アドレスから、印刷されるストレージの最初のバイトまでのオフセットを指定します。単一のオフセットまたはオフセット範囲のいずれかで指定します。POSITION と POSITIONs の両方が有効です。

オフセットはすべて、16 進数 (インディケータを 'X...' で付加) または符号付 10 進数 (オプションで、インディケータを 'F...' で付加) のいずれかで指定する必要があります。

**DSECT(dsect\_name) DSECT(dsect\_name data\_set\_name)**

要求されたアドレスでストレージをフォーマットする際に使用される DSECT マッピング・メンバーの名前を指定します。

データ・セット名を指定しない場合、IDIDSECT DDname 名で DSECT が使用可能である必要があります (詳細については、[DataSets ページ 576](#)を参照してください)。または、DSECT が指定されたデータ・セット名の中に存在する必要があります。

**ALIGN(AUTO | LEFT)**

16 進データの位置合わせを指定します。

**AUTO**

位置合わせは、リストされたアドレスと使用可能なレポート幅によって決まります。これはデフォルトです。

**LEFT**

16 進データは無条件に左寄せされます。

あるパラメーターが複数回指定された場合、最後に指定されたパラメーターのみが有効になります。

DSECT パラメーターが指定されなかった場合、ストレージは (実行中の障害分析タイプで優先されるフォーマット幅に依存して) 1 行 16 バイトまたは 32 バイトでフォーマットされ、16 進と EBCDIC の両方の値が表示されます。

DSECT パラメーターが指定された場合、ストレージは DSECT コマンドが使用されたときと同様にフォーマットされて表示されます。詳しくは、[DSECT 情報を使用したストレージ域のマッピング ページ 227](#)を参照してください。

**戻りコード**

LIST コマンドでは以下の戻りコードが使用されます。

**0**

コマンドは正常に終了しました。

**4**

DSECT フォーマット処理中にエラーが発生しました。エラーの説明が IDITRACE DDname に書き込まれます。

12

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

例

図 266. LIST コマンドの例

```
/* REXX */
"LIST 0 LENGTH(128)"
```

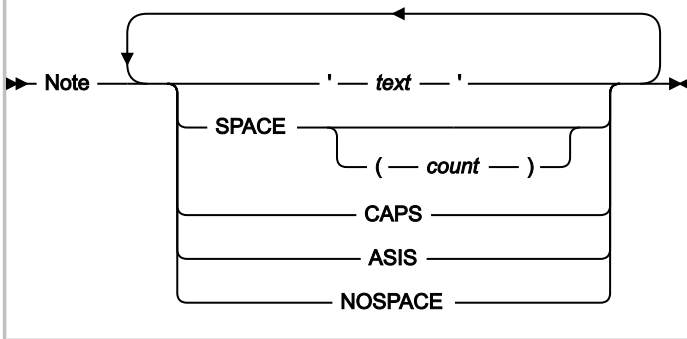
上記の例によって、下記のような出力が作成されます。

Address	Offset	Hex	EBCDIC
00000000		040C0000 810692C8 00000000 00000000	*...a.kH.....*
00000010	+10	00FC7F08 00000000 070C1000 85532492	*.."......e..k*
00000020	+20	078D0000 00FC7F5A 078D1000 8775BB42	*....."!....g...*
00000030	+30	00000000 00000000 070C0000 85532496	*.....e..o*
00000040	+40	00000000 00000000 00000000 00FC7F08	*.....".*
00000050	+50	00000000 00000000 040C0000 810676D0	*.....a..}* *.....`.....a...*
00000060	+60	040C0000 80FFB080 00080000 BF286880	*.....*
00000070	+70	00080000 BF287940 040C0000 81068A00	*.....`.....a...*

NOTE コマンド

NOTE コマンドは、フォーマット・ユーザー出口で、1 行のテキストを印刷するために使用できます。

図 267. Syntax



パラメーター

'テキスト'

印刷するテキストをアポストロフィで囲んで指定します。ストリングを囲むには、単一引用符 (') または二重引用符 (") のいずれかを使用できます。このとき最初の引用符と最後の引用符には、同じ文字を使用してください。

テキスト中の引用符がストリングを囲むために使用された引用符と同じ場合、いずれの引用符も 2 回指定する必要があります。例えば、次のテキストを印刷する場合

```
It isn't there
```

テキストを次のいずれかのように指定します。

```
"It isn't there"
```

または

```
'It isn''t there'
```

テキストを指定しない場合、データは印刷されません。ただし、SPACE パラメーターを使用してブランク行を指定した場合は、常に書き込まれます。

### SPACE SPACE(カウント)

次のテキスト行までに書き込まれるブランク行の行数を指定します。

SPACE パラメーターを *count* なしで指定した場合、デフォルトの 1 に指定されます。

### CAPS

テキストがすべて大文字で書き込まれるように指定します。

### ASIS

テキストを大文字にせず記述どおりに「そのまま」書き込むように指定します。

### NOSPACE

次のテキスト行までにブランク行が書き込まれないように指定します。この値はデフォルトです。

あるパラメーターが複数回指定された場合、最後に指定されたパラメーターのみが有効になります。

## 戻りコード

NOTE コマンドでは、以下の戻りコードが使用されます。

0

コマンドは正常に終了しました。

12

コマンド構文エラーです。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

図 268. NOTE コマンドの例

```
/* REXX */

"NOTE 'This is a simple note.'"
"NOTE 'This note follows the previous without any blank lines inserted.'"
"NOTE 'This note has 2 blank lines ahead of it.' SPACE(2)"
```

上記の例により、以下出力されます。

```
This is a simple note.
This note follows the previous without any blank lines inserted.
```

This note has 2 blank lines ahead of it.

## フォーマット・タグ

レポート用にデータをフォーマットする際、フォーマット・ユーザー出口で使用できるタグを以下で説明します。これらのタグにより、HTML に類似した構文を使用して表示データに見出し、リストなどを作成できます。タグ・ストリームを Fault Analyzer に戻すには、フォーマット・ユーザー出口から IDIWRITE コマンドを使用します。

フォーマット・タグの使用を示している以下の例は、データ・セット IDI.SIDISAM1 内のメンバー IDISUFM3 としてソフトコピー・フォーマットでも提供されています。

図 269. REXX フォーマット・ユーザー出口 3 ソースの例

```
"IDIWRITE '<P>First paragraph.'"
"IDIWRITE '<AREA INDENT=5>'"
"IDIWRITE '<P>Second paragraph, indented 5 characters from the first. '"
"IDIWRITE 'This <DATA 3><P> tag is treated as text only.'"
"IDIWRITE '<P COMPACT>Third paragraph. '"
"IDIWRITE 'Note that this paragraph is not preceded by a blank line.'"
"IDIWRITE '</AREA>'"
"IDIWRITE '<P>Fourth paragraph - now we are back at the left margin.</P>'"
"IDIWRITE '<L>***** This line will '"
"IDIWRITE '<HP>not</HP> wrap at the preferred formatting width!'"
"IDIWRITE '<P><ADDR 625f22>Previous area</ADDR> and <ADDR 625f22></ADDR> are '"
"IDIWRITE 'both point-and-shoot fields to the Dump Storage '"
"IDIWRITE 'display for address 00625F22 in the interactive reanalysis report.'"
"IDIWRITE '<DL BREAK=STDLBL>'"
"IDIWRITE '<DT>This is a long definition term'"
"IDIWRITE '<DD>This is the matching definition description which might wrap '"
"IDIWRITE 'depending on the preferred formatting width.'"
"IDIWRITE '<DT>A shorter definition term'"
"IDIWRITE '<DD>The definition description of the second term.'"
"IDIWRITE '</DL>'"
"IDIWRITE '<P><DUMP 0 20>Address 0 storage for a length of 32 bytes:</DUMP>'"
"IDIWRITE '<UL>'"
"IDIWRITE '<LI>In an unordered list, each item is preceded by a bullet. '"
"IDIWRITE 'If necessary, the item description will wrap at the '"
"IDIWRITE 'preferred formatting width.'"
"IDIWRITE '<LI>Another item in the same list.'"
"IDIWRITE '</UL>'"
"IDIWRITE '<P><NOTEL>'"
"IDIWRITE '<LI>In a note list, each note is numbered and the list is '"
"IDIWRITE 'preceded by a ""Notes:"" heading. If necessary, the note '"
"IDIWRITE 'description will wrap at the preferred formatting width.'"
"IDIWRITE '<LI>Another note in the same list.'"
"IDIWRITE '</NOTEL>'"
"IDIWRITE '<P><TH>Column Column</TH>'"
"IDIWRITE '<L><U>1    <U>2    </U>'"
"IDIWRITE '<L> 123    17'"
exit 0
```

上記のフォーマット結果は次のようになります (ポイント・アンド・シュート・フィールドと強調表示テキストが Bold スタイルで表示されます)。

## 図 270. REXX フォーマット・ユーザー出口 3 出力の例

```

File View Services Help
-----
Interactive Reanalysis Options                               Line 1 Col 1 80
Command ==>                                                Scroll ==> CSR
JOBNAME: P35777      SYSTEM ABEND: 0C7                      FAE1      2019/10/31  22:51:13

First paragraph.

    Second paragraph, indented 5 characters from the first. This <P> tag is
    treated as text only.
    Third paragraph. Note that this paragraph is not preceded by a blank
    line.

Fourth paragraph - now we are back at the left margin.
***** This line will not wrap at the prefer

Previous area and 00625F22 are both point-and-shoot fields to the Dump Storage
display for address 00625F22 in the interactive reanalysis report.

This is a long definition
term. . . . . : This is the matching definition description
                which might wrap depending on the preferred
                formatting width.

A shorter definition term . : The definition description of the second term.

Address 0 storage for a length of 32 bytes:
Address  Offset      Hex                               EBCDIC / ASCII
00000000          040C0000 810692C8 00000000 00000000 *....a.kH.....*
00000010          +10  00FC7F08 00000000 070E0000 00000000 *.."......*

```

以下は、フォーマット・タグの一般的な規則です。

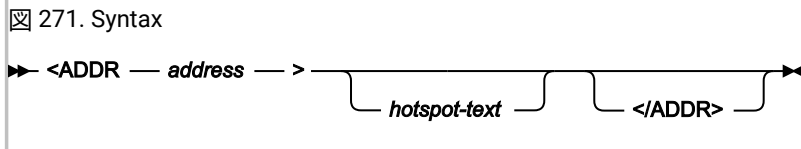
- パラグラフの行の先頭と末尾、および定義記述の先頭と末尾 (<DD> タグが前に付加されたテキスト) を除く、すべての空白が有効です。
- 先行タグが指定されておらず、空白文字を含むテキストでは、テキストの前に <P> タグが暗黙的に挿入されません。
- すべてのタグと属性は、大/小文字の区別がありません。
- 行の最大幅は、すべての出力において 132 文字です。最大幅を超えるテキストはラップされます。

各タグの詳細を以下に説明します。

## ADDR (アドレス)

ADDR タグでアドレスのポイント・アンド・シュート・フィールドを定義します。

アドレス・フィールドが Interactive Reanalysis Report に表示されている場合、カーソルをアドレス・フィールド上に置き Enter を押すと、指定されたアドレスの Dump Storage 画面が表示されます。



### address

ポイント・アンド・シュート・フィールドが選択されている場合に表示される 16 進アドレス。

address パラメーターは 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。

### hotspot-text

表示対象のポイント・アンド・シュート・フィールド。指定されていない場合、アドレスが使用されます。

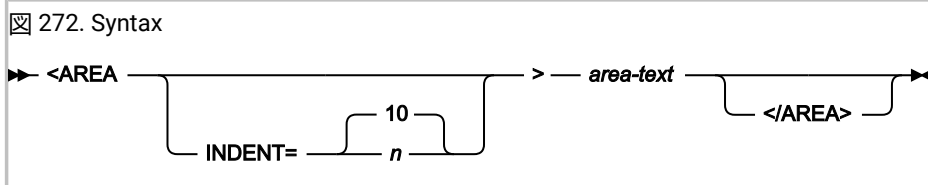
## 説明

ADDR タグによる改行は行われません。

Interactive Reanalysis Report では、ポイント・アンド・シュート・フィールドは黄色で示されます。

## AREA (領域)

AREA タグでは、表示のセクションを定義します。このタグを使用して、高水準タグ規定用に提供されているデフォルトのインデントを超えて、テキストのインデントを制御します。



### INDENT

この属性で、現行のインデントを増分する文字数を指定します。

### area-text

表示セクション用のタグとテキストです。

## 説明

AREA タグは他の AREA タグ内でネストさせることができます。その他のタグは、いずれも暗黙的に終了されます。



## DD (定義説明)

DD タグでは、定義リストの用語の記述を定義します。

図 273. Syntax



### description-text

この値は用語の説明です。

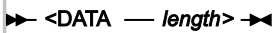
## 説明

[DL \(定義リスト\) ページ 545](#)を参照してください。

## DATA (データ)

DATA タグでは、このタグ以降にあるタグ処理の対象外とする文字の数を定義します。このタグは通常、誤ってフォーマット・タグと解釈される可能性のある文字が書き込まれるテキストに含まれている場合に限り使用します。

図 274. Syntax



### length

入力ストリームで DATA

タグの直後から始まり、何らかのタグが含まれている可能性がある場合でもテキスト・データとして扱う必要のある文字の数です。

## 説明

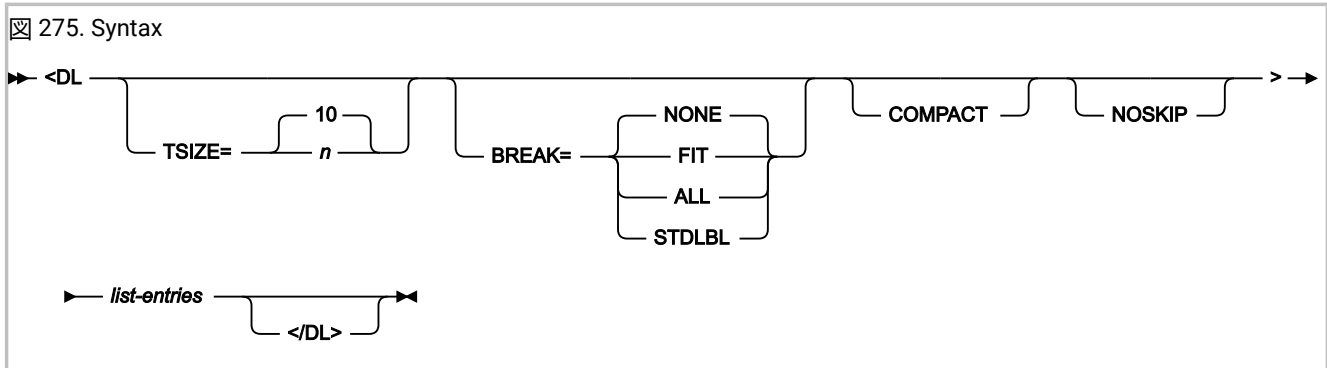
有効なタグが含まれている可能性のあるテキストでフォーマット上の問題が発生しないよう、DATA タグを使用してテキストとしてのみ扱う必要のある文字の数を指定できます。



**注:** 有効なタグ構文とならない場合に限り、タグ区切り文字 (「<」または「>」) を DATA タグが前に付加されないテキストで使用できます。無効なタグおよび未完了のタグは、テキスト・データとして入力ストリームに戻されます。

## DL (定義リスト)

DL タグでは、用語と定義のリストを定義します。

**TSIZE**

この属性で、定義記述のインデントを指定します。最小値は 3 文字、デフォルト値は 10 文字です。

**BREAK**

この属性で、以下のように定義用語と記述のフォーマットを制御します。

- BREAK=NONE を指定すると、用語は説明と同じ行に表示され、長さが TSIZE を超える場合は説明エリアに文字あふれします。この値はデフォルトです。
- BREAK=FIT を指定すると、用語が TSIZE を超える場合に説明が用語の下の行に表示されます。
- BREAK=ALL を指定すると、各定義は用語の下の行に表示されます。
- BREAK=STDLBL を指定すると、標準の Fault Analyzer ラベルが使用されます。用語の長さが有効な TSIZE を超える場合、テキストは TSIZE の幅にラップされます。最後の用語の行の末尾にピリオドまたはコロンが追加されます。

**COMPACT**

この属性を指定すると、リストの項目間に空白行を作成せずにリストがフォーマットされます。

**NOSKIP**

この属性を指定すると、リストの先頭に空白行を作成せずにリストがフォーマットされます。

**list-entries**

リスト項目を作成する DT タグおよび DD タグです。

**説明**

Interactive Reanalysis Report では、リスト項目は白色で示されます。ただし STDLBL 属性が使用される場合、定義用語は緑色になります。

**DT (定義語)**

DT タグでは、定義リストの用語を定義します。



**term-description**

この値は DD タグ経由で指定されます。

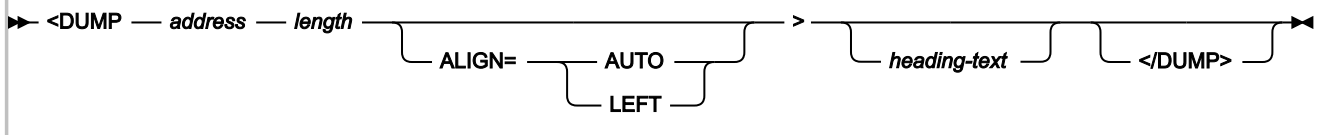
**説明**

DL (定義リスト) ページ 545を参照してください。

**DUMP (EBCDIC ダンプ)**

DUMP タグでは、アドレス自体のみを表示する ADDR タグとは対照的に、フォーマット済みの出力に 16 進ダンプがインラインで挿入されます。この挿入は特に、ユーザーが他の方法ではストレージを参照できないために、バッチ・レポートをフォーマット設定する場合に役立ちます。

図 277. Syntax

**address**

表示対象のストレージ域の 16 進フォーマットのアドレスです。

address パラメーターは 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。

**length**

表示対象のストレージ域の 16 進フォーマットの長さ (バイト) です。

**ALIGN = AUTO | LEFT**

16 進データの位置合わせを指定します。

**AUTO**

位置合わせは、リストされたアドレスと使用可能なレポート幅によって決まります。これは、デフォルトです。

**LEFT**

16 進データは無条件に左寄せされます。

**heading-text**

16 進ダンプ表示の直前の見出しテキストです。

**説明**

16 進ダンプの表示は、現行のインデントにかかわらず、常に 1 列目から開始されます。

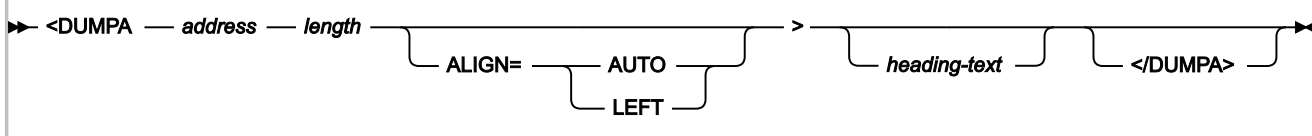
画面の右側にある文字で表現されるセクションは、EBCDIC でエンコードされた 16 進値に基づいています。データに ASCII でエンコードされた値が入っていることが分かっている場合は、代わりに DUMPA タグ ([DUMPA \(ASCII ダンプ\) ページ 548](#)を参照) を使用してください。

Interactive Reanalysis Report では、見出しテキストは白色で示されます。

## DUMPA (ASCII ダンプ)

DUMPA タグでは、アドレス自体のみを表示する ADDR タグとは対照的に、フォーマット済みの出力に 16 進ダンプがインライン表示で挿入されます。この挿入は特に、ユーザーが他の方法ではストレージを参照できないために、バッチ・レポートをフォーマット設定する場合に役立ちます。

図 278. Syntax



### address

表示対象のストレージ域の 16 進フォーマットのアドレスです。

address パラメーターは 64 ビット対応です。詳しくは、[64 ビット・アドレスの指定 ページ 147](#)を参照してください。

### length

表示対象のストレージ域の 16 進フォーマットの長さ (バイト) です。

### ALIGN = AUTO | LEFT

16 進データの位置合わせを指定します。

#### AUTO

位置合わせは、リストされたアドレスと使用可能なレポート幅によって決まります。これは、デフォルトです。

#### LEFT

16 進データは無条件に左寄せされます。

### heading-text

16 進ダンプ表示の直前の見出しテキストです。

## 説明

16 進ダンプの表示は、現行のインデントにかかわらず、常に 1 列目から開始されます。

画面の右側にある文字で表現されるセクションは、ASCII でエンコードされた 16 進値に基づいています。データに EBCDIC でエンコードされた値が入っていることが分かっている場合は、代わりに DUMP タグ ([DUMP \(EBCDIC ダンプ\) ページ 547](#)を参照) を使用してください。

Interactive Reanalysis Report では、見出しテキストは白色で示されます。

## HP (強調表示されたフレーズ)

HP タグでは、強調表示させるテキストを指定します。


 **注:** 強調表示は、Interactive Reanalysis Report でのみ使用可能です。

図 279. Syntax



### **text-to-be-highlighted**

このテキストが強調表示されます。

## 説明


このタグによる改行は行われません。

Interactive Reanalysis Report では、このテキストは青緑色で示されます。

## L (行)

L タグでは、ユーザーが設定した優先表示幅に従属しない行テキストを定義します。

図 280. Syntax



### **line-text**

行テキスト。

## 説明

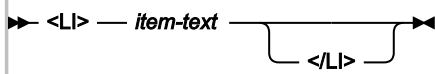
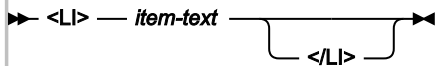
各行は、次の行にあふれたテキストをインデントせずに、フォーマットされます。行テキストの前後に空白行は追加されません。

行の幅は、画面の行の最大幅により指定されます。この制限を超える行は次の行にあふれるため、この幅を超えて表示されることはありません。

Interactive Reanalysis Report では、行テキストは白色で示されます。

## LI (リスト項目)

LI タグでは、Note List または Unordered List 中のリスト項目を定義します。


 281. Syntax
**item-text**


項目テキスト。

**説明**

Interactive Reanalysis Report では、行テキストは白色で示されます。

**NOTEL (注釈リスト)**

NOTEL タグでは、注釈のリストを定義します。


 282. Syntax
**COMPACT**

この属性を指定すると、リスト項目間に空白行を作成せずにリストがフォーマットされます。

**NOSKIP**

この属性を指定すると、リストの先頭に空白行を作成せずにリストがフォーマットされます。

**list-entries**

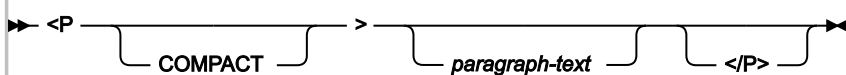
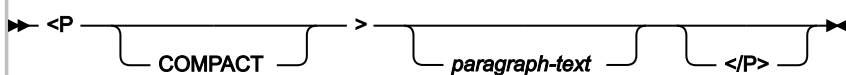
LI タグを使用して指定されます。

**説明**

このタグでは改行が行われます。

**P (パラグラフ)**

P タグでは、テキストのパラグラフを定義します。


 283. Syntax
**COMPACT**

この属性を指定すると、パラグラフの前に空白行を作成せずにパラグラフがフォーマットされます。

**paragraph-text**

パラグラフのテキスト。

## 説明

各パラグラフは、次のブロックにあふれた一連のテキストをインデントせずに、フォーマットされます。COMPACT 属性が指定されていない場合、空白行がパラグラフの前に追加されます。

表示されるパラグラフの幅は、「View」プルダウン・メニューからユーザーが指定した優先フォーマット幅により指定されます。

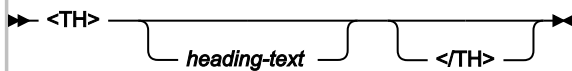
リスト中のパラグラフは、リスト項目のテキストに合わせて調整されます。

Interactive Reanalysis Report では、パラグラフ・テキストは白色で示されます。

## TH (表の見出し)

TH タグでは、表の見出し行を定義します。

図 284. Syntax



### heading-text

表の見出し行。

## 説明

各テーブルの見出し行は、次の行にあふれたテキストをインデントせずに、フォーマットされます。行テキストの前後に空白行は追加されません。

行の幅は、画面の行の最大幅により指定されます。この制限を超える行は次の行にあふれるため、この幅を超えて表示されることはありません。

Interactive Reanalysis Report では、表の見出し行は青色で示されます。

## U (下線)

U タグでは、下線付きで表示されるテキストを指定します。



**注:** 下線は、Interactive Reanalysis Report でのみ使用可能です。

図 285. Syntax



### text-to-be-underlined

このテキストが下線付きで表示されます。

## 説明

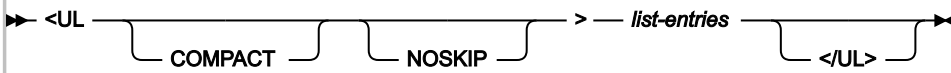
このタグによる改行は行われません。

Interactive Reanalysis Report では、下線付きテキストは青色で示されます。

## UL (順序なしリスト)

UL タグでは、順序なしリストを定義します。

図 286. Syntax



### COMPACT

この属性を指定すると、リスト項目間に空白行を作成せずにリストがフォーマットされます。

### NOSKIP

この属性を指定すると、リストの先頭に空白行を作成せずにリストがフォーマットされます。

### list-entries

LI タグを使用して指定されます。

## 説明

このタグでは改行が行われます。

## IDIXUFMT ロード・モジュール・フォーマット・ユーザー出口

IDIXUFMT ロード・モジュール・フォーマット・ユーザー出口 (以降のトピックでは IDIXUFMT 出口と呼びます) は、特殊なタイプのユーザー出口です。このユーザー出口は、REXX フォーマットかロード・モジュール・フォーマットかにかかわらず、この章で説明されている、他のユーザー出口に適用される通常の規則には従いません。

IDIXUFMT 出口は、Exits オプションでは指定できず、ロード・モジュール名で検索されます。IDIXUFMT という名前のロード・モジュールが Fault Analyzer 実行中に APF 許可ライブラリーで見つかり、分析レポートのフォーマット時にこのロード・モジュールが呼び出されます。IDIXUFMT 出口は、Exits オプションで指定された他のフォーマット・ユーザー出口の直前に呼び出されます。

IDIXUFMT 出口には以下の条件があります。

- LE 準拠でなければなりません。
- 「main」関数を含んでいてはなりません。
- NORENT オプションを使用してリンク・エディットされている必要があります。

Fault Analyzer は、IDIXUFMT (IDIXUFMT に付属する実行不能ロード・モジュール) 内に含まれている IDIXUFMT エントリー・ポイント経由で Fault Analyzer 出口を呼び出します。出口ユーザー・コードは、エントリー・ポイント IDIXUFMT 経由で呼び出されます。



## エントリーの指定

ユーザー・コード IDIXUFMT エントリー・ポイントは、以下を使用して呼び出されます。

- 以下の2つのフルワードを指す R1。
  - 最初のフルワードは、ENV データ域のアドレスです ([ENV - 共通出口環境情報 ページ 650](#)を参照)。
  - 2番目のフルワードは、UFM データ域のアドレスです ([UFM - ユーザー出口パラメーター・リストのフォーマット ページ 673](#)を参照)。
- 保管域を指す R13。
- 戻りアドレスを含む R14。
- IDIXUFMT エントリー・ポイント・アドレスを含む R15。

## 戻り指定

IDIXUFMT エントリー・ポイントからの戻り時には、以下のようになります。

- R0 および R1 は未定義です。
- R2 から R14 は未変更でなければなりません。
- R15 は未定義です。

## サンプル IDIXUFMT 出口

2つのサンプル IDIXUFMT 出口が、コンパイル (またはアセンブル) およびリンク用の JCL とともに提供されています。

- C IDIXUFMT サンプル出口が IDI.SIDISAM1(IDIXUFMC) に提供されています。
- アセンブラー IDIXUFMT サンプル出口が IDI.SIDISAM1(IDIXUFMA) に提供されています。

## IDIXUFMT 機能

Fault Analyzer 出口内から使用可能な、多数の IDIXUFMT 関数があります。

関数はすべて、8文字以内で構成された大文字の名前を持っています。これらの基本的な名前により、例えば高水準アセンブラーで書かれた出口でこれらの関数を呼び出すことが可能になります。

関数はすべて C リンテージを使用します。

以下のトピックでは、これらの関数について説明します。

## IDIXDLOC – システム・バッファリングを使用したダンプ・ストレージの検索

### フォーマット

```
#include "idixufmh.h"

int IDIXDLOC(int addr, int len);
```

## 概要

IDIXDLOC() 関数は、仮想アドレスと長さによって、分析されたアドレス・スペース内のストレージにアクセスする場合に使用します。



**注:** 大容量の仮想ストレージにアクセスする際にストレージ不足状態を防ぐには、代わりに IDIXXLOC 関数を使用します。詳しくは、[IDIXXLOC – 独自のバッファリングを使用したダンプ・ストレージの検索 ページ 561](#)を参照してください。

リアルタイム処理の場合、アクセスされるストレージは一般的に、分析中の実際のアドレス・スペースのストレージです。再分析の場合、ストレージはミニダンプから取得されます。

リアルタイム・モードか再分析モードかにかかわらず、保護例外が発生するおそれがあるため、IDIXDLOC() 関数の呼び出しによって取得されていないストレージ域を間接参照しないでください。

IDIXDLOC() は、Fault Analyzer REXX コマンド「Evaluate」と機能的に同等です。

## 戻り値

要求された長さでストレージが使用可能な場合、IDIXDLOC() はストレージ域のアドレスを返します。

IDIXDLOC() は、要求されたアドレスが使用可能でも、長さが要求された長さよりも短い場合は負の値を返します。使用可能な長さは、戻り値を 0 から減算することで判別できます。部分的なストレージ域のアドレスを取得するには、削減した長さを使用して再度 IDIXDLOC() を呼び出します。

長さにかかわらず、要求されたアドレスが使用できない場合は、値 0 が返されます。

## 例

```
#include "idixufmh.h"

int maddr, cvt;
maddr = IDIXDLOC(16,4); /* Get address of CVT pointer /
if (maddr > 0)
    cvt = (int)maddr; /* Get CVT pointer */
```

## IDIXEINF – イベント情報の取得

### フォーマット

```
#include "idixhfmt.h"

int IDIXEINF(UFM *p_ufm, int event_no);
```

## 概要

指定された UFM データ域には、指定されたイベント番号に該当する情報が取り込まれます。

IDIXEINF() は、Fault Analyzer REXX コマンド「IDIEventInfo」と機能的に同等です。

## 戻り値

情報が正常に取得された場合、IDIXEINF() はゼロを返します。

指定されたイベント番号に該当する情報がない場合、IDIXEINF() はゼロ以外を返します。エラーの説明が IDITRACE DDname に書き込まれます。

## 例

```
#include "idixhfmt.h"

UFM ufm;
int rc;
rc = IDIXEINF(&ufm, 1);
If (!rc) { // Successful completion
    ...
}
```

## IDIXGETN – データ域 10 進文字フィールド値の取得

### フォーマット

```
#include "dixhfmt.h"

#define IDIXGETN(pSrc) \
    (IDIXGETN)(pSrc, sizeof(pSrc))
int    (IDIXGETN)(char *pSrc, int src_len);
```

## 概要

IDIXGETN() は、10 進文字を含むデータ域文字フィールドを「int」値に変換する場合に使用します。

IDIXGETN() はマクロであり、同じ名前に関数を呼び出します。このマクロを使用する利点は、必要な引数が 1 つのみであるという点です。

## 戻り値

正常に実行された場合、IDIXGETN() は、文字列で表された変換済みの符号付き「int」値を返します。正常に実行されなかった場合は、未定義の値を返します。

## 例

```
#include "idixhfmt.h"

int num_events;
UFM ufm;
...
num_events = IDIXGETN(ufm.NUM_EVENTS);
```

## IDIXGETS - データ域文字フィールドを C ストリングとして取得

### フォーマット

```
#include "idixhfmt.h"

#define IDIXGETS(pSrc) \
    (IDIXGETS)(pSrc, sizeof(pSrc))
char * (IDIXGETS)(char *pSrc, int src_len);
```

### 概要

IDIXGETS() は、データ域文字フィールドからヌル終了ストリングを作成する場合に使用します。バッファ・データ・フォーマットではないデータ域フィールドの場合、ストリングは LE HEAP ストレージに作成され、IDIXUFMT 終了処理の完了時に自動的に解放されます。IDIXGETS() の呼び出し元が、返されたストレージ域を解放してはなりません。バッファ・データ・フォーマットであるデータ域フィールドの場合は、既存のバッファ・アドレスが返されます。

IDIXGETS() から返されたストリングを変更することによってデータ域フィールドを更新することはできません。

IDIXGETS() はマクロであり、同じ名前関数を呼び出します。このマクロを使用する利点は、必要な引数が 1 つのみであるという点です。

### 戻り値

要求されたデータ域フィールドのアドレスを、ヌル終了ストリングとして返します。

### 例

```
#include "idixhfmt.h"

UFM ufm;
char *psz;
...
psz = IDIXGETS(ufm.EVENT_TYPE);
If (strlen(psz) >= 6 && memcmp(psz,"Abend ") == 0) { // Abend event
...
}
```

## IDIXGETX - データ域 16 進文字フィールド値の取得

## フォーマット

```
#include "idixhfmt.h"

#define IDIXGETX(pSrc) \
    (IDIXGETX)(pSrc, sizeof(pSrc))
int    (IDIXGETX)(char *pSrc, int src_len);
```

## 概要

IDIXGETX() は、16 進文字 (0 から 9 または A から F) を含むデータ域文字フィールドを「int」値に変換する場合に使用します。

IDIXGETX() はマクロであり、同じ名前関数を呼び出します。このマクロを使用する利点は、必要な引数が 1 つのみであるという点です。

## 戻り値

正常に実行された場合、IDIXGETX() は、ストリングで表された変換済みの符号付き「int」値を返します。正常に実行されなかった場合は、未定義の値を返します。

## 例

```
#include "idixhfmt.h"

int pgm_len;
UFM ufm;
...
pgm_len = IDIXGETX(ufm.PROGRAM_LENGTH);
```

## IDIXLIST – ストレージ域をレポートに出力

### フォーマット

```
#include "idixhfmt.h"

int IDIXLIST(int addr, int len);
```

## 概要

IDIXLIST() は、分析後の障害環境からストレージ域を出力する場合に使用できます。addr 引数は、仮想ダンプ・アドレスでなければなりません。

出力されたストレージ域のフォーマットには、次の例に示すように、ストレージの見出しと、16 進および EBCDIC/ASCII の両方の表記が含まれます。

```
Address  Offset      Hex _____ EBCDIC / ASCII _____
```

```

00000000      040C0000 810692C8 00000000 00000000 *....a.kH.....*
00000010      +10 00FC7F08 00000000 070C1000 85532492 *.."......e..k*
00000020      +20 078D0000 00FC7F5A 078D1000 8775BB42 *....."!....g...*
00000030      +30 00000000 00000000 070C0000 85532496 *.....e..o*
00000040      +40 00000000 00000000 00000000 00FC7F08 *.....".*
00000050      +50 00000000 00000000 040C0000 810676D0 *.....a.}*
00000060      +60 040C0000 80FFB080 00080000 BF286880 *.....*
00000070      +70 00080000 BF287940 040C0000 81068A00 *.....`....a...*

```

IDIXLIST() は、Fault Analyzer REXX コマンド [List] と機能的に同等です。

## 戻り値

IDIXLIST() は常にゼロを返します。

## 例

```

#include "idixhfmt.h"

IDIXLIST(0,128); // Show address 0 for a length of 128 bytes

```

## IDIXNOTE – テキストの単純な行をレポートに書き込む

### フォーマット

```

#include "idixhfmt.h"

int IDIXNOTE(char *psz, ...);

```

## 概要

IDIXNOTE() は、フォーマットされていないテキストの行をレポートに書き込みます。psz 引数は、ヌル終了ストリングを指していなければなりません。(フォーマット済みテキストを書き込むには、代わりに IDIXWRIT() 関数を使用します)。

psz 引数が C sprintf() 関数での使用に適している書式制御ストリングである場合は、さらに必要な引数を後に続けることができます。

この関数は、Fault Analyzer REXX コマンド [Note] と同等です。

## 戻り値

IDIXNOTE() は常にゼロを返します。

## 例

```

#include "idixhfmt.h"

int i;

```

```
IDIXNOTE("Important data follows");
...
IDIXNOTE("A total of %d entries listed.",i);
```

## IDIXTRCE – テキストの単純な行を IDITRACE に書き込む

### フォーマット

```
#include "idixhfmt.h"

int IDIXTRCE(char *psz, ...);
```

### 概要

IDIXTRCE() は、以下の機能を実行する場合に使用できます。

- 渡されたパラメーターと最終戻りコードに関する情報を IDI DDname に書き込むことで、IDITRACE\* 関数呼び出しのトレースを制御します。
  - psz 引数が [(char)-1] として指定されている場合、IDI\* 関数呼び出しのトレースは使用可能です。
  - psz 引数が [NULL] として指定されている場合、IDI\* 関数呼び出しのトレースは使用不可です。
- フォーマットされていないテキストの行を IDITRACE DDname に書き込みます。psz 引数は、ヌル終了ストリングを指していなければなりません。このモードの関数は、REXX コマンド [SAY] を Fault Analyzer REXX ユーザー出口内から使用するのと同様です。

psz 引数が C sprintf() 関数での使用に適している書式制御ストリングである場合は、さらに必要な引数を後に続けることができます。

### 戻り値

IDIXTRCE() は常にゼロを返します。

### 例

```
#include "idixhfmt.h"

int i = 5;
IDITRACE((char *)-1); /* start IDI* function tracing */
IDIXTRCE("Couldn't format data area ABC.");
IDIXTRCE("A total of %d control blocks formatted.", i);
```

## IDIXWRIT – フォーマット済みテキストをレポートに書き込む

### フォーマット

```
#include "idixhfmt.h"
```

```
int IDIXWRIT(char *psz, ...);
```

## 概要

IDIXNOTE() 関数は、テキストの単純な行をレポートに書き込む場合にのみ使用できます。ただし、psz 引数で IDIXWRIT() 関数に渡されるストリングには、フォーマット・タグを含めることができます ([フォーマット・タグ ページ 542](#)を参照)。

psz 引数が C sprintf() 関数での使用に適している書式制御ストリングである場合は、さらに必要な引数を後に続けることができます。

この関数は、Fault Analyzer REXX コマンド [IDIWRITE] と同等です。

## 戻り値

IDIXWRIT() は常にゼロを返します。

## 例

```
#include "idixhfmt.h"

int i;
IDIXWRIT("<DL>");
for (i = 0; i < 10; ++i) {
    IDIXWRIT("<DT>Item %d</DT>",i);
    IDIXWRIT("<DD>Item description</DD>");
}
IDIXWRIT("</DL>");
```

## IDIXWTO – メッセージを MVS コンソールに書き込む

### フォーマット

```
#include "idixhfmt.h"

int IDIXWTO(char *psz, ...);
```

## 概要

IDIXWTO() は、メッセージを MVS™ コンソールに書き込む場合に使用します。psz 引数は、ヌル終了ストリングを指していなければなりません。

psz 引数が C sprintf() 関数での使用に適している書式制御ストリングである場合は、さらに必要な引数を後に続けることができます。

この関数は、Fault Analyzer REXX コマンド [IDIWTO] と同等です。



## 戻り値

IDIXWTO() は常にゼロを返します。

## 例

```
#include "idixhfmt.h"

IDIXWTO("Unable to complete IDIXUFMT exit processing!");
```

## IDIXXLOC – 独自のバッファリングを使用したダンプ・ストレージの検索

### フォーマット

```
#include "idixufmh.h"
int IDIXXLOC(void *bufptr, unsigned long long addr, int len);
```

### 概要

IDIXXLOC() 関数は、仮想アドレスと長さによって、分析されたアドレス・スペース内のストレージにアクセスする場合に使用します。

バッファのアドレスを *bufptr* で指定する必要があります。バッファの長さは *len* 以上でなければなりません。

リアルタイム処理の場合、アクセスされるストレージは一般的に、分析中の実際のアドレス・スペースのストレージです。再分析の場合、ストレージは、ミニダンプまたは拡張ダンプ・データ・セットから取得されます。

リアルタイム・モードか再分析モードかにかかわらず、保護例外が発生するおそれがあるため、IDIXXLOC() 関数の呼び出しによって取得されていないストレージ域を間接参照しないでください。

IDIXXLOC() は、Fault Analyzer REXX コマンド [Evaluate] と機能的に同等です。

## 戻り値

要求された長さでストレージが使用可能な場合、IDIXXLOC() は指定されたバッファ (*bufptr*) のアドレスを返します。IDIXXLOC() は、要求されたアドレスが使用可能でも、長さが要求された長さよりも短い場合は負の値を返します。使用可能な長さは、戻り値を 0 から減算することで判別できます。部分的なストレージ域のアドレスを取得するには、削減した長さを使用して再度 IDIXXLOC() を呼び出します。

長さにかかわらず、要求されたアドレスが使用できない場合は、値 0 が返されます。

## 例

```
#include "idixufmh.h"

int maddr, i;
char buffer[500];
```

```
maddr = IDIXXLOC(buffer, 0x001F0000, 500); /* Get 500 bytes of storage at addr 1F0000 */
if (maddr > 0)
    i = *(int *)maddr; /* Get first 4 bytes */
```

## ADFzCC イベント処理ユーザー出口とのインターフェース

ADFz Common Component 製品は、イベント処理ユーザー出口機能を備えています。この機能は、非同期インストール先作成バックエンドによるデータの処理を可能にします。この機能を使用するには、ADFz Common Components V1.8 をインストールします。詳しくは、『*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*』を参照してください。

この機能は、通知ユーザー出口を介して使用できます。詳しくは、「[通知ユーザー出口 ページ 499](#)」を参照してください。

## FA で提供されるサンプル

イベント処理機能を使用するために、FA では以下のサンプルを `IDI.SIDISAMI` に用意しています。

### サンプル通知出口 IDISXEPN

使用可能なすべての IDIWRITE データ域変数および `ENV` データ域変数用に `NFY` を呼び出す汎用通知ユーザー出口。このサンプル出口の詳細については、[通知ユーザー出口 ページ 499](#) を参照してください。

### サンプル・イベント処理ユーザー出口 IDISRTCC

C で書かれたサンプル出口。このサンプルは、IDISXEPN によって提供されたデータをデータ・セットに書き込みます。サンプルは次に、作成されたデータ・セットを入力として使用する REXX exec (`IDIINTR`) を実行するためのジョブを実行依頼する内部リーダーに、`IDISRTCR` という DD を動的に割り振ります。この他に、COBOL および PL/I のサンプルも、`IDISRTCB` および `IDISRTCP` にそれぞれ用意されています。

サンプル IDISRTCC 出口を使用するには、この出口を ADFzCC EventProcessingExit オプションで指定する必要があります。このオプションについて詳しくは、『*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*』を参照してください。

### サンプル REXX exec IDISRTCR

サンプル REXX exec IDISRTCR は、`IDISRTCC` で作成されたジョブによって実行されます。この exec が指定された入力データ・セットを読み取り、その内容 (IDISXEPN で指定されたデータ域変数 `ENV` および `NFY`) を使用して、障害項目についての関連情報で RTC 作業項目を作成します。次に、スクリプト内で指定された RTC サーバーに要求を送信します。当該 RTC プロジェクト下の作業項目が作成されます。この項目には発生した障害項目に関する情報が含まれます。

## 第 31 章. Fault Analyzer ヒストリー・ファイルにアクセスするための非 ISPF インターフェースのインストール

TSO/ISPF 以外のプラットフォームから Fault Analyzer ヒストリー・ファイルにアクセスできるようにするオプションのインターフェースのインストール要件について、以下のトピックで説明します。

### IBM Fault Analyzer プラグイン (Eclipse 用) のインストール

IBM® Fault Analyzer プラグイン (Eclipse 用) はオプション機能です。



**注:** この機能を使用するには IBM Application Delivery Foundation for z/OS® Common Components (ADFzCC) をインストールする必要があります。

IBM Fault Analyzer プラグインをインストールするには:

1. ホスト z/OS システムで、ADFzCC サーバーをカスタマイズします。
2. クライアント・ワークステーションで、IBM Explorer for z/OS または IBM Developer for z/OS と IBM Fault Analyzer プラグイン (Eclipse 用) をインストールします。

プラグインの詳細については、[IBM Fault Analyzer プラグイン \(Eclipse 用\) ページ 286](#) を参照してください。

### IBM Application Delivery Foundation for z/OS® Common Components サーバーのカスタマイズ

IBM Application Delivery Foundation for z/OS® Common Components サーバーのインストールについては、『ADFzCC カスタマイズ・ガイドおよびユーザー・ガイド』を参照してください。

Fault Analyzer サーバー用のサンプル ADFzCC 構成ファイルが、以下のように、データ・セット IDI.SIDISAM1 内にメンバー IDIGSVRJ として用意されています:

```
CONFIG=FA
SPAWN_PROGRAM=IDIGMAIN
SPAWN_REGIONSZ=500 ❶
SPAWN_JOBNAME=IDISVRF
SPAWN_PARMS_SECTION
ISPF_PROF_DSN=&USERID..ISPF.ISPPROF ❷
ISPF_APPL=IDI ❸
```

変更が必要なパラメーターは、次のものです。

❶

80,000 ページを超える非常に大きい障害項目ミニダンプをダウンロードする場合は、SPAWN\_REGIONSZ を大きくします。

❷

ISPF プロファイル・メンバーに別のデータ・セットが使用される場合は、&USERID..ISPF.ISPPROF を変更します。

&USERID は、接続されているユーザーのユーザー ID で置き換えます。例えば、ユーザー ID が FRED で、③ の ISPF アプリケーション ID が IDI である場合、ISPF プロファイルは FRED.ISPF.ISPPROF(IDIPROF) から取得されます。

ユーザー ID の一部のみを置換値として使用する場合、サブstring指定を使用できます (詳しくは [シンボル・サブstringの指定 ページ 583](#) を参照)。例えば、ユーザー ID の最初の 3 文字のみを高位修飾子として使用するには、&USERID(1:3)..ISPF.ISPPROF と指定します。

### ③

IDI に別の ISPF アプリケーション ID が使用される場合は、Fault Analyzer を変更します。デフォルト値は IDI です。

## IBM Fault Analyzer プラグイン (Eclipse 用) のインストール

IBM® Fault Analyzer プラグイン (Eclipse 用) は、IBM Explorer for z/OS® または IBM Developer for z/OS に対するプラグインです。

- ダウンロードは <https://ibm.github.io/mainframe-downloads/index.html> から利用できます。
- IBM Explorer for z/OS および IBM Developer for z/OS の資料は、<https://www.ibm.com/docs/en/adfz> から参照できます。

## CICS 環境での対話式再分析の使用可能化

この機能はオプション機能です。この機能を使用すると、Fault Analyzer ISPF インターフェースを使用して CICS® ログオンからヒストリー・ファイルおよび障害項目を表示でき、TSO ログオンは不要です。

このインターフェースの使用については、[以下の環境での対話式再分析の実行 CICS ページ 286](#) を参照してください。



**注:** この機能を使用するには、IBM Application Delivery Foundation for z/OS Common Components (必須の対話式パネル・ビューアーを含む) をインストールする必要があります。

Fault Analyzer をこのモードで実行するための特別な CICS® 領域をセットアップし、他の通常の CICS® トランザクションに影響を与えずに CPU およびストレージの使用を制御することをお勧めします。その理由は、このモードの Fault Analyzer 分析プログラムは、CICS® 領域で生成されるサブタスクで実行されるためであり、Fault Analyzer に同時にログオンして実行することが予想されるユーザー数に対応できる十分な大きさに JCL REGION= サイズを設定する必要があるためです。ストレージ必要量は分析対象の障害のサイズと複雑さに依存しますが、開始点として、同時ユーザー当たり 32 メガバイトが推奨されます。このストレージは CICS® DSA ストレージではなく、生成されるサブタスクの MVS™ GETMAIN 要求に対応できる量が使用可能でなければなりません。すなわち、REGION= サイズと CICS® EDSALIM 値の差分には、最低でも推定ストレージ所要量 (例えば、10 x 32 M = 320 M) が必要です。

Fault Analyzer を対話式 CICS® トランザクションとして実行するために必要な CICS® 領域のその他のセットアップ要件は、以下のとおりです。

1. 必要な CICS® リソース定義を作成します。
2. 必要な CICS® JCL の変更を行います。

以下のセクションで、これらの要件を説明します。IDI は、Fault Analyzer インストール時に使用されたデータ・セット名の高位修飾子とします。

## 必要な CICS リソース定義の作成

サンプル・ジョブが、IDI.SIDISAM1 データ・セット内にメンバー IDIWCIDI として提供されています。これを使用して、必要な CICS® リソース定義を作成できます。必要なトランザクション定義と関連のプログラム定義が、それぞれ 1 つずつあります。オプションのトランザクション・プロファイル定義も 1 つあり、そこでは、画面サイズを変更できるように SCRNSIZE(ALTERNATE) オプションを指定します。

## 必要な CICS JCL の変更

CICS® JCL には、以下の修正が必要です。

1. データ・セット IDI.SIDIAUTH を CICS® JCL の DFHRPL 連結に追加します。
2. 下記の IPVPROF および IPVTLIB DD 名に割り当てる新しいプロファイル・データ・セットを割り振ります。例えば、IDI.IDIPPROF。このデータ・セットは、LRECL=80 および RECFM=FB の PDS または PDSE として定義します。データ・セットは小さなものでかまいません。例えば、5トラックの 1 次スペース割り振りと 5トラックの 2 次スペース割り振りなどです。



**注:** ユーザーはそれぞれ、自身の ISPF 式の設定を IPVPROF データ・セットに書き込むため、このデータ・セットへの UPDATE アクセスを必要とします。

3. 以下の DD 名を CICS® JCL に追加します。

```
//IPVPLIB DD DISP=SHR,DSN=IPV.SIPVPENU
//          DD DISP=SHR,DSN=IPV.SIPVMENU
//          DD DISP=SHR,DSN=IDI.SIDIPLIB
//          DD DISP=SHR,DSN=IDI.SIDIMLIB
//          DD DISP=SHR,DSN=IDI.SIDISLIB
//IPVTLIB DD DISP=SHR,DSN=IDI.IDIPPROF
//          DD DISP=SHR,DSN=IPV.SIPVTENU
//          DD DISP=SHR,DSN=IDI.SIDITLIB
//IPVPROF DD DISP=SHR,DSN=IDI.IDIPPROF
```



**注:** データ・セット IPV.SIPV\* は、ADFz Common Components のインストールの一環として作成されます。

## 第 III 部. Fault Analyzer 参照情報

## 第 32 章. オプション

オプションを使用して、Fault Analyzer による出力を制御できます。例えば、次のようなオプションがあります。

- 障害分析レポート内容の変更。
- 異常終了時の Fault Analyzer のアクションの変更。

処理のほとんどの段階においてオプションを提供できます。オプションが現行の処理モードに関係ない場合 (例えば、バッチ再分析中に Exclude オプションを設定しようとした場合)、これは無視されます。この場合、Fault Analyzer は不要な警告メッセージを作成することはありません。

オプションは以下の優先順位で設定または変更できます。

1. Fault Analyzer が提供する製品のデフォルト。
2. SMP/E USERMOD。

詳しくは、「[USERMOD を使用した Fault Analyzer のカスタマイズ ページ 334](#)」を参照してください。

3. IBM Application Delivery Foundation for z/OS (ADFz) Common Components IPVOPTLM 構成オプション・モジュール。

Fault Analyzer では、IPVOPTLM 構成オプション・モジュールが存在する必要はありません。存在する場合、IDIOPTLM 構成オプション・モジュールで NOIPVOPT 値が 1 に設定されていない限り、Fault Analyzer により処理されます。[IBM Application Delivery Foundation for z/OS Common Components オプションを無視 \(NOIPVOPT\) ページ 343](#) を参照してください。

IPVOPTLM 構成オプション・モジュールの詳細については、『*IBM Application Delivery Foundation for z/OS Common Components: カスタマイズ・ガイドおよびユーザー・ガイド*』を参照してください。

4. ADFz Common Components IPVCNF00 parmlib メンバーに指定されたインストール・システム全般のデフォルト。

Fault Analyzer では、IPVCNF00 parmlib メンバーが存在する必要はありません。存在する場合、IDIOPTLM 構成オプション・モジュールで NOIPVOPT 値が 1 に設定されていない限り、Fault Analyzer により処理されます。詳しくは、[IBM Application Delivery Foundation for z/OS Common Components オプションを無視 \(NOIPVOPT\) ページ 343](#)を参照してください。

IPVCNF00 parmlib メンバーについては、『*IBM Application Delivery Foundation for z/OS Common Components: カスタマイズ・ガイドおよびユーザー・ガイド*』を参照してください。

5. 構成オプション・モジュール IDIOPTLM。

詳しくは、「[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#)」を参照してください。

6. ユーザー・オプション・モジュール IDICNFUM で検索されるオプション。

このモジュールはリアルタイム分析にのみ使用でき、検出された場合はステップ 7 [ページ 568](#) を省略します。

詳しくは、「[ユーザー・オプション・モジュール IDICNFUM ページ 570](#)」を参照してください。

7. parmlib のメンバー IDICNFxx に指定されたインストール・システム全般のデフォルト。

parmlib メンバーは、ユーザー・オプション・モジュールがステップ 6 ページ 567 で検出されなかった場合のみ、読み取られます。

詳しくは、「[parmlib メンバー IDICNFxx ページ 569](#)」を参照してください。

8. \_IDI\_OPTSFILE 環境変数を使用してユーザー・オプション・ファイルに指定されたオプション。

検出された場合は、ステップ 9 ページ 568 を置換してください。

詳しくは、「[\\_IDI\\_OPTSFILE 環境変数 ページ 571](#)」を参照してください。

9. IDIOPTS DDname を使用してユーザー・オプション・ファイルに指定されたオプション。

ユーザー・オプション・ファイルがステップ 8 ページ 568 で検出されなかった場合のみ、読み取られます。

詳しくは、「[User options file IDIOPTS ページ 571](#)」を参照してください。

10. バッチ再分析を実行する際に JCL EXEC ステートメントの PARM フィールドに指定されたオプション。

詳しくは、「[JCL EXEC ステートメント PARM フィールド ページ 572](#)」を参照してください。

11. \_IDI\_OPTS 環境変数を介して提供されるオプション。

詳しくは、「[\\_IDI\\_OPTS 環境変数 ページ 572](#)」を参照してください。

12. 分析制御ユーザー出口を介して設定されるオプション。

詳しくは、「[分析制御ユーザー出口 ページ 475](#)」を参照してください。

13. End Processing ユーザー出口での EPC データ域フィールドの設定値。有効な RetainDump および MaxMinidumpPages オプションを事実上オーバーライドする可能性があります。

詳しくは、「[終了処理ユーザー出口 ページ 494](#)」を参照してください。

オプションを指定しない場合は、製品のデフォルト (各オプションの構文図に示す) が使用されるか、あるいは値が指定されません。

オプションによっては、1 つの値しか保存できないものがあります。このようなオプションの複数のインスタンスが指定された場合、最後に現れたインスタンスのみが有効となります。例えば、次のように指定された場合、

```
PARM='Detail(LONG) Detail(SHORT)'
```

Detail(SHORT) がアクティブになります。

複数の値を適用できるオプションが、いくつかあります。例えば、DataSets、Exits、Include、および Exclude オプションがあります。これらが情報をどのように累積するかは、各オプションの項に説明されています。

オプションを指定するときは、以下の構文規則が適用されます。

- 列 1 から 71 のみが処理されます。
- オプションは行内の任意の位置に指定できます。1 列目から開始する必要はありません。



- ブランクまたはコンマを区切り文字として使用できます。
- オプションは、任意の行数にまたがって指定できますが、JCL EXEC ステートメントの PARM フィールドに指定する際は、z/OS® による 100 文字以内という制限があります。
- 複数行にまたがりオプション値が続く場合 (例えば、DataSets(IDIJAVA(...)) オプションを使用した長い HFS パス名の指定)、次のいずれかがサポートされます。
  - 列 71 を含むそこまでの値を指定し、次の行の列 1 から開始します。
  - 続く行の終わりにオプション値の継続文字「+」を指定します。プラス記号の前に 1 文字以上の空白が必要です。正符号を使用して継続する引用符で囲まれた値は、引用符で囲まれた値の各部分で指定する必要があります。

例:

```
DataSets(IDIJAVA('/this/might/be/a/really/long' +
                '/path/name'))
```

- オプション名およびキーワード・パラメーターでは大文字と小文字は区別されません。特定のオプションで明示的に言及された場合を除き、オプション値も大文字と小文字は区別されません。
- コメントは任意の位置に指定でき、ネストすることもできます。文字「/\*」がコメントの開始を、「\*/」が終了を示します。

## オプションの指定位置

オプションは 2 つのファイル (IDICNFxx はインストール・システム全般のデフォルト・オプションを、IDIOPTS はユーザー・オプションを保持しています) または JCL EXEC ステートメントの PARM フィールドに指定できます。オプションをファイルに指定した場合、これはすべての分析モードで使用できます。

リアルタイム分析のみの場合、ユーザー・オプション・モジュール IDICNFxx を介して、IDICNFUM のインストール・システム全般のデフォルト・オプションをジョブ・レベルで置換できます。

JCL EXEC ステートメントの PARM フィールドは、バッチ再分析においてのみ使用可能です。

障害再分析を実行する場合のオプションの変更方法については、[バッチ再分析オプション ページ 154](#)および [対話式再分析オプション ページ 162](#)を参照してください。

オプションの指定位置にかかわらず、分析制御ユーザー出口は、結果の設定値をオーバーライドできます。この出口タイプの詳細については、[分析制御ユーザー出口 ページ 475](#)を参照してください。

## parmlib メンバー IDICNFxx

メンバー IDICNFxx (xx は 00 または現在の MVS™ &SYSCONE シンボル値) は、SYS1.PARMLIB 内、または論理 parmlib 連結の一部であるその他のデータ・セット内に作成できます。IDIOPTLM 構成オプション・モジュールを使用して代替データ・セットを指定することもできます (オプション)。詳しくは、[代替 parmlib データ・セットを IDICNFxx\(CNFDSN\) に指定 ページ 340](#)を参照してください。

IDICNFxx メンバーには、Fault Analyzer を実行するたびに読み取られる、インストール・システム全般のデフォルト・オプションが含まれています。

parmlib 構成メンバーの例については、[図 199: IDICNFxx parmlib メンバーの例 ページ 373](#) を参照してください。

## ユーザー・オプション・モジュール IDICNFUM

リアルタイム分析でインストール・システム全般のデフォルト・オプションを置換するために、Fault Analyzer オプションを持つ 1 つ以上の区分データ・セットおよびメンバーが含まれているユーザー・オプション・モジュールを作成できます。

Fault Analyzer は、指定された順にデータ・セット・メンバーをオープンします。使用可能であると判別された最初のデータ・セットおよびメンバーは、論理 parmlib 連結、または IDICNFxx 構成オプション・モジュールの CNFDSN オプションで指定された代替 parmlib データ・セットの IDIOPTLM parmlib メンバーの代わりに使用されます。

ユーザー・オプション・モジュールは、IDICNFUM という名前にする必要があります。また、標準 MVS™ 検索パスから使用可能なロード・モジュールである必要があります。このモジュールが、JOB LIB ddname に割り振られているロード・ライブラリーに配置されると、デフォルト・オプションがジョブ・レベルで効果的に制御されるようになります。

このロード・モジュールは、次の標準 MVS™ JCL 構文フォーマットになっている区分データ・セット/メンバー名のみを含むことができます。

```
data-set-name (member-name)
```

各データ・セットの指定は、X'00' バイトで終了させてください。2 番目の X'00' バイトは、リストの終わりを示すために、最後に指定されたデータ・セットの後に置かなければなりません。データ・セットが指定されていない場合、少なくとも 1 つの X'00' バイトが必要です。

ユーザー・オプション・モジュールはより柔軟に使用できるようになりました。以下のシンボル名を使用してデータ・セットまたはメンバー名を指定できます。

### &SYSUID.

異常終了ジョブまたは CICS® トランザクションに関連したユーザー ID。

### &JOBNM.

異常終了ジョブのジョブ名。

### &PGMNM.

異常終了ジョブの EXEC ステートメントにあるプログラム名。

サンプル・ユーザー・オプション・モジュールを作成するためのジョブは、ソフトコピーのサンプル・データ・セットのメンバー IDISCNFU として提供されます。

Fault Analyzer により選択されたデータ・セットおよびメンバーは、メッセージ [IDI0001I ページ 695](#) で示されます。(変数を置換した後) 選択されなかったデータ・セット名およびメンバー名を確認する場合は、IDITRACE DDname をジョブ・ステップに組み込んでください。例:

```
//IDITRACE DD SYSOUT=*
```

(CICS® 環境でこのトレースを有効にする別の方法については、[IDITRACE CICS において ページ 407](#) を参照してください。)

選択されたユーザー・オプション・モジュールのデータ・セットおよびメンバーの名前は、Fault Analyzer によりヒストリー・ファイルに保管され、障害の再分析でデフォルト・オプション・ファイルとして自動的に再使用されます。バッチ再分析の場合、データ・セットおよびメンバーは、生成された JCL の IDIOPT DDname に組み込まれます。

## \_IDI\_OPTSFILE 環境変数

アプリケーション・プログラムは、異常終了または \_IDI\_OPTSFILE を呼び出す前に、MVS™ データ・セット (およびオプションでメンバー) 名、または HFS パスとファイル名 (Fault Analyzer へのオプションを含む) を使用して、環境変数の IDISNAP を初期化できます。

### • MVS™ データ・セット名の指定

MVS™ データ・セット名を指定する場合は、名前は 2 つのスラッシュ (//) で始まる必要があります。指定するデータ・セット名には、大/小文字の区別はありません。

指定するユーザー・オプション・ファイルは、80 バイトの固定レコード長のフォーマットで、すべてのオプションはカラム 1 から 71 までに指定する必要があります。

### • HFS パスおよびファイル名の指定

HFS パスおよびファイル名を指定する場合は、名前は 2 つのスラッシュで始めないでください。

指定するユーザー・オプション・ファイルは特別な属性に従うことはなく、オプションは任意のカラムに指定できます。

例:

```

/* Sample BPXBATCH job
//RUN EXEC PGM=BPXBATCH, ...
:
//STDENV DD * (select one of the following)
_IDI_OPTSFILE=//my.SEQ.OPTIONS
_IDI_OPTSFILE=//my.PDS.OPTIONS(faopts)
_IDI_OPTSFILE=/u/fred/options_file
:
/*

```



**注:** ユーザー・オプション・ファイルが \_IDI\_OPTSFILE 環境変数を使用して検出された場合は、IDIOPTS DDname を使用してオプションの読み取りを試行することはありません ([User options file IDIOPTS ページ 571](#)を参照)。

## User options file IDIOPTS

Fault Analyzer は、IDIOPTS DDname によるユーザー・オプション・ファイルの指定をサポートします (CICS® ユーザーの場合は、[IDIOPTS DDname を使用した CICS オプションの指定 ページ 417](#)を参照してください)。

インストリームユーザー・オプション・ファイルを使用して、プログラム MYAPPL が異常終了した場合にダンプ抑止をオーバーライドする、ジョブの例を以下に示します。

図 287. ユーザー・オプション・ファイルを指定するサンプル・ジョブ

```
//MYJOB1 JOB ...
//STEP1 EXEC PGM=MYAPPL
//SYSMDUMP DD DISP=SHR,DSN=MY.DUMP.DATA.SET
//IDIOPTS DD *
    RetainDump(ALL) /* do not suppress the dump if MYAPPL
                    abends */
/*
```

ユーザー・オプション・ファイルは 80 バイトの固定レコード長フォーマットでなければなりません。すべてのオプションを 1 桁目から 71 桁目までに指定する必要があります。



**注:** ユーザー・オプション・ファイルが `_IDI_OPTSFILE` 環境変数により検出された場合 ([\\_IDI\\_OPTSFILE 環境変数 ページ 571](#) を参照)、IDIOPTS DDname を使用したオプションの読み取りを試行することはありません。

## JCL EXEC ステートメント PARM フィールド

Fault Analyzer をバッチ再分析モードで実行する際、JCL EXEC ステートメントの PARM フィールドにオプションを指定できます。例:

```
//MYJOB2 JOB
//STEP1 EXEC PGM=IDIDA,
// PARM='/Detail(LONG),PreferredFormattingWidth(132)'
:
```

## \_IDI\_OPTS 環境変数

アプリケーション・プログラムは、異常終了または `_IDI_OPTS` を呼び出す前に、Fault Analyzer のオプションを使用して環境変数 `IDISNAP` を初期化できます。

例:

```
/* Sample BPXBATCH job
//RUN EXEC PGM=BPXBATCH, ...
:
//STDENV DD *
_IDI_OPTS=DataSets(IDIHIST(MY.HIST)),Detail(L)
:
/*
```

## オプションに対する変更が有効な場合

一般に、オプションに対する変更は、その指定方法に関係なく、即時に有効になります。唯一の例外は IDIS サブシステムです。このサブシステムは、以下のいずれかのオプションに変更を加えた場合には、停止して再始動する必要があります。

- DataSets (IDIVSxxx(...),IDIEEXEC(...),IDIDOC(...),IDIDOxxx(...),IDIVIEWS(...))
- DumpRegistrationExits
- Exclude/Include
- Language
- Locale

- NoDup(ImageFast(...))
- Quiet

ここで xxx は、有効な Language オプションの言語 ID です。(IDIDOxxx は ENU をサポートしません。)

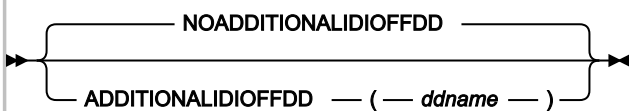
CNFDSN 設定が上記のいずれかのオプションを変更する IDICNFxx メンバーを指定するために使用されていない限り、IDIOPTLM 設定オプション・モジュールを実装したり、変更したりするために IDIS サブシステムを停止して再起動する必要はありません。

## オプションの説明

各オプションの詳細を以下に説明します。

### AdditionalIDIOffDD

図 288. Syntax



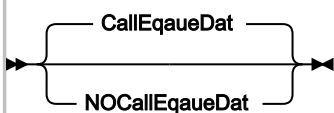
AdditionalIDIOffDD オプションを使用すると、通常の Fault Analyzer IDIOFF DDname と同等の追加の DDname を提供できます (詳細については、[JCL スイッチ \(Fault Analyzer\) を使用した IDIOFF の停止 ページ 463](#)を参照)。このオプションが指定された場合、ddname が割り振られているジョブはいずれも、リアルタイム異常終了処理を行うために Fault Analyzer を起動しません。

このオプションは、Fault Analyzer IDIS サブシステムによって読み取られ、その設定は、クロスメモリー・アクセスによる領域の異常終了に対して対応可能にされます。AdditionalIDIOffDD オプションに対する変更は、Fault Analyzer IDIS サブシステムを停止して再始動した後で、初めて有効になります (詳細については、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照)。

このオプションは、有効なオプションを示す、障害分析レポートのセクションには含まれません。

### CallEqueueDat

図 289. Syntax



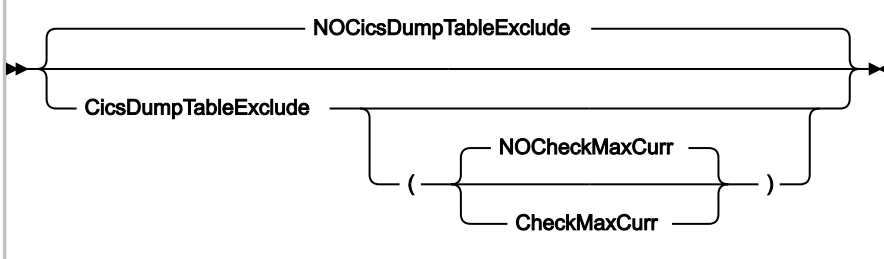
CallEqueueDat (デフォルト) では Fault Analyzer に IBM® z/OS® Debugger リスト出口を呼び出し、分析中に使用されるソース・リストを検索するよう指示します。見つからなかった場合、処理が続行されます。

NOCallEqueueDat は出口を呼び出さないようにします。

このオプションは、有効なオプションを示す、障害分析レポートのセクションには含まれません。

## CICSDumpTableExclude

図 290. Syntax



CICSDumpTableExclude オプションは、同じ異常終了コードに対する CICS® トランザクション・ダンプ・テーブルのアクションに NOTRANDUMP が指定されている場合に、Fault Analyzer リアルタイム処理から CICS® トランザクション異常終了を除外します。

除外されていると、メッセージ [IDIO101I ページ 717](#) が出され、分析レポートは生成されず、履歴ファイルに障害項目が書き込まれません。

このオプションに基づいた Fault Analyzer 処理の除外は、Exclude オプションまたは NoDup オプション ([リアルタイム除外処理 ページ 54](#)を参照) など、分析を阻止する Fault Analyzer の他のどの方法よりも優先されます。

CheckMaxCurr サブオプションを指定した場合、Fault Analyzer は、特定ダンプ・コードに対する CICS® ダンプ要求の現行数を、そのダンプ・コードの最大設定と比較します。現行値が最大値を超えている場合、Fault Analyzer 分析はスキップされ、メッセージ [IDIO180I ページ 736](#) が表示されます。



**注:** CICS では最初の異常終了が発生した後にのみトランザクション・ダンプ・テーブル (TDT) 項目が作成され、Fault Analyzer が呼び出されます (XPCABND または LE 異常終了出口を使用している場合)。このため、CICS パラメーター TRDUMAX が 0 に設定されている場合、CheckMaxCurr がチェックする TDT エントリーがないため、Fault Analyzer は、最初の異常終了インスタンスの処理を除外できません。Fault Analyzer は、CICS が TDT を作成した後の異常終了の分析を除外できます。これは、CheckMaxCurr が現在のインスタンス数が 0 を超えていることを検出できるためです。

デフォルト設定は NOCheckMaxCurr です。この場合、ダンプ・コードの最大設定とは無関係に分析が続行されます。

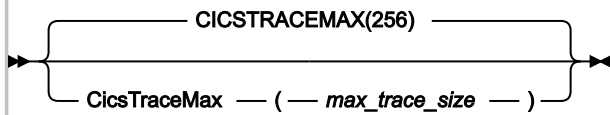
NOCICSDumpTableExclude が有効である場合 (デフォルト)、Fault Analyzer は、その除外検査で CICS® トランザクション・ダンプ・テーブルを使用しません。

このオプションは、有効なオプションを示す、障害分析レポートのセクションには含まれません。

このオプションに対する変更を有効にするには、CFA トランザクションを使用してすべての CICS 呼び出し出口をアンインストールおよび再インストールします。詳しくは、[CICS トランザクション異常終了分析の制御 ページ 405](#)を参照してください。

## CICSTraceMax

図 291. Syntax

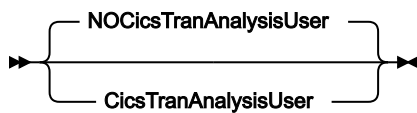


CICSTraceMax オプションは、CICS® システム・ダンプ分析にのみ適用できます。このオプションを使用して、障害項目ミニダンプに含めることができる最大 CICS® トレース・テーブル・サイズをキロバイト数で指定します。

*max\_trace\_size* の値の有効な範囲は 0 から 2097151 キロバイトです。指定されていない場合、CICSTraceMax オプション値はデフォルトにより 256 キロバイトになります。

実際の CICS® トレース・サイズが有効な CICSTraceMax オプション値を超えている場合、障害項目の再分析時に、トレース・テーブルは関連の CICS® システム・ダンプから読み取られます。

## CICSTranAnalysisUser

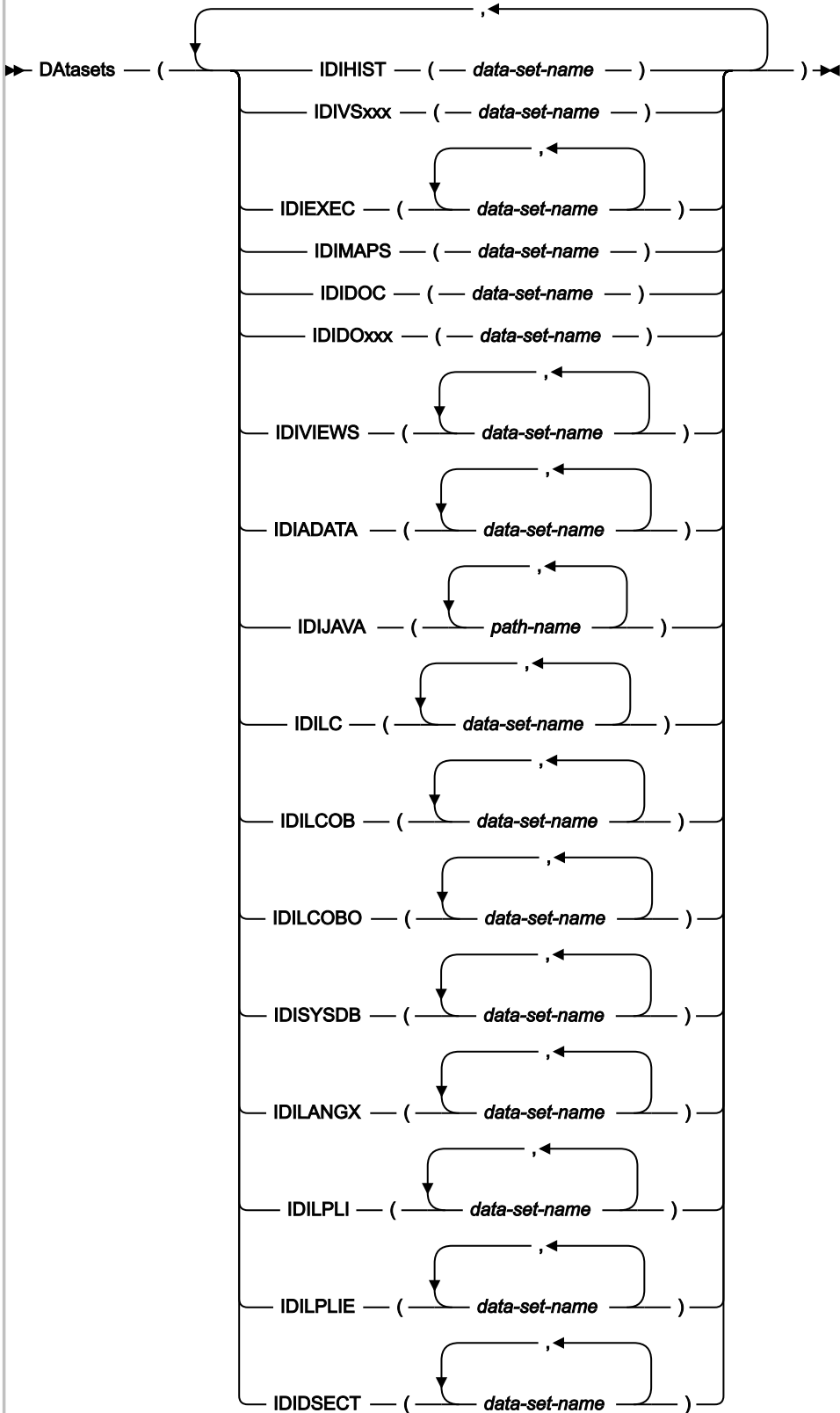


CICSTranAnalysisUser オプションを指定して、サインオン CICS ユーザーの資格情報で障害分析を実行できるようにします。それ以外の場合、障害分析は CICS 領域のユーザー ID の下で実行されます。

このオプションは、有効なオプションを示す、障害分析レポートのセクションには含まれません。

## DataSets

292. Syntax





DataSets オプションでは、Fault Analyzer により動的に割り振られる DDname とその関連データ・セット名をサブオプションとして指定します。

#### IDIHIST

障害項目が書き込まれるか、または書き込まれた、PDS または PDSE ヒストリー・ファイルの名前。デフォルト値は IDI.HIST です。

#### IDIEXEC

REXX ユーザー出口を含む 1 つ以上の PDS または PDSE データ・セットの名前。

#### IDIMAPS

Fault Analyzer for z/OS® に提供されているデータ域マッピングが含まれる PDS または PDSE データ・セットの名前。デフォルト値は IDI.SIDIMAPS です。

#### IDIDOC

分散ブック索引とオーバーライド PDS または PDSE データ・セットの名前。デフォルト値は IDI.SIDIDOC1 です。

#### IDIDOxxx

多文化サポート・フィーチャー用の追加の分散ブック索引とオーバーライド PDS または PDSE データ・セットの名前。xxx は、ENU 以外の Language オプションの有効な言語 ID です (例えば、IDIDOJPN)。指定されたデータ・セットが使用されるのは、同等な Language オプションが有効な場合のみです。デフォルト値は指定されません。

#### IDIVSxxx

VSAM KSDS メッセージおよび異常終了コード説明リポジトリの名前。ここで、xxx は Language オプションに有効な言語 ID です (例えば、IDIVSENU)。ENU 以外の xxx 値の場合、指定されたデータ・セットが使用されるのは、同等な Language オプションが有効な場合のみです。デフォルト値は IDI.IDIVSENU です。

#### IDIVIEWS

単一の ISPF 表示に示される障害ヒストリー・ファイルを定義するメンバーが含まれる 1 つ以上の PDS または PDSE データ・セットの名前。

#### IDIADATA

アセンブラー SYSADATA ファイルを保持する 1 つ以上の順次、PDS または PDSE データ・セットの名前。

この DD 名に必要なデータ・セット属性について詳しくは、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

#### IDIJAVA

1 つ以上の HFS パス名は、Java ソース・コードを検索するときに使用されます。

デフォルトでは、このオプションを使用して指定されたパス名は、Java クラス・パスの前に連結されます。ただし、特別な値 -DROPCP- が IDIJAVA 指定内のどこかで発生した場合、IDIJAVA DDname

に指定したパス名だけが検索されます。例えば、次のように指定すると、パス名 /a/b/c および /d/e/f だけが検索され、Java クラス・パスは無視されます。

```
DataSets(IDIJAVA(/a/b/c,-dropcp-,/d/e/f))
```

次の IDIJAVA ステートメントが指定されると、Java ソース・コードは検索されません。

```
DataSets(IDIJAVA(-dropcp-))
```

パス名はスラッシュ (/) で始まる必要があり、オプションで単一引用符で囲むことができます。

パス名に単一引用符、コンマまたはブランクが含まれる場合も、単一引用符で囲む必要があります。パス名内の単一引用符はすべて、2つ重ねる必要があります。例えば、パス名 /a/b', c は次のように指定する必要があります。

```
'/a/b', c'
```

各パス名の最大長は、囲んだ単一引用符を除き、1023 文字です。

パス名には大/小文字の区別があります。-DROP- キーワードの大/小文字は区別されません。

#### IDILC

C コンパイラー・リストを保持する 1 つ以上の順次、PDS または PDSE データ・セットの名前。

この DD 名に必要なデータ・セット属性については、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

#### IDILCOB

COBOL コンパイラー・リスト (OS/VS COBOL 以外) を保持する 1 つ以上の順次、PDS または PDSE データ・セットの名前。

この DD 名に必要なデータ・セット属性については、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

#### IDILCOBO

OS/VS COBOL コンパイラー・リストを保持する 1 つ以上の順次、PDS または PDSE データ・セットの名前。

この DD 名に必要なデータ・セット属性については、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

#### IDISYSDB

COBOL または Enterprise PL/I SYSDEBUG サイド・ファイル、または XL C/C++ MDBG サイド・ファイルを含む 1 つ以上の順次、PDS または PDSE データ・セットの名前。(これらのサイド・ファイルは、TEST(,SEPARATE) オプションを指定した COBOL プログラムのコンパイル時に作成されます。MDBG サイド・ファイルは CDADBGLD ユーティリティーを使用して作成されます。)

この DD 名に必要なデータ・セット属性については、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

**IDILANGX**

LANGX サイド・ファイルを保持する 1 つ以上の順次、PDS または PDSE データ・セットの名前。

この DD 名に必要なデータ・セット属性については、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

**IDILPLI**

PL/I コンパイラー・リスト (Enterprise PL/I 以外) を保持する 1 つ以上の順次、PDS または PDSE データ・セットの名前。

この DD 名に必要なデータ・セット属性については、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

**IDILPLIE**

Enterprise PL/I コンパイラー・リストを保持する 1 つ以上の順次、PDS または PDSE データ・セットの名前。

この DD 名に必要なデータ・セット属性については、[コンパイラー・リストおよびサイド・ファイルの属性 ページ 391](#)を参照してください。

**IDIDSECT**

対話式再分析 DSECT コマンドで使用される、アセンブラー・マクロまたは DSECT サンプル集を含む 1 つ以上の PDS または PDSE データ・セット。詳しくは、[DSECT 情報を使用したストレージ域のマッピング ページ 227](#)を参照してください。

IDIHIST、IDIEEXEC、IDIMAPS、IDIDOC、IDIVSxxx、および IDIDOxxx はデータ・セット名の値を 1 つだけ持ち、2 つ目の指定は累積されず置き換えられます。これら以外の DataSets サブオプションの複数指定は、累積します。すべてのデータ・セットは (既に指定されていた場合)、それぞれの DDname の最終の論理連結に組み込まれます。

どの DDname に対して DUMMY データ・セットを指定する場合も、データ・セット名 NULLFILE を使用できます。

リアルタイム分析時に使用されるコンパイラー・リストまたはサイド・ファイルのデータ・セットの名前は、障害項目とともにヒストリー・ファイルに保管され、再分析が実行される場合に自動的に使用されます。したがって、リアルタイム分析時に使用できなかったコンパイラー・リストまたはサイド・ファイルが使用可能になる場合を除いて、一般に、再分析に DataSets オプションを使用してデータ・セットを指定する必要はありません。

**データ・セットの論理的な置換または連結の順序**

Fault Analyzer データ・セットの論理的な置換または連結の順序は、次のとおりです。

1. 分析制御ユーザー出口により提供されるデータ・セット (詳細については、[分析制御ユーザー出口 ページ 475](#) ページを参照してください)。
2. DDname について明示的にコーディングされた JCL。
3. JCL EXEC ステートメントの PARM フィールドに指定された、すべての DataSets オプションからのデータ・セット (再分析の場合のみ)。
4. ユーザー・オプション・ファイルに指定された、すべての DataSets オプションからのデータ・セット。

5. parmlib 構成メンバーに指定された、すべての DataSets オプションからのデータ・セット。

これらのオプションには、システム全体の代替 parmlib データ・セット (詳細については、[parmlib メンバー IDICNFxx ページ 372](#)を参照)、およびユーザー・オプション・モジュールを介して指定されるデータ・セット (詳細については、[ユーザー・オプション・モジュール IDICNFUM ページ 570](#)を参照) が含まれます。

## IDICNFxx parmlib メンバーのデータ・セット指定の廃棄

特定のデータ・セット名 -DROPCNF- を使用して IDICNFxx parmlib メンバーに指定された DDname への、すべてのデータ・セット指定を廃棄できます。

例えば、IDICNFxx parmlib メンバーが以下を含むとします。

```
DataSets (IDILCOB (FRED.L1, FRED.L2))
```

また、IDIOPTS ユーザー・オプション・ファイルが以下を含むとします。

```
DataSets (IDILCOB (FRED.L3, -DROPCNF-, FRED.L4))
```

この場合、IDILCOB DDname のデータ・セットの最終論理連結は、次のようになります。

```
IDILCOB DD DISP=SHR, DSN=FRED.L3
          DD DISP=SHR, DSN=FRED.L4
```

-DROPCNF- データ・セット名は DataSets オプションを指定可能なすべての DDname に指定できます。ただし、分析制御ユーザー出口からは指定されません。

## DataSets オプションのデータ・セット名の置換シンボル

DataSets オプションを使用して指定されたすべてのデータ・セット名には、標準の MVS™ シンボルを含めることができます。それらのシンボルは、Fault Analyzer によって DDname 固有のシンボル置換が行われる前に解決されます。

[表 14: DDname ごとの使用可能なデータ・セット名置換シンボル ページ 580](#) で、X は各 DDname 用にデータ・セット名の一部として使用できる許可されたシンボルであることを示します。

**表 14. DDname ごとの使用可能なデータ・セット名置換シンボル**

DDname	&PGM.	&SYSUID.	&TSOPFX.	&USERID.	&SYSNAME.
IDIHIST					X
IDIVSxxx					X
IDIEEXEC				X (注 1)	X
IDIMAPS					X
IDIDOC					X
IDIDOxxx					X
IDIVIEWS (注 2)		X	X		X
IDIADATA	X	X (注 3)	X (注 3)	X	X

表 14. DDname ごとの使用可能なデータ・セット名置換シンボル (続く)

DDname	&PGM.	&SYSUID.	&TSOPFX.	&USERID.	&SYSNAME.
IDIJAVA	X	X (注 3)	X (注 3)	X	X
IDILC	X	X (注 3)	X (注 3)	X	X
IDILCOB	X	X (注 3)	X (注 3)	X	X
IDILCOBO	X	X (注 3)	X (注 3)	X	X
IDISYSDB	X	X (注 3)	X (注 3)	X	X
IDILANGX	X	X (注 3)	X (注 3)	X	X
IDILPLI	X	X (注 3)	X (注 3)	X	X
IDILPLIE	X	X (注 3)	X (注 3)	X	X
IDIDSECT				X	X

**注記:**

1

MVS™ ダンプ分析の実行時に &USERID. 変数には値を使用できません。(File->Analyze MVS Dump Data Set ページ 100を参照。)

2

これらのシンボルは、DataSets オプションの IDIVIEWS サブオプションで指定するデータ・セット名の中で使用できるほか、IDIVIEWS データ・セットに含まれる個々のビュー・メンバーの中で指定するデータ・セット名の中でも使用できます。詳しくは、[ビューのセットアップ ページ 350](#)を参照してください。

3

これらのシンボルの置換は、対話式再分析時にのみ実行されます。

指定した MVS™ システムで使用可能なシンボルは、MVS™ オペレーター・コマンド D SYMBOLS で表示できます。

シンボル置換の実行時にシンボルの値が使用できない場合、そのシンボルを含んでいるデータ・セット名はすべて無視されます。この無視の例は、前述の注にある &USERID. 変数です。

シンボル名がデータ・セット名の最後でない限り、指定されたすべてのシンボル名には終了ピリオドが含まれていなければなりません。シンボル名は、先行アンパーサンド (&) と終了ピリオドを含め、シンボル値に置換されます。そのため、シンボル名をデータ・セット修飾子区切り文字 (ピリオド) の直前に指定する場合は、2つの連続するピリオドを指定する必要があります (例えば「&SYSUID..LISTINGS」)。

そのデータ・セットがないと分析を実行できない場合を除いて、置換の後でデータ・セットが見つからない場合でも、シンボルを含むデータ・セット名が原因で Fault Analyzer 処理中にエラーが発生することはありません。

置換シンボル:

#### &PGM.

このシンボルのインスタンスはすべて、Fault Analyzer がコンパイラー・リストまたはサイド・ファイルを検索する際に、現行のプログラム名で置換されます。この機能を使用するには、順次データ・セットのコンパイラー・リストまたはサイド・ファイルを、プログラム名を含む命名規則で保管できます。

例えば、プログラム P1 および P2 のコンパイラー・リストがデータ・セット FRED.LISTING.P1 および FRED.LISTING.P2 にそれぞれ保管される場合、これらは両方とも DataSets オプションの適切な DDname に「FRED.LISTING.PGM.」を指定することで Fault Analyzer によって配置できます。

#### &SYSUID.

このシンボルのインスタンスはすべて、現在 Fault Analyzer ISPF インターフェイスを使用している TSO ユーザー ID で置換されます。IDICNFxx parmlib メンバー内で指定されたデータ・セット名にこのシンボルを使用すると、インストールで適切なデータ・セット命名規約が遵守されている場合、プライベート・データ・セットに対し自動的にユーザー固有のアクセスが可能になります。

例えば、次のオプションを

```
DataSetS(IDIVIEWS(&SYSUID..VIEWS,PROD.VIEWS))
```

IDICNFxx parmlib メンバーに指定します。これにより、*user-id.VIEWS* というデータ・セット (*user-id* はユーザーの TSO ユーザー ID) が存在する Fault Analyzer ISPF インターフェイスのすべてのユーザーは、専用の Fault Analyzer View 定義をこのデータ・セットに配置できます。これらの定義は、PROD.VIEWS データ・セットにあるインストール済み環境全体の定義を補完するか、またはそれらの定義をオーバーライドします。

#### &TSOPFX.

このシンボルのインスタンスはすべて、現在 Fault Analyzer ISPF インターフェイスを使用している TSO プロファイル接頭部で置換されます。IDICNFxx parmlib メンバー内で指定されたデータ・セット名にこのシンボルを使用すると、インストールで適切なデータ・セット命名規約が遵守されている場合、プライベート・データ・セットに対し自動的にユーザー固有のアクセスが可能になります。

例えば、次のオプションを

```
DataSetS(IDIVIEWS(&TSOPFX..VIEWS,PROD.VIEWS))
```

IDICNFxx parmlib メンバーに指定します。これにより、*tso-prefix.VIEWS* というデータ・セット (*tso-prefix* はユーザーの TSO プロファイル接頭部) が存在する Fault Analyzer ISPF インターフェイスのすべてのユーザーは、専用の Fault Analyzer View 定義をこのデータ・セットに配置できます。これらの定義は、PROD.VIEWS データ・セットにあるインストール済み環境全体の定義を補完するか、またはそれらの定義をオーバーライドします。

#### &USERID.

このシンボルのインスタンスはすべて、異常終了が発生したユーザー ID で置換されます。

**&SYSNAME.**

このシンボルのインスタンスはすべて、置換時に CVTSNAME のシステム名で置き換えられます。

**シンボル・サブストリングの指定**

すべてのシンボル (MVS™ システム・シンボルと Fault Analyzer 固有のシンボル) のサブストリングの指定が許可されます。

図 293. Syntax

The diagram illustrates the syntax for specifying a substring of a symbol. It shows the format: `& symbol_name ( start : number ) .`. A horizontal line with arrows at both ends spans from the opening parenthesis to the closing parenthesis. A vertical line descends from the `start` parameter, and another vertical line descends from the `number` parameter. A horizontal line connects these two vertical lines, with a colon `:` positioned between them, indicating the range of characters to be extracted.

説明:

**symbol\_name**

シンボルの名前 (USERID など)。

**start**

サブストリングが開始する文字の位置。オリジナルのシンボルの最初の文字は位置 1、2 番目は位置 2、というようになります。 *start*

が正の数るとき、システムは開始位置からストリングの最後までカウントします。 *start* が負の数るとき (つまり、前に負符号が付いているとき)、システムはストリングの終わりの位置から先頭の方に向かってカウントします。

**number**

サブストリングに入れるストリングの開始位置から終了位置までの文字数。 *number*

を指定しない場合、サブストリングの長さはデフォルトで 1 になります。 *number* に負の数を指定しないでください。

例えば、シンボル &PGM. の代替値に MYPROG1 が含まれており、修飾子を最初の 6 文字に制限する場合は、次のように指定できます:

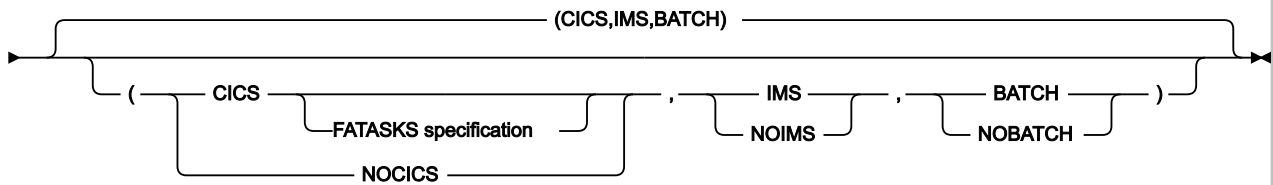
```
&PGM(1:6).
```

シンボル・サブストリングの指定に関する詳細情報については、「MVS™ 初期設定およびチューニング解説書」の『システム・シンボルのサブストリングの使用法』のセクションを参照してください。二重アンパーサンド表記の使用はサポートされていないことに注意してください。

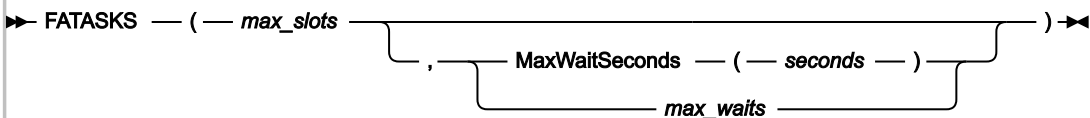
## DeferredReport

図 294. Syntax

DEFERREDREPORT →



FATASKS specification



このオプションは、リアルタイム分析にのみ適用され、リアルタイム・パフォーマンスがクリティカルなときに使用できます。使用時には、リアルタイム分析レポートは生成されませんが、後から再分析できる障害項目が書き込まれます。

Fault Analyzer の以前のバージョンとの互換性のために、次の指定

```
DeferredReport(CICSFATasks(max_slots<,max_waits>))
```

は、今後通知されるまで使用できます。この指定は、以下に相当します。

```
DeferredReport(CICS(FATasks(max_slots<,max_waits>)))
```

DeferredReport オプションを有効にする実行環境の指定は、CICS® | NoCICS、IMS™ | NoIMS、または Batch | NoBatch サブオプションを使用して行います。

- DeferredReport オプションが指定されていない場合、製品のデフォルトは CICS® のみです:

```
DEFERREDREPORT(CICS(FATASKS(1,20)),NOIMS,NOBATCH)
```

- サブオプションなしで DeferredReport オプションが指定されている場合、デフォルトはすべての実行環境です。

DeferredReport オプションについて言えば、CICS® とのインターフェースにもなる IMS™ トランザクションは、CICS® サブオプションを使用して制御されます。従って、

```
DeferredReport(NoCICS,IMS)
```

の指定は、そのようなアプリケーションに対する DeferredReport オプションを使用不可にしますが、

```
DeferredReport(CICS,NoIMS)
```

はこれを使用可能にします。

CICS® で使用される場合は、オプションの FATASKS サブオプションを指定できます:

### max\_slots

使用可能にする実行スロットの最大数を指定します。実行スロットごとに、Fault Analyzer リアルタイム分析の 1 つのインスタンスの実行が許可され、効果的な並列実行またはマルチタスキングを可能にします。



有効な値の範囲は、1 から 6 です。

デフォルトは 1 です。

NODeferredReport が指定された場合、実行スロットの最大数には 1 が設定されます。

### MaxWaitSeconds(seconds)

障害がキューに入れられて分析を待機できる最大の秒数を指定します。有効な現在の制限より長く障害が待機していた場合は、メッセージ [IDIO132W ページ 725](#) が発行され、その障害の分析はスキップされます。このため、通常の (すなわち、Fault Analyzer を使用しない) CICS® トランザクション・ダンプ分析の実行が必要になります。

このサブオプションが指定されている場合、待機キュー長は常に 20 です。max\_waits サブオプションに説明された動作は、20 個の障害が既にキューに待機している場合に発生した障害にそのまま適用できます。

有効な値の範囲は、0 から 3600 です。0 が指定されると、時間は無制限になります。

デフォルトは 0 です。

max\_waits の代わりに MaxWaitSeconds(seconds) を使用することをお勧めします。

### max\_waits

分析を開始するまで待機する障害の最大数を指定します。別の障害が発生したときに既に最大数に達していた場合は、メッセージ [IDIO118W ページ 721](#) が発行され、その障害はスキップされるため、通常の (つまり、Fault Analyzer を使用しない) CICS® トランザクション・ダンプ分析の実行が必要になります。

有効な値の範囲は、1 から 20 です。

デフォルトは 20 です。



**注:** CPU に制約がある環境で分析処理に時間がかかる場合は、max\_waits の値を小さくする必要があるかもしれません。この削減により、[IDIO118W ページ 721](#) メッセージが出され、異常終了の分析が拒否されるおそれがありますが、異常終了アクティビティ数が多いときに、オーバーコミット環境で異常終了処理のためにタスクを待機させるよりは好ましい場合があります。



**注:**

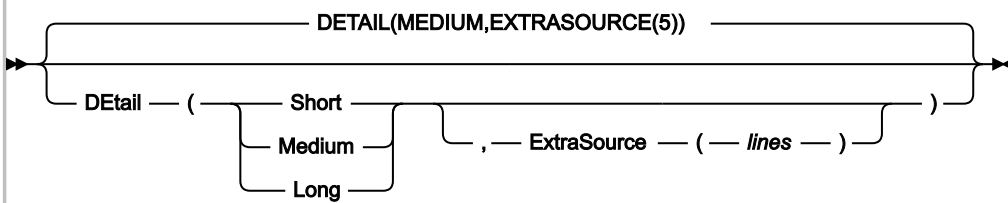
1. XDUMP が有効ではなく、DeferredReport が有効で MaxMinidumpPages 制限を超えた場合、DeferredReport オプションは指定変更されて、レポートが書き込まれます。この状態が発生した場合は、メッセージ [IDIO133W ページ 725](#) が出され、分析レポートの「Options in Effect」セクション内の DeferredReport オプションに注記が追加されます。この状態が頻繁に発生しないようにするには、XDUMP を有効にし、MaxMinidumpPages 制限を適切に設定します。



- DeferredReport オプションを有効にして書き込まれた障害項目には当初は保管レポートが含まれていませんが、Fault Entry List 画面で「V」行コマンドが初めて使用されたときに、保管レポートが追加されます。ただし、このアクションを実行するユーザーに履歴・ファイルに対する更新アクセス権限が必要です。
- このオプション (FATASKS サブオプションは除きます) は、分析制御ユーザー出口で変更または設定できます。詳しくは、[分析制御ユーザー出口 ページ 475](#)を参照してください。
- Fault Analyzer パフォーマンスを最大にするには、IDIS サブシステム内の \$\$INDEX メンバー・キャッシングを使用することも検討してください。詳しくは、[履歴・ファイル \\$\\$INDEX データのキャッシュ ページ 320](#)を参照してください。

## Detail

図 295. Syntax



Detail オプションは、障害分析レポートに組み込む詳細さのレベル (SHORT、MEDIUM、または LONG のいずれか) を指定します。

イベントのソース行の前後に、デフォルトの 5 行の追加ソース行より多く (または少なく) Fault Analyzer レポートに表示される場合は、ExtraSource(*lines*) サブオプションを使用できます。ExtraSource(*lines*) サブオプションは、詳細イベント情報セクションで示されるソース行数にのみ影響を与えます。

完全な Fault Analyzer 障害分析レポートは、以下のセクションから構成されます。

- Fault Analyzer の概要
- Fault Analyzer の要約
- 個々のイベントについての詳細情報
- コンソール・メッセージなどの、イベントには直接関連しない情報
- 異常終了するジョブの情報
- 有効な Fault Analyzer オプション

Detail オプションの指定に応じて、完全なレポートのすべてまたは一部が作成されます。

### Detail(SHORT)

- システム全体の情報を除き、レポートのすべてのセクションが組み込まれます。詳細情報は、障害の時点のイベントに対してのみ、含まれます。
- 要約された CICS® トレースが組み込まれます (該当する場合)。
- CICS Transaction Storage は含まれません。

**Detail(MEDIUM)**

- レポートのすべてのセクションが組み込まれます。ただし、障害点イベントでもある、または障害点イベントに続く最初の異常終了イベントまでのイベントに関してのみ、詳細情報が組み込まれます。このオプションはデフォルトです。
- 要約された CICS® トレースが組み込まれます (該当する場合)。
- CICS Transaction Storage は含まれません。

**Detail(LONG)**

- レポートのすべてのセクションが組み込まれ、すべてのイベントに関して詳細情報が提供されます。
- 情報を入手可能な場合、実行されるマシン命令の実行モードまたはタイプに関係なく、汎用レジスター、アクセス・レジスター、浮動小数点レジスター、ベクトル・レジスターが「イベントの詳細」セクションに表示されます。
- 全 CICS® トレースが組み込まれます (該当する場合)。
- 各ユーザー・コード・イベント汎用レジスターの周囲の 4K ストレージがミニダンプに組み込まれます。
- CICS Transaction Storage が含まれます。

このオプションは、対話式再分析には適用されません。

**DumpDSN**

異常終了により SYSMDUMP がダンプ・データ・セットに書き込まれ、ユーザーが自らの JCL を使用してダンプを再分析する場合に、この DumpDSN オプションを使用します。このオプションは、実行する障害分析の対象となるダンプ・データ・セットを指定します。

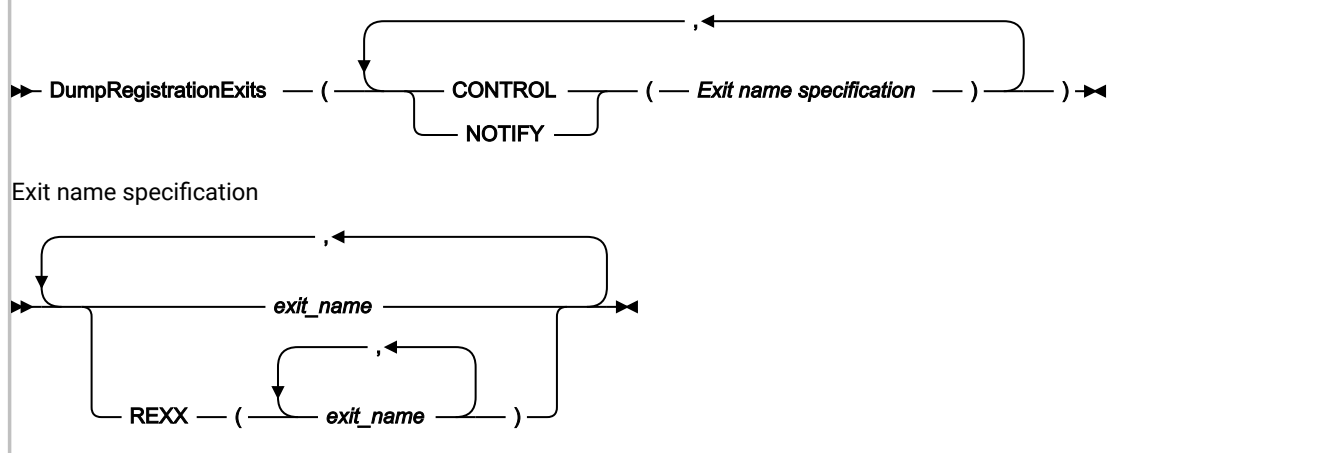
このオプションは、バッチ再分析に対してのみ適用され、IDICNFxx parmlib メンバー内に指定されると無視されます。

障害履歴・ファイルから再分析を開始した場合は、Fault Analyzer が自動的にダンプ・データ・セット名を提供します。

SYSMDUMP データ・セットが ASA プリンター制御文字を含まない場合、その論理レコード長は 4160、含む場合は 4161 である必要があります。一般的に、SDSF によって JES SPOOL から取り出された SYSMDUMP データ・セットにのみ、ASA プリンター制御文字が含まれます。

## DumpRegistrationExits

図 297. Syntax



`DumpRegistrationExits` オプションは、IDIXTSEL MVS™ ポスト・ダンプ出口処理中に呼び出されるユーザー出口のタイプと名前を指定します。

SVC ダンプ・データ・セット名が履歴・ファイル障害項目に記録されると、エンド・ユーザーが後で分析できるように、ダンプ登録分析制御および通知ユーザー出口が呼び出されます。SVC ダンプは、異常終了、または Fault Analyzer リカバリー障害記録 (recovery fault recording - RFR) 処理によって起動されている可能性があります。ただし、ダンプ登録出口は RFR 処理については起動されません。IDIXTSEL プロセスは、IDI.SIDISAM1(IDIWTSEL) サンプル ++USERMOD がインストールされていることを必要とします (詳しくは、[MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#)を参照してください)。

MVS™ SVC ダンプ登録分析制御および通知ユーザー出口は、Fault Analyzer IDIS サブシステムから実行されます。

`DumpRegistrationExits` オプションを、IDICNFxx parmlib メンバーに指定するか、または IDIOPTS DD ステートメントを介して IDIS サブシステム JCL に指定する必要があります。`DumpRegistrationExits` オプションは、IDIOPTS DD ステートメントを介してどこか他の場所 (CICS® 領域内やバッチ・ジョブ内など) に指定された場合には無視されます。

`DumpRegistrationExits` オプションの複数指定は累積されます。

出口は、REXX EXEC かロード・モジュールのどちらかです。

- REXX EXEC は次のように指定する必要があります。

```
REXX(exit_name_1, exit_name_2, ...)
```

また、IDIEEXEC DDname を介して使用可能である必要があります。

- ロード・モジュールは、標準システム検索パス (LPA、LINKLIST、または JOBLIB/STEPLIB JCL ステートメント) から使用可能である必要があります。

考えられる出口タイプは、以下のとおりです。

**CONTROL**

分析制御ユーザー出口。この出口は、有効なオプションの変更に使用できます。詳しくは、[分析制御ユーザー出口 \(MVS SVC ダンプ登録\) ページ 479](#)を参照してください。

**NOTIFY**

通知ユーザー出口。この出口を使用して、例えば、記録された障害についてインストール・システム固有の通知を示すことができます。詳しくは、[通知ユーザー出口 \(MVS SVC ダンプ登録\) ページ 507](#)を参照してください。

*exit\_name* として指定する出口名は、有効な TSO/E REXX EXEC またはロード・モジュールの名前にすることができます。ただし、一部の名前は、特別な目的のために予約されています。

**NONE**

特殊名「NONE」は「ヌル」出口を表します。この出口は呼び出されません。結果、指定されたタイプの出口を呼び出すそれ以降の試みは終了します。

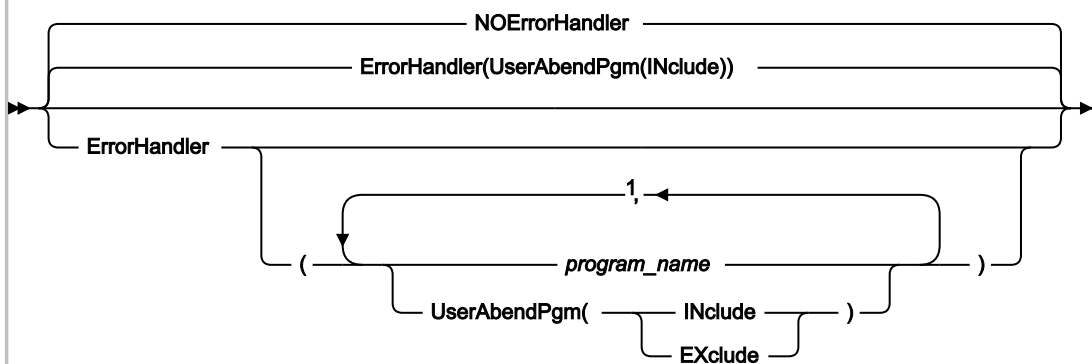
**-DROPCNF-**

特殊名「-DROPCNF-」を使用して、IDICNFxx parmlib メンバーから出口指定を廃棄します。詳しくは、[IDICNFxx parmlib メンバーのユーザー出口指定の廃棄 ページ 598](#)を参照してください。

指定した出口タイプについて、複数の出口名を指定でき、すべての出口の呼び出しが試行されます。

**ErrorHandler**

図 298. Syntax



注:

<sup>1</sup> コンマまたは空白文字のいずれかを区切り文字に使用できます。

ErrorHandler オプションを使用して、Fault Analyzer レポート内で障害点イベントとして選択すべきではない、1つ以上の共通のエラー・ハンドラー・プログラムまたは CSECT 名を指定することができます。代わりに、次に早く発生したユーザー・コード・イベントが障害点イベントになり、そのイベントが重複障害判別などに使用されます。

以下のサブオプションを指定できます。

**program\_name**

共通エラー・ハンドラーの1つ以上のプログラムまたは CSECT 名。  
これらは障害点イベントとして選択されません。

**UserAbendPgm(INclude|EXclude)**

デフォルトでは、ユーザー異常終了のイベントは障害点として選択されません。これは、UserAbendPgm(Include) を指定することと同等です。ユーザー異常終了イベントを障害点イベントとして選択できるようにするには、program\_name リストに名前が示されないことを条件に、UserAbendPgm(Exclude) を指定します。



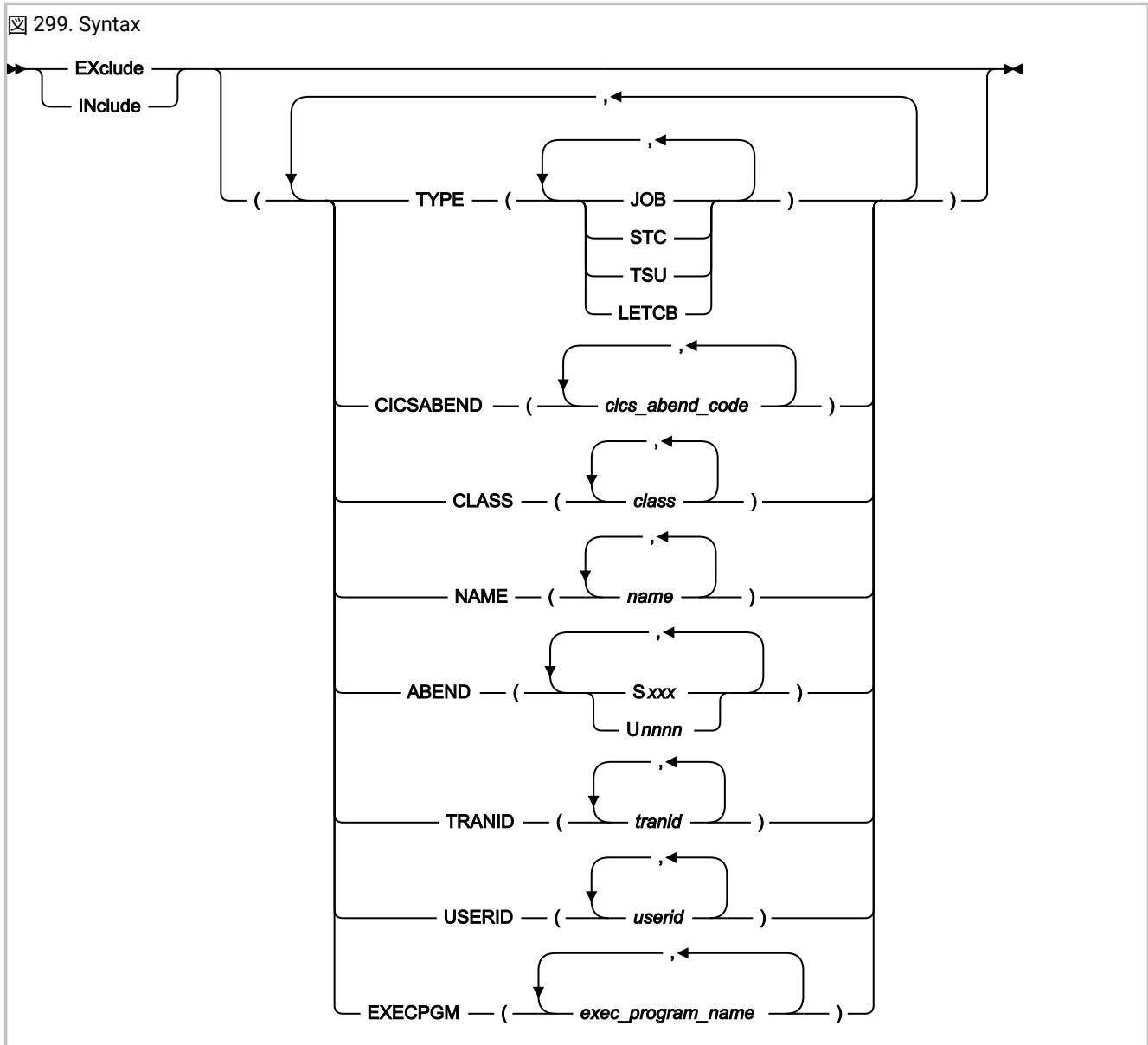
**注:** 早く発生したユーザー・コードのイベントを判別できない場合は、エラー・ハンドラー・プログラムのイベントが、依然として障害点として指定されます。

最後に指定された ErrorHandler オプションまたは NOErrorHandler オプションが有効になります。後続の指定により、直前の指定は完全にオーバーライドされます。つまり、プログラムまたは CSECT 名は累積されません。

## Exclude/Include

Exclude オプションと Include オプションは、相補的なプロセスです。これらは Fault Analyzer で処理する必要があるジョブ例外を制御するために、共通のパラメーターを共有します。このセクションでパラメーターについて学習する前に、[Exclude 処理による分析対象のジョブの制御 ページ 374](#)に説明されている Exclude/Include プロセスを読んで、理解しておく必要があります。

このセクションでは、バッチ・ジョブ、開始タスク、または TSO ユーザーのいずれかを参照するために「作業単位」という用語を使用しています。



各コマンドの説明。

### TYPE

作業単位のタイプを、JOB (バッチ・ジョブ)、STC (開始済みタスク)、または TSU (TSO ユーザー) のいずれかとして指定します。

追加のタイプ LETCB は、異常終了 TCB 用に LE

がアクティブでなければならないことを指定します。これは、アプリケーションが Language Environment® を使用する言語で書かれている場合、異常終了がシステム・タスク内で発生したのではなく、アプリケーション・タスク内で発生したことを識別するために使用できます。

**CICSABEND**

CICS® トランザクションに対する 1 つ以上の異常終了コードを *cics\_abend\_code* として指定します。各異常終了コードは、4 文字の英数字である必要があります。

テストされる異常終了コードは、トランザクションの最後の異常終了コードです。

IDIXTSEL CICS® ポスト・ダンプ出口でキャプチャーされた MVS™ システム・ダンプは、Exclude/Include オプションの影響を受けないことに注意してください。

**CLASS**

バッチ・ジョブの 1 つ以上の実行クラスを *class* として指定します。

**NAME**

1 つ以上のジョブ、タスク、または TSO ユーザーの名前を *name* として指定します。

**ABEND**

1 つ以上のシステム異常終了コードまたはユーザー異常終了コードを、次のいずれかとして指定します。

- **Sxxx**

ここで、xxx は、3 文字の 16 進数システム異常終了コード (例: S0C4) です。

- **Unnnn**

ここで、nnnn は、4 文字の 10 進数ユーザー異常終了コード (例: U4039) です。

異常終了コード値の指定が「S」または「U」で始まらず、3 文字以下である場合 (ワイルドカード文字を含む)、システム異常終了コードと見なされます。テストされる異常終了コードは、異常終了ジョブ・ステップの最後の異常終了コードです。

**TRANID**

1 つ以上の CICS® トランザクションの名前を *tranid* として指定します。

**USERID**

TSO または CICS® ユーザー ID、あるいはバッチ・ジョブ、CICS® トランザクション、または開始済みタスクを実行するユーザー ID を *userid* として指定します。

**EXECPGM**

JCL EXEC ステートメントの PGM キーワードのプログラム名を、*exec\_program\_name* として指定します。



**注:** ABEND サブオプションに置き換えられた SYSABEND サブオプションは、後方互換性のためにのみサポートされます。

異常終了タスクが Exclude 基準を満たしている場合、以降の Include 基準もそのタスクに一致していなければ、異常終了はヒストリー・ファイルに記録されず、これ以上の Fault Analyzer 処理は行われません。

指定の規則



- 個々のサブオプション、およびサブオプション内の値は、1 つ以上の空白文字、またはコンマで区切られていなければなりません。
- 基準値の指定にはワイルドカードを使用できます。サポートされているワイルドカード文字には、アスタリスク (\*) (0 個以上の文字を表す) と % 記号 (単一の必須文字を表す) があります。

基準値にワイルドカードを使用する例については [Exclude/Include ワイルドカードの例 ページ 595](#) を参照してください。

- Exclude 基準が指定されない場合、デフォルトによりすべてが包含されます。
- 基準が満たされるためには、Exclude または Include 基準のすべてのサブオプション (TYPE、CICSABEND、CLASS、NAME、ABEND、TRANID、USERID、および EXECPGM) が満足されなければなりません (論理 AND)。決して満たされることのない次のような Exclude または Include 基準を作成することも可能です。

```
Exclude(TYPE(STC) CLASS(A))
```

これらの基準が満たされることのない理由は、開始タスクは JES イニシエーター・アドレス・スペースでは実行されず、したがって特定のクラスに関連付けられないためです。

この場合、および他の Exclude 基準が (どこにも) 指定されていない場合は、ジョブは除外されません。つまり、Fault Analyzer は、発生するすべての異常終了を分析することになります。

- 複数のタイプ (JOB、STC、TSU、または LETCB)、CICS 異常終了コード、クラス、名前、異常終了コード (Sxxx または Unnn)、*tranid*、*userid*、または EXEC プログラム名の値が、単一のタイプ、CICSABEND、CLASS、NAME、ABEND、TRANID、USERID、または EXECPGM サブオプション内で指定されている場合、任意の値に一致すると、サブオプション全体が一致します (論理 OR)。
- 複数の Exclude オプションが指定されている場合、そのうちの 1 つについて基準が一致すれば、除外が行われます (ただし、一致する Include 基準が後に続かないことが前提となります)。

このオプションは、バッチまたは対話式再分析には適用されません。

このオプションは、有効なオプションを示す、障害分析レポートのセクションには含まれません。

IDIS サブシステムが始動していてデフォルトの PARM=FASTEXCLUDE' オプションが有効な場合、Include/Exclude オプションに対する変更は、Fault Analyzer IDIS サブシステムを停止して再始動した後に高速 Exclude 処理のみで有効になります。詳しくは、[高速 Exclude オプション処理 ページ 375](#) を参照してください。



**注:** すべての Include および Exclude 基準は、それ以前の Include または Exclude 基準にかかわらず、異常終了するタスクに対して検査されます。このため、parmlib 構成メンバー内および、使用可能な場合にはユーザー・オプション・ファイル内のこれらの基準の順序が、必要なインストール・システム固有のルール・セット内になっていることを確認することが重要です。例えば、クラス A で実行されるものを除く、すべてのバッチ・ジョブを除外するには、以下の順序で基準を指定します。

```
Exclude          /* This excludes everything */
Include(CLASS(A)) /* This includes batch jobs in class A only */
```



詳しくは、[Exclude 処理による分析対象のジョブの制御 ページ 374](#)を参照してください。

Fault Analyzer が障害分析をしないようにするもう 1 つの方法は、IDIOFF DD ステートメント・スイッチの使用です。詳しくは、[JCL スイッチ \(Fault Analyzer\) を使用した IDIOFF の停止 ページ 463](#)を参照してください。

## ダンプ登録処理での Exclude/Include オプションの使用

Exclude/Include オプションは、ダンプ登録処理にも適用できますが、ダンプを発生させた状態についてあまり多くのことは分からないため、非常に有用というわけではありません。この情報の不足は、ダンプがダンプ・サービス・アドレス・スペースによって書き込まれたため（詳細については[ダンプ登録処理 ページ 53](#)を参照）、Fault Analyzer が（ダンプを発行したアドレス・スペースではなく）このアドレス・スペースで起動されたことが原因です。

したがって、基本的には ENV データ域（詳細は[分析制御ユーザー出口 \(MVS SVC ダンプ登録\) ページ 479](#)を参照）に基づいて、代わりにダンプ登録分析制御ユーザー出口（[ENV - 共通出口環境情報 ページ 650](#)を参照）を使用することにより、ダンプ登録処理の包含と除外を制御することをお勧めします。

ENV データ域では、ダンプ登録処理の場合には JOB\_TYPE フィールドを「D」に設定します。

## Exclude/Include トレース情報

IDITRACE DDname をジョブに追加すると、Fault Analyzer はトレース情報を提供できます。この情報は、特定の障害がリアルタイム分析に組み込まれる、または除外される原因を理解するのに役立ちます。例:

```
//IDITRACE DD SYSOUT=*
```

(CICS® 環境でこのトレースを有効にする別の方法については、[IDITRACE CICS において ページ 407](#)を参照してください。)

トレース情報は、IDITRACE DDname 宛先に書き込まれます。Exclude/Include オプション処理トレースの例は、次のとおりです。

図 300. Exclude/Include オプション処理トレースの例

```
EXCLUDE/INCLUDE option criterion match values from abending job:
  ABEND. . . . . : S0CB
  CICSABEND. . . . . : n/a
  CLASS. . . . . : A
  EXECPGM. . . . . : TEST1
  NAME . . . . . : ICC20010
  TRANID . . . . . : n/a
  TYPE . . . . . : JOB,LETCB
  USERID . . . . . : FRED
  Parmlib config member EXCLUDE(TYPE(TSU)) option did not match; Exclude/include status is: INCLUDE
  Parmlib config member INCLUDE(USERID(FRED)) option matched; Exclude/include status is: INCLUDE
  Parmlib config member EXCLUDE(TYPE(STC) NAME(WTAM DFHSM)) option did not match; Exclude/include status is:
  INCLUDE
  Parmlib config member EXCLUDE(SYSABEND(*37)) option did not match; Exclude/include status is: INCLUDE
```

IDIS サブシステム FastExclude 処理によって障害が除外されている場合、EXCLUDE/INCLUDE トレース情報は使用できません。

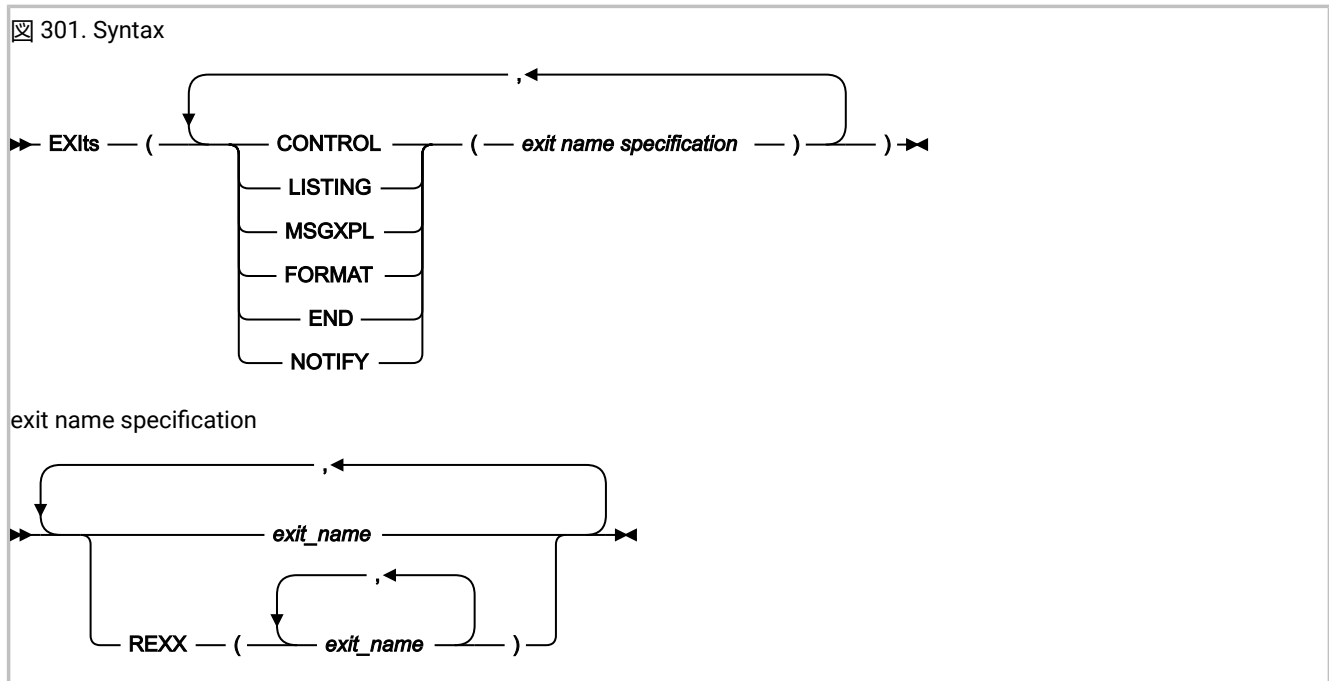
## Exclude/Include ワイルドカードの例

ワイルドカードを使用する Exclude/Include 基準値の例を、以下に示します。

表 15. ワイルドカードの例

基準値	比較値	一致
ABCD*	ABC	なし
ABCD*	ABCD	あり
ABCD*	ABCDE	あり
A*CD	ABC	なし
A*CD	ABCD	あり
A*CD	ABCDE	なし
A*DE	ABCDE	あり
A**DE	ABCDE	あり
A%DE	ABCDE	なし
AB%DE	ABCDE	あり
%B*	ABCDE	あり

## Exits



Exits オプションは、Fault Analyzer 実行中に呼び出されるユーザー出口のタイプおよび名前を指定します。指定した出口タイプについて、複数の出口名を指定でき、すべての出口の呼び出しが試行されます。



**注:** IDIUTIL バッチ・ユーティリティーに対するユーザー出口の指定に関する詳細は、[EXITS 制御ステートメント ページ 450](#)を参照してください。

Exits オプションの複数指定は累積されます。

Exits は REXX EXEC またはロード・モジュールのいずれかです (フォーマット・ユーザー出口では、REXX が唯一のサポートされるプログラミング言語であることに注意してください)。

- REXX EXEC は次のように指定する必要があります。

```
REXX(exit_name_1, exit_name_2, ...)
```

また、IDIEEXEC DDname を介して使用可能である必要があります。

- ロード・モジュールは、標準システム検索パス (LPA、LINKLIST、または JOBLIB/STEPLIB JCL ステートメント) から使用可能である必要があります。

考えられる出口タイプは、以下のとおりです。

#### CONTROL

分析制御ユーザー出口。この出口を使用して、有効なオプションを変更、または障害を分析から除外できます。詳しくは、[分析制御ユーザー出口 ページ 475](#)を参照してください。

#### LISTING

Compiler Listing Read ・ユーザー出口。この出口を使用して、使用可能な MVS™ PDS (または PDSE) データ・セットに含まれるコンパイラー・リストまたは Fault Analyzer サイド・ファイル以外のソースから、ソース・コード情報を取得できます。詳しくは、[コンパイラー・リスト読み取りユーザー出口 ページ 480](#)を参照してください。

#### MSGXPL

Message and Abend Code Explanation

・ユーザー出口。この出口を使用して、メッセージおよび異常終了コードの説明を示し、Fault Analyzer が提供するそれらの説明を補足または置換できます。詳しくは、[メッセージおよび異常終了コード説明ユーザー出口 ページ 485](#)を参照してください。

#### FORMAT

フォーマット・ユーザー出口。この出口を使用して、分析レポートにユーザー固有の情報を追加できます。詳しくは、[フォーマット・ユーザー出口 ページ 491](#)を参照してください。

#### END

End Processing ・ユーザー出口。このユーザー出口を使用して、MVS™ システム・ダンプ、Fault Analyzer ミニダンプ、または障害ヒストリー・ファイル項目全体の抑止を要求できます。詳しくは、[終了処理ユーザー出口 ページ 494](#)を参照してください。

**NOTIFY**

通知ユーザー出口。この出口を使用して、例えば、記録された障害についてインストール・システム固有の通知を示すことができます。詳しくは、[通知ユーザー出口 ページ 499](#)を参照してください。

`exit_name` として指定する出口名は、有効な TSO/E REXX EXEC またはロード・モジュールの名前にすることができます。ただし、一部の名前は、特別な目的のために予約されています。

**NONE**

特殊名「NONE」は「ヌル」出口を表します。この出口は呼び出されません。結果、指定されたタイプの出口を呼び出すそれ以降の試みは終了します。

**-DROPCNF-**

特殊名「-DROPCNF-」を使用して、IDICNFxx parmlib メンバーから出口指定を廃棄します。詳しくは、[IDICNFxx parmlib メンバーのユーザー出口指定の廃棄 ページ 598](#)を参照してください。

Exits オプションを介して1つ以上の出口が指定されている場合は、出口に関する情報が分析レポートの“有効になっているオプション”セクションに書き込まれます。このセクションでは、指定されたすべての出口がリストされ、それぞれのタイプの中で呼び出された出口がある場合、それらの出口が示されます。



**注:** 通知ユーザー出口は分析レポートの完了後に呼び出されるため、これらの出口の呼び出し状況は入手できません。

分析レポートの“有効になっているオプション”セクションに書き込まれる情報の例は、以下のとおりです。

```
Exits:

The following user exits were specified via Exits options.

Type      Name      Type Invoked
-----
CONTROL   CTLEXITA  LMOD No - module not found
          CTLEXITB  REXX Yes
          CTLEXITC  REXX Yes
          CTLEXITD  LMOD No - module not found
END        ABC1      LMOD Yes
NOTIFY    NONE      n/a (Not attempted)
```

この例には、以下が示されています。

- 4つの分析制御ユーザー出口が指定されました。1番目 (CTLEXITA) は、呼び出すことができなかったロード・モジュールです。2番目に指定されたユーザー出口 (CTLEXITB) は、呼び出された REXX EXEC です。2番目のユーザー出口の呼び出しが成功したため、3番目 (CTLEXITC) または 4番目 (CTLEXITD) のユーザー出口の呼び出しは試行されませんでした。
- 1つのEnd Processingユーザー出口 (ABC1) がロード・モジュールとして指定されました。このユーザー出口は呼び出されました。
- ヌルの通知ユーザー出口が指定されました。これは呼び出されません。

## IDICNFxx parmlib メンバーのユーザー出口指定の廃棄



**注:** このセクションの情報は、Exits と DumpRegistrationExits の両方のオプションに適用されます。

特殊な出口名 `-DROPCNF-` を使用して IDICNFxx parmlib メンバーに指定された出口タイプへの、すべてのユーザー出口指定を廃棄できます。

例えば、IDICNFxx parmlib メンバーが以下を含むとします。

```
Exits(Control(FRED1,FRED2))
```

また、IDIOPTS ユーザー・オプション・ファイルが以下を含むとします。

```
Exits(Control(FRED3,-DROPCNF-,FRED4))
```

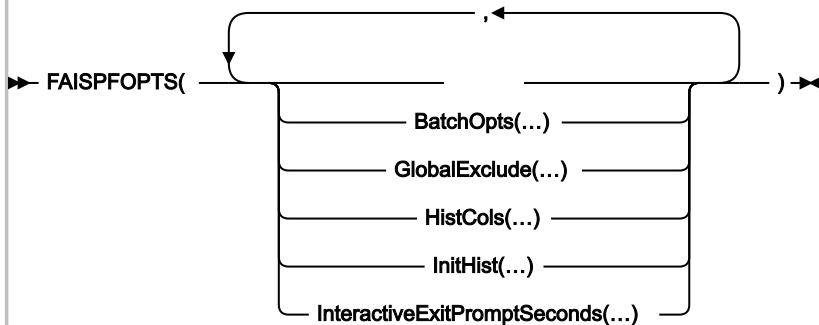
この場合、分析制御ユーザー出口の最終論理連結は、次のようになります。

Type	Name
CONTROL	FRED3
	FRED4

`-DROPCNF-` 出口名は、IDICNFxx parmlib メンバーを除いて、Exits オプションを指定できる場所であれば、どこにでも指定できます。

## FAISPFopts

図 302. Syntax



FAISPFopts オプションは、Fault Analyzer ISPF インターフェースのオプションを指定するために使用されます。以下を参照してください。

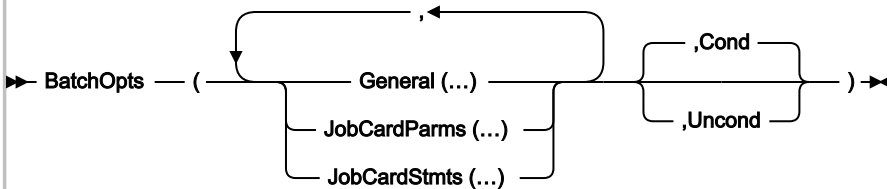
- [BatchOpts ページ 120](#)
- [GlobalExclude ページ 603](#)
- [HistCols ページ 604](#)
- [InitHist ページ 604](#)
- [InteractiveExitPromptSeconds ページ 605](#)

このオプションは、有効なオプションを示す、障害分析レポートのセクションには含まれません。

## BatchOpts

BATCHOPTS サブオプションを使用して、Batch Reanalysis Options 画面でデフォルト設定の代替設定を指定します。[バッチ再分析オプション ページ 154](#) を参照してください。

図 303. Syntax



### BatchOpts の COND および UNCOND サブオプション

BATCHOPTS サブオプションの各レベルで、COND および UNCOND サブオプションを指定できます:

- 最高レベル (BATCHOPTS サブオプション)
- 中間レベル (GENERAL、JOB CARDPARMS および JOB CARDSTMTS サブオプション)
- 最低レベル (JCLEDIT や MSGCLASS などの個々の設定)

COND (条件付き) では、ユーザーが別の設定をまだ指定していない場合、選択済みの値が画面上でデフォルト設定として使用されます。そのため、はじめてのユーザーにはこちらが適用されます。また、ユーザーが画面上で設定をクリアした場合にもこちらが毎回適用されます。

UNCOND (無条件) では、以前の設定に関係なく、任意の Fault Analyzer ISPF インターフェース・セッションの開始時に、選択済みの値が画面の現在の設定として使用されます。セッション中に、設定をその他の有効な値に変更することができません。この値は、次のセッションまで有効になります。

より高いレベルで UNCOND が指定されていない限り、すべてのレベルで COND がデフォルトになります。つまり、上位レベルで指定した内容は、デフォルトでそれぞれの下位レベルに伝搬されます。

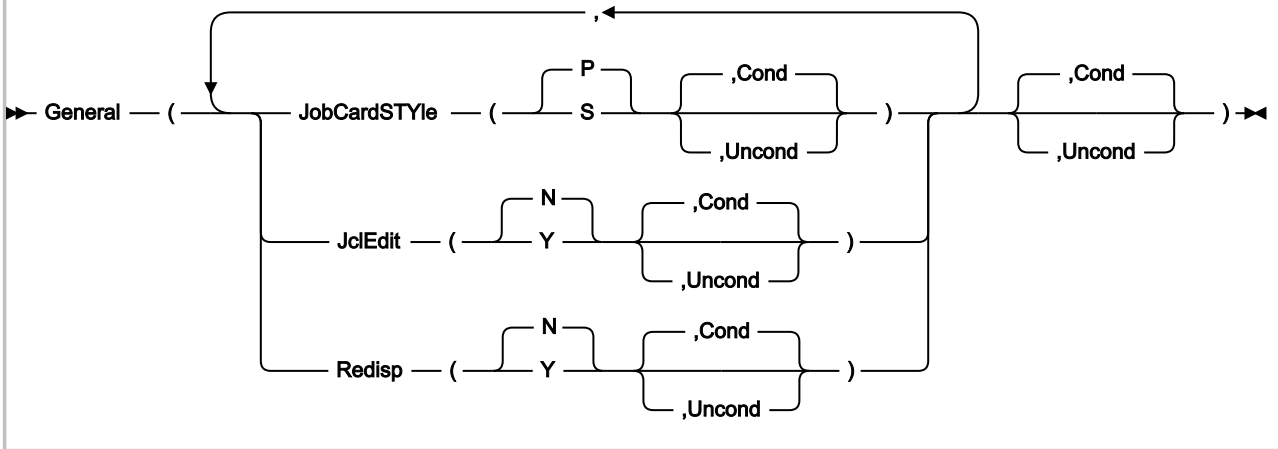
次の例を考えてみます。

```
FAISPFOPPTS(BATCHOPTS(GENERAL(JOBCARDSTYLE(P),JCLEDIT(Y),JOB CARDSTMTS(<stmt1>,<stmt2>),
JOB CARDPARMS(MSGCLASS(A),COND),UNCOND))
```

- `JOB CARDPARMS(MSGCLASS(A))` は条件付きです。
- `GENERAL(JOBCARDSTYLE(P), JCLEDIT(Y), JOB CARDSTMTS(<stmt1>, <stmt2>))` は無条件です。

## BatchOpts の一般サブオプション

図 304. Syntax



GENERAL サブオプションは、Batch Reanalysis Options 画面の「General Options」セクションに適用されます。

**JOBCARDSTYLE(P |S [,COND |UNCOND]))**

**JCSTY(P |S [,C |U])**

Batch Reanalysis Options 画面の「Job card style」フィールドに次の値を指定します。

**P**

パラメーター

**S**

ステートメント

指定していない場合、デフォルトは P です。

**JCLEEDIT(Y |N [, COND |UNCOND]))**

**JE(Y |N [, C |U])**

Batch Reanalysis Options 画面の「Display panel to edit generated JCL」フィールドに次の値を指定します。

**Y**

あり

**N**

なし

指定していない場合、デフォルトは N です。

**REDISPLY(Y |N [, COND |UNCOND]))**

**R(Y |N [, C |U])**

Batch Reanalysis Options 画面の「Redisplay this panel before each reanalysis」フィールドに次の値を指定します。



Y

あり

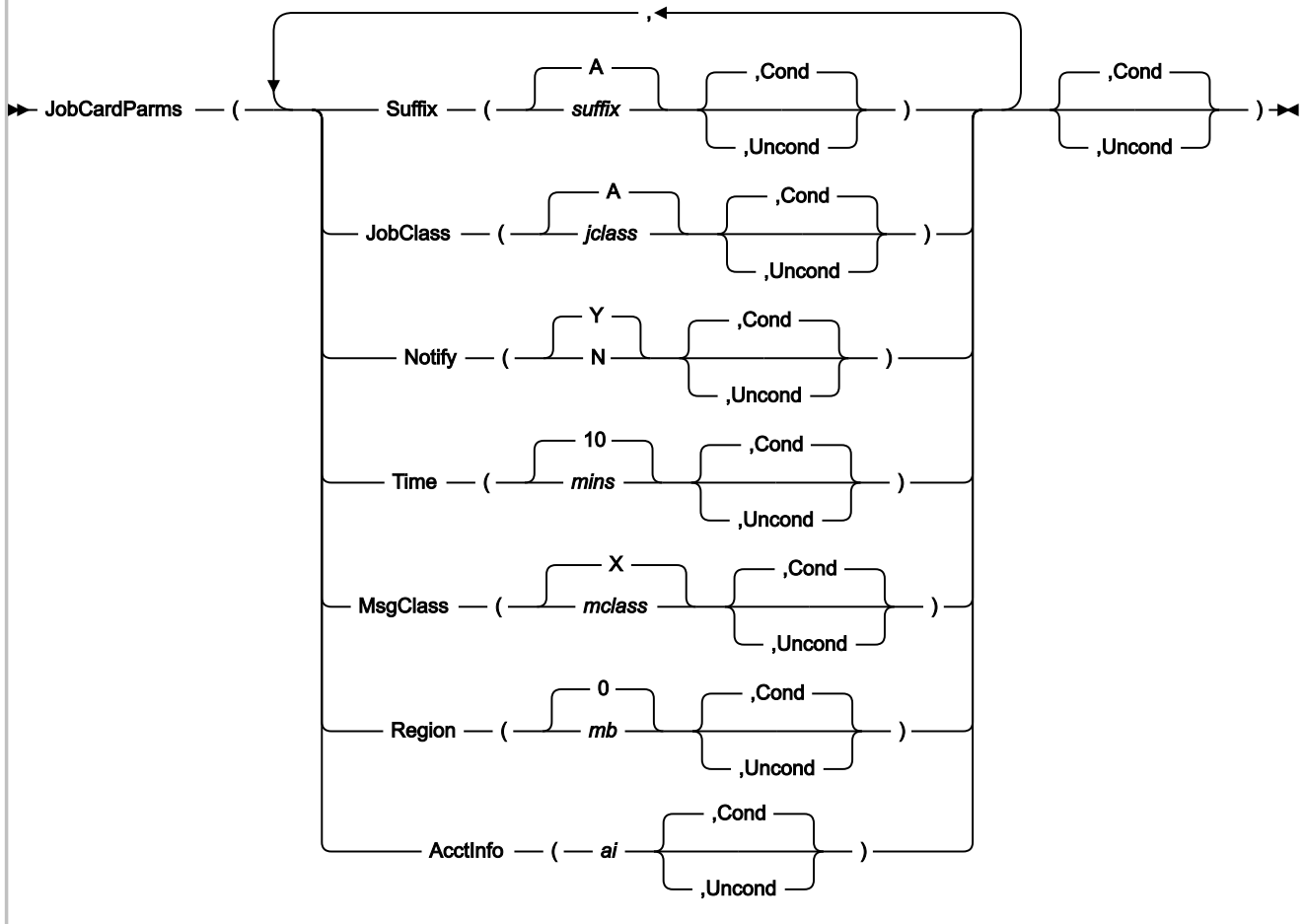
N

なし

指定していない場合、デフォルトはNです。

### BatchOpts JobCardParms サブオプション

図 305. Syntax



JOBCARDPARMS サブオプションは、Batch Reanalysis Options 画面の「**Job Card Parameters**」セクションに適用されます。

**SUFFIX( *suffix* [, COND |UNCOND])**

**S( *suffix* [, C |U])**

Batch Reanalysis Options 画面の「**Job name suffix**」フィールドの値を指定します:

***suffix***

A ~ Z、0 ~ 9、@、# または \$

指定していない場合、デフォルトは A です。

**JOBCLASS( *jclass* [, COND |UNCOND])**

**JC( *jclass* [, C |U])**

Batch Reanalysis Options 画面の「**Job class**」フィールドに次の値を指定します。

***jclass***

A ~ Z または 0 ~ 9

指定していない場合、デフォルトは A です。

**NOTIFY(Y |N [, COND |UNCOND])**

**N(Y |N [, C |U])**

Batch Reanalysis Options 画面の「**Job notify**」フィールドに次の値を指定します。

**Y**

あり

**N**

なし

指定していない場合、デフォルトは Y です。

**TIME( *mins* [, COND |UNCOND])**

**T( *mins* [, C |U])**

Batch Reanalysis Options 画面の「**Job time minutes**」フィールドに次の値を指定します。

***mins***

0 から 99。

指定していない場合、デフォルトは 10 です。

**MSGCLASS( *mclass* [, COND |UNCOND])**

**MC( *mclass* [, C |U])**

Batch Reanalysis Options 画面の「**Message class**」フィールドに次の値を指定します。

***mclass***

A ~ Z または 0 ~ 9

指定していない場合、デフォルトは X です。

**REGION( *mb* [, COND |UNCOND])**

**R( *mb* [, C |U])**

Batch Reanalysis Options 画面の「**Region megabytes**」フィールドに次の値を指定します。

**mb**

0 から 2047。

指定していない場合、デフォルトは 0 です。

**ACCTINFO( ai [, COND |UNCOND])****AI( ai [, C |U])**

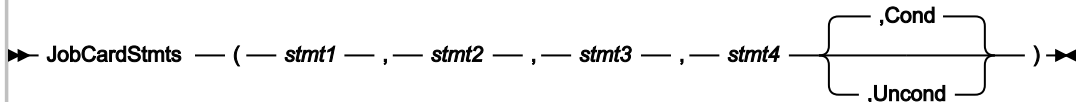
Batch Reanalysis Options 画面の「Accounting info」フィールドに次の値を指定します。

**ai**

最大 50 文字。

**BatchOpts JobCardStmts サブオプション**

図 306. Syntax



最大 4 つのコンマ区切り JCL ステートメントを指定できます。

- ステートメントが指定されていない場合 (例えば、3 つのステートメントのみが必要な場合) でも、省いたステートメントの区切りコンマを指定する必要があります。
- ステートメントに空白またはコンマが含まれている場合は、一重引用符または二重引用符で囲む必要があります。ステートメントに同じタイプの引用符が既に含まれている場合は、その引用符を 2 つ重ねます。

例えば、次のような JCL ステートメントを作成するには:

```
//MYJOB JOB (123), 'X Y Z',
```

以下の項目を指定できます:

```
'//MYJOB JOB (123), "X Y Z",'
```

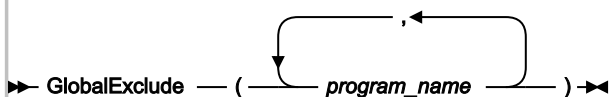
```
"//MYJOB JOB (123), 'X Y Z',"
```

```
'//MYJOB JOB (123), 'X Y Z','
```

最後の例では、2 つの一重引用符が xyz を囲んでいます。

**GlobalExclude**

図 307. Syntax



GlobalExclude オプションを使用して、インストール済み環境全体のプログラム名のデフォルト・リストを指定できます。これらについては、対話式再分析中にサイド・ファイルのプロンプトは行われません。

プログラム名は有効な PDS または PDSE メンバー名でなければなりません。ただし、プログラム名にはワイルドカード文字「\*」（ゼロ個以上の文字）および「%」（単一の必須文字）を使用できます。

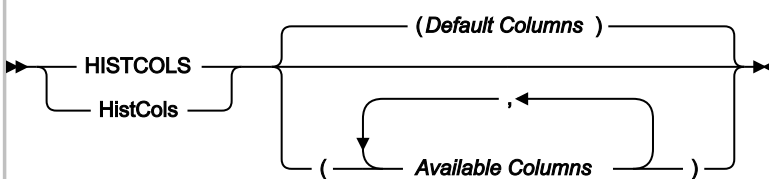
指定するプログラム名の大/小文字は区別されません。

有効なプログラム名の指定例を以下に示します。

```
*XMAI*
PAYROLL0
SELOPT%
SUBRTN*
TZ%%C*
```

## HistCols

図 308. Syntax

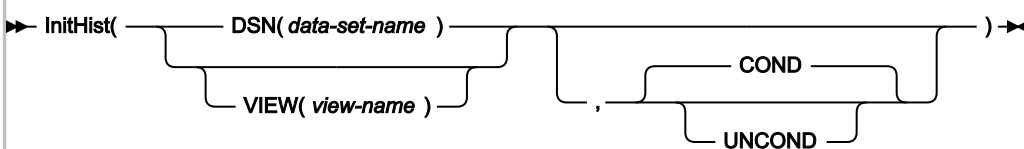


HistCols オプションでは、Fault Analyzer ISPF インターフェースの Fault Entry List 画面に表示される情報の列をカスタマイズできます。このオプションを IDICNFxx 構成メンバーに追加すると、インストールによって、カスタマイズした Fault Entry List 画面をすべてのユーザーに提供できます。個々のユーザーは、このオプションの指定にかかわらず、ユーザー自身の Fault Entry List 画面を変更できます。

Fault Entry List 画面のカスタマイズの詳細および HistCols オプションを使用して指定できる個々の列の説明については、[Fault Entry List の列構成 ページ 71](#) を参照してください。

## InitHist

図 309. Syntax



InitHist オプションを使用すれば、Fault Analyzer ISPF インターフェースのすべてのユーザーが特定のヒストリー・ファイル/ビューを最初に表示するように指定できます。

**DSN(*data-set-name*)**

これは、初期 ISPF  
インターフェース表示に使用される履歴・ファイルのデータ・セット名を指定します。

**VIEW(*view-name*)**

これは、初期 ISPF インターフェース表示に使用される IDVIEWS  
連結内のビュー・メンバーの名前を指定します。

**COND**

ユーザーが別の履歴・ファイル/ビューを以前に選択したことがない場合に限り、指定された履歴  
ファイル/ビューが、Fault Analyzer ISPF インターフェースの開始時に表示される初期 Fault Entry List  
画面に使用されます。

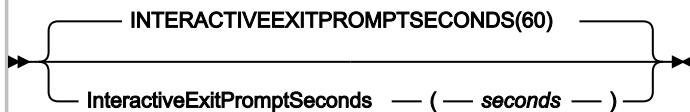
この値はデフォルトです。

**UNCOND**

ユーザーが別の履歴・ファイル/ビューを以前に選択したことがあるかどうかに関係なく、指定された  
履歴・ファイル/ビューが、Fault Analyzer ISPF インターフェースの開始時に表示される初期 Fault  
Entry List 画面に必ず使用されます。

**InteractiveExitPromptSeconds**

図 310. Syntax



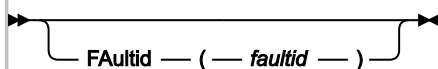
InteractiveExitPromptSeconds オプションは、対話式レポート画面の終了時に終了のプロンプト・パネルが表示される前  
に、対話式再分析が必要とする経過時間の最小秒数を指定します。対話式再分析が指定した秒数未満の経過時間を使用し  
た場合、プロンプトは表示されません。代わりに、ユーザーは Enter キーを押して、対話式レポートの終了を確定する必要  
があります。

有効な秒数の範囲は 0 から 99999 です。

- 0 を指定する場合、無条件にプロンプトが表示されます。
- 99999 を指定する場合、プロンプトは表示されません。

**FaultID**

図 311. Syntax



FaultID オプションは、割り当てられた障害 ID (*faultid*) により、履歴・ファイル項目を識別します。このオプションは、障害再分析を実行するときに使用されます。障害 ID は次のフォーマットである必要があります。

```
cccnnnn
```

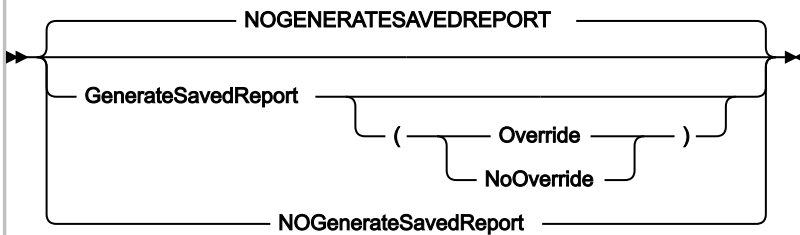
ここで、ccc は、割り振られた履歴・ファイル接頭部 (1 から 3 文字) であり、nnnn は 5 桁の 10 進数の障害番号です。



1. 障害 ID、F00000 は、ヌル ID を示します。これを使用して再分析用の履歴・ファイル項目を選択できません。使用可能な障害番号が存在しない場合は、この ID を障害項目に割り当てることができます。
2. FaultID オプションのデフォルト設定値は適切ではありません。履歴・ファイル障害項目のバッチ再分析用に独自の JCL を作成する場合のみ、使用してください。Fault Analyzer ISPF インターフェースを使用して再分析を実行する場合、このオプションは自動的に正しく使用されます。

## GenerateSavedReport

図 312. Syntax



GenerateSavedReport オプションを使用すると、再分析中に保管レポートを障害項目に作成したり、または置換したりすることができます。

Override サブオプションを使用すると、既に存在しているかどうかにかかわらず、新規の保管レポートが生成されます。

NoOverride サブオプションを使用すると、既に存在していない場合のみ保管レポートが生成されます。この値は、GenerateSavedReport オプションがサブオプションなしに指定される場合のデフォルトです。

GenerateSavedReport オプションが有効な場合、リアルタイム処理中に使用されるソース情報データ・セット (コンパイラー・リストまたはサイド・ファイル) が再分析中に自動的に含まれます。

このオプションの実用的な応用として、新規に作成された障害項目の自動バッチ再分析に、このオプションを使用することができます。この障害項目には、パフォーマンス上の理由でリアルタイム処理に DeferredReport オプションが有効であったため、保管レポートが含まれていません。例えば、バッチ再分析のようなジョブを通知ユーザー出口から実行依頼することにより、「Fault Entry List」画面からユーザーが「V」または「S」の行コマンドを出す前に、保管レポートが作成されるようにできます。この目的のために使用可能な通知ユーザー出口のサンプルが、[例 4 ページ 506](#) で提供されました。

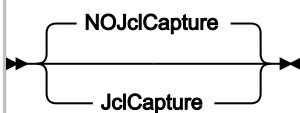
GenerateSavedReport オプションが、DumpDSN オプションを使用して指定した MVS™ ダンプ・データ・セットのバッチ分析で 사용되는場合、ユーザー・アクセス許可に応じて、このオプションは、現行履歴・ファイルに障害項目を作成します。障害項目をデフォルトの履歴・ファイル以外の履歴・ファイル上に作成する場合、履歴・ファイルの指定は、IDIHIST DD ステートメントの指定、または DataSets(IDIHIST(*dsm*)) オプションの使用などで可能です。障害項目の作成が成功すると、メッセージ [IDI0164I ページ 733](#) が出されます。

## Include

[Exclude/Include ページ 590](#) を参照してください。

## JclCapture

図 313. JclCapture syntax



JCLcapture オプションを使用して、Fault Analyzer が異常終了ジョブの JCL を取得し、それを障害項目に保存するように指定します。JES2 のみがサポートされています。正常に実行されると、対話式再分析レポートの「Abend Job Information」セクションのポイント・アンド・シュート・フィールドを介して JCL が使用可能になります。

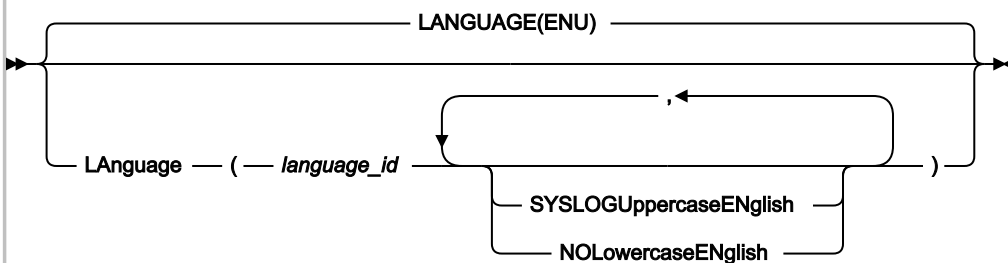
デフォルトは NOJclCapture です。

このオプションは、リアルタイム分析にのみ適用できます。他のすべてのモードの実行では、これは無視されます。

このオプションは、JCLcapture を用いたリアルタイム分析が有効である場合にのみ、有効なオプションを示す障害分析レポートのセクションに含まれます。

## Language

図 314. Syntax



Language オプションは、適切な言語依存メッセージの選択に使用する、各国語 ID を指定します。

以下の言語 ID を使用できます。

**ID**

**Language**

**ENU**

米国英語 (デフォルト)

**JPN**

日本語 (Fault Analyzer の日本語フィーチャーがインストールされている場合にのみ使用可能)

オプションで、以下のサブオプションを指定できます。

**SyslogUppercaseEnglish**

可能な場合、WTO メッセージはすべて大文字の英語のみとなります。

このサブオプションは SYSLOGUEN と省略できます。

**NoLowercaseEnglish**

可能な場合、Fault Analyzer

によって書き込まれるすべての出力の英語の小文字は大文字に変換されます。このサブオプションは、例えば、「KANJI」モードで 3270 端末装置で使用すると、小文字の英文字が読めないことを防止できます。

このサブオプションは NOLEN と省略できます。

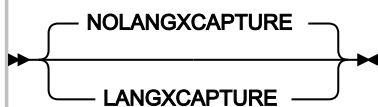
Language オプションが IDICNFxx parmlib 構成メンバーで指定されている場合は、このオプションを最初に指定する必要があります。これにより、Fault Analyzer は最大数の言語依存メッセージを発行できるようになります。



**注:** Fault Analyzer IDIS サブシステムを使用している場合、異常終了、メッセージ、理由コードなどの説明に関連するものについては、ユーザーが指定した Language オプションはいずれも無視されます。これらの説明は常に、IDIS サブシステム用に有効になっている Language オプションに従って提供されます。これは、パフォーマンス向上と入出力削減を目的として、IDIS サブシステムを使用して説明をキャッシュに入れるためです。

**LangxCapture**

図 315. Syntax



LangxCapture オプションでは、コンパイラー・リストまたはサイド・ファイルからのソース・レベルのデバッグ情報を障害項目に書き込みます。ソース・レベルのデバッグ情報を障害項目に取り込むと、コンパイラー・リストまたはサイド・ファイルは不要になります。

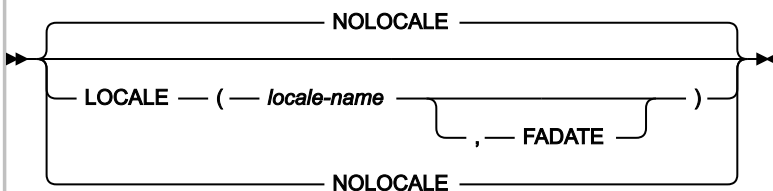
DeferredReport オプションまたは NoSource オプションが指定されている場合、このオプションは無効です。一致しないサイド・ファイルは取り込まれません。LangxCapture オプションは、対話式再分析または LangxCapture を用いたリアルタイム分析が有効である場合にのみ、有効なオプションを示す障害分析レポートのセクションに含まれます。

対話式再分析レポート内で、「サービス」メニューから現在の障害項目のキャプチャーされたソースを表示できます ([Services-> LANGP Side File Formatting Utility ページ 102](#)を参照)。据え置きブレークポイントも設定できます。



## Locale

図 316. Syntax



Locale オプションは、国/地域別環境に対応した表示に使用されるロケールを指定します。影響を受けるのは、日時フォーマット、ソートされた情報の照合シーケンス、およびピリオドとして表示される非印字文字の判別などです。

*locale-name* として指定されるロケール名は、z/OS® C/C++ で `setlocale()` ランタイム関数に対して指定されるロケール名のいずれかにすることができます。ロケール名のリストについては、z/OS® C/C++ プログラミング・ガイドの『付録 D. z/OS® C/C++ で提供されるロケール』を参照してください。

オプションの FADATE サブオプションが指定された場合、使用されるロケールにかかわらず、表示されるすべての日付は Fault Analyzer の標準フォーマット YYYY/MM/DD で、すべての時間値は Fault Analyzer の標準の 24 時間フォーマット HH:MM:SS です。

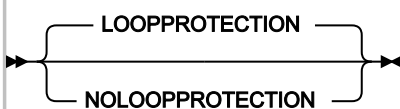
NoLocale オプションの指定は、Locale(C,FADATE) の指定と同じです。



**注:** *locale-name* の Fault Analyzer 指定は、IPVCNF00 parmlib メンバーの IBM Application Delivery Foundation for z/OS Common Components Locale オプション指定をオーバーライドします。IPVCNF00 parmlib メンバーについて詳しくは、*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*を参照してください。

## LoopProtection

図 317. Syntax



このオプションは、Fault Analyzer ループ/待機保護機構を活動化または非活動化するために使用します。この機能がアクティブである場合 (デフォルト状態)、Fault Analyzer は、複数の内部チェックポイントのいずれかで最大予想経過時間を超えると、処理を終了します。この終了が発生した場合は、メッセージ [IDI0092S ページ 715](#) が出力されます。

ユーザー出口は、ENV.LOOPPROTECTION\_OPT フィールドを N に設定することによって、ループ/待機保護機能を非活動化できます。このフィールドの詳細については、[ENV・共通出口環境情報 ページ 650](#)を参照してください。



**注:** NoLoopProtection オプションが有効な場合でも、オプションが無効の場合にはメッセージ [IDI0092S ページ 715](#) で終了する可能性のある分析に関しては、正常終了する保証はありません。この原因は、ジョブが依然と



して、通常のインストール済み環境で課せられる MVS™ 実行時間制限の対象になる可能性があること、または結局ジョブを取り消すことが必要になる無限ループまたは待ち状態にあることです。

このオプションは、リアルタイム分析にのみ適用できます。他のすべてのモードの実行では、これは無視されます。

このオプションは、リアルタイム分析で NoLoopProtection が有効である場合に、有効なオプションを示す障害分析レポートのセクションにのみ含まれます。

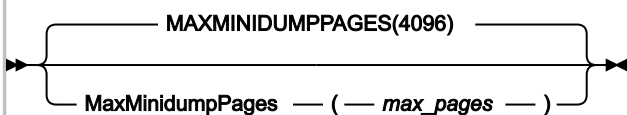


**注:** リアルタイム分析中に *max\_pages* が超過したが、Fault Analyzer が XDUMP に十分なページをスプिलできた場合、MaxMinidumpPages を超える状況にはなりません。

## MaxMinidumpPages

MaxMinidumpPages オプションは、ミニダンプとして各ヒストリー・ファイルの障害項目に保管できる 4K ページの最大数を指定します。

図 318. Syntax



XDUMP データ・セットの使用が有効化された場合、リアルタイム分析中に参照される一部のストレージ・ページが、関連する XDUMP データ・セットに書き込まれることがあり、残りは障害項目ミニダンプに書き込まれます。XDUMP データ・セットの使用が有効化されていない場合、すべての参照されるページは障害項目ミニダンプに書き込まれます。

障害項目ミニダンプは、Fault Analyzer のリアルタイム処理中に作成された障害項目の再分析を実行するために必要です。

ミニダンプを宛先とする参照されたページ数が *max\_pages* を超える場合、保管済みレポートを持つ障害項目は作成されますが、それにミニダンプは含まれず、関連する XDUMP データ・セットもありません。



**注:** リアルタイム分析中に *max\_pages* が超過したが、Fault Analyzer が XDUMP に十分なページをスプिलできた場合、MaxMinidumpPages 超えた状況は存在しません。



**注:** フォーマット・ユーザー出口の実行中に参照されたストレージによって生じたミニダンプの追加ページは、このオプションで検査されるミニダンプのページ数には含まれません。また、終了処理ユーザー出口への入力として提供されるミニダンプのページ数にも含まれません。

MaxMinidumpPages オプションによって、ストレージまたは DASD スペースが事前割り振りされることはありません。これは、ヒストリー・ファイル障害項目で、指定されたサイズを超えたミニダンプが書き込まれるのを防止するための、単なるカットオフ・メカニズムです。このオプションは、GenerateSavedReport オプションが有効になっている MVS® ダンプ・データ・セットのリアルタイム処理またはバッチ再分析に適用されます。

有効なオプションを表示する障害分析レポートのセクションには、このオプションが含まれます。さらに、制限を超えているかどうか、超えている場合には何ページ超えているのかが示されます。制限を超えた場合、メッセージ [IDIO052I ページ 706](#) が発行されます。

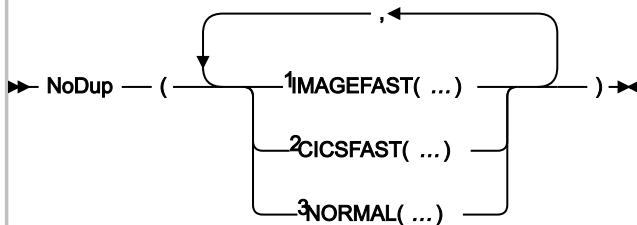
有効な `max_pages` の範囲は 0 から 524288 です。



**注:** 分析制御またはEnd Processingユーザー出口を使用すると、有効な `MaxMinidumpPages` オプションを事実上オーバーライドする場合があります。

## NoDup

図 319. Syntax



注:

- <sup>1</sup> 詳しくは、[NoDup\(ImageFast\(...\)\) ページ 613](#)を参照してください。
- <sup>2</sup> 詳しくは、[NoDup\(CICSFAST\(...\)\) ページ 616](#)を参照してください。
- <sup>3</sup> 詳しくは、[NoDup\(NORMAL\(...\)\) ページ 619](#)を参照してください。

NoDup オプションは、重複障害検出のしきい値を指定します。

重複障害検出のタイプは実行環境によってまったく異なります。

### • 高速重複検出:

CICS® または IMS™ に適用可能:

- CICS® 高速重複検出は、NoDup オプションの CICSFAST サブオプションによって制御されます ([NoDup\(CICSFAST\(...\)\) ページ 616](#)を参照)。

重複検出の有効範囲は、単一 CICS® 領域に制限されます。

- IMS™ 高速重複検出は、NoDup オプションの ImageFast(IMS™) サブオプションによって制御されます ([NoDup\(ImageFast\(...\)\) ページ 613](#)を参照)。

重複検出の有効範囲は、MVS™ イメージ全体に及びます。

高速重複検出の主な目的は、指定された期間内で許可する特定の問題の障害分析を 1 回に制限することで、比較的短い期間内に複数回発生する同じ異常終了によって 異常終了リカバリーが不必要にスローダウンするのを回避する

ことです。重複障害の検出は障害分析の前に行う必要があるため、検出の基準は、通常のリバース検出で使用する基準とは異なります。

「高速」重複検出の規則で重複と見なされなかった障害は、その後の「通常」重複検出の対象になります。

• **通常重複検出:**

すべての実行環境に適用できます。

NoDup オプションの NORMAL サブオプションによって制御されます ([NoDup\(NORMAL\(...\)\) ページ 619](#)を参照)。

通常重複検出の目的は、同じ障害がヒストリー・ファイルに別々に複数回記録されないようすることであり、これにより、DASD スペースが節約され、ヒストリー・ファイルに含まれる項目が固有の問題を表すようになります。

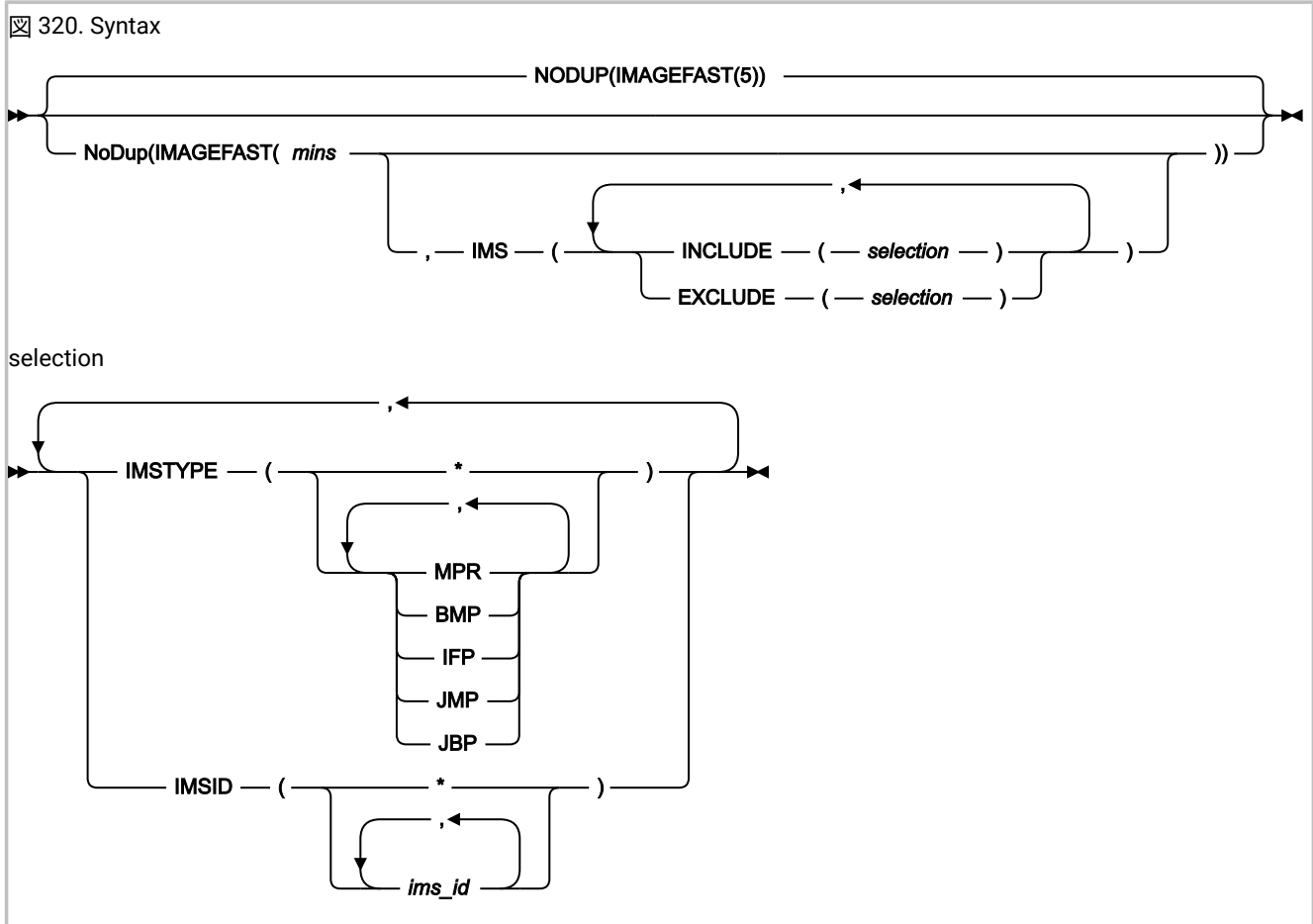
障害の重複を指定するために使用された手法にかかわらず、重複している障害の障害項目の重複カウントが、適宜加算されます。

このオプションは、リアルタイム分析のみに適用されます。

NoDup(NORMAL(*hours*, ...)) が有効であり、*hours* がゼロ以外の値である場合、このオプションは、有効なオプションを示す障害分析レポートのセクションに含まれます。さらに、重複障害が判別されるかどうか、または識別された重複の障害 ID についての情報も、提供されます。関連する重複基準が一致した場合にはレポートが作成されないの  
で、NoDup(CICSFAST) または NoDup(ImageFast) オプション情報は組み込まれません。

以下では、各重複サブタイプの構文を別々に表示していますが、必要に応じて、単一 NoDup オプションと一緒に指定することもできます。

## NoDup(ImageFast(...))



このサブオプションは、最後の障害の発生以降の経過時間を分数で指定する場合に使用します。この時間内に同一 *MVS™* イメージで発生した後続の障害が、該当する障害特性基準を満たしている場合、それらの障害に対する以後の Fault Analyzer 呼び出しは、最後の障害の重複と見なされます。このタイプの障害抑止は、「*IMS™*高速重複障害抑止」と呼ばれます。

 注:

1. *IMS™* 高速重複障害抑止を有効にするには、IMAGEFAST PARM フィールド・オプションと UPDINDEX オプションを使用して Fault Analyzer IDIS サブシステムを始動しておく必要があります。異常終了 TCB で TCBFX フラグがオンになっている場合、NoDup(ImageFast) 処理は実行されません。
2. IDIS サブシステムによって管理される PDSE ヒストリー・ファイルのみを、*IMS™* 高速重複障害抑止の対象に含めることができます。
3. NoDup(ImageFast) オプションは、Fault Analyzer IDIS サブシステムのみが使用します。NoDup(ImageFast) オプションに対する変更は、Fault Analyzer IDIS サブシステムを停止して再始動した後にのみ有効になります。



詳しくは、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照してください。

*mins* に指定できる値 (分数) の有効な範囲は 0 から 10080 です (10080 は 1 週間に相当します)。0 分を指定すると、「IMAGEFAST」重複障害は検出されないことを意味します。

デフォルトの経過時間は 5 分です。

IMS™ 環境の場合、障害が別の障害の重複であると見なされるのは、指定された経過時間 (*mins*) 内に障害が発生し、次の障害の詳細情報が同一である場合です。

- IMS™ program name
- IMS™ サブシステム ID
- 最新の IMS™ 状況コード
- 最新の DB2® SQLCODE
- 障害が発生したプログラム名
- 障害が発生したプログラムのコンパイル日付
- 障害が発生したプログラム内のエラーへのオフセット
- 障害が発生したプログラムの長さ
- 異常終了コード
- IDISNAP 呼び出しで指定されたユーザー・タイトル
- 呼び出しチェーン

NoDup(ImageFast) シグニチャーは、各 MVS™ イメージ上の IDIS サブシステムに保持されます。NoDup(ImageFast) オプションに対する変更は、Fault Analyzer IDIS サブシステムを停止して再始動した後にのみ有効になります。詳しくは、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照してください。

デフォルトでは、すべての IMS™ ジョブが IMS™ 高速重複障害抑止に適格です。これは、次のオプションを指定した場合と同等です。

```
NoDup(ImageFast(minutes,IMS(INCLUDE(IMSTYPE(*),IMSID(*))))))
```

NoDup(ImageFast(*minutes*,IMS(...))) オプションの INCLUDE サブオプションまたは EXCLUDE サブオプションを使用して、IMS™ 高速重複障害抑止に対する適格性を制限します。指定規則は以下のとおりです。

- 指定値が、現行障害に関連した IMS™ 領域タイプを含んでいる場合、IMSTYPE 基準は一致します。

例えば、以下のオプションを指定するとします。

```
NoDup(ImageFast(5,IMS(EXCLUDE(IMSTYPE(MPR,BMP,IFP))))))
```

この場合、現行障害が IMS™ BMP 領域であれば、IMSTYPE 基準は一致するため、現行障害は IMS™ 高速重複障害抑止に不適格と見なされます。

- *ims\_id* に指定された 1 つ以上の値が、現行障害に関連した IMS™ ID と一致する場合、IMSID 基準は一致します。

例えば、以下のオプションを指定するとします。

```
NoDup(ImageFast(5,IMS(EXCLUDE(IMSID(ABC1,ABC2,ABC3))))))
```

この場合、現行障害が IMS™ サブシステム ID ABC2 に関連付けられていれば、IMSID 基準は一致するため、現行障害は IMS™ 高速重複障害抑止に不適格と見なされます。

- INCLUDE サブオプションまたは EXCLUDE サブオプションの各指定が、現行障害に対してテストされます。指定されたサブオプションの基準がすべて一致する場合、それに応じて適格性の状態が変化します。

例えば、以下のオプションを指定するとします。

```
NoDup(ImageFast(5,IMS(EXCLUDE(IMSTYPE(MPR,BMP),IMSID(ABC1,ABC2,ABC3)))))
```

この場合、現行障害が IMS™ BMP 領域で、サブシステム ID が ABC4 であると、IMSTYPE 基準は一致しますが、IMSID 基準は一致しないため、EXCLUDE 基準は不一致になります。

概念上、結果となる一致状況を判断するうえで、IMSTYPE または IMSID の値では論理 OR 演算が行われ、INCLUDE または EXCLUDE サブオプションでは論理 AND 演算が行われると考えることができます。「|」が「論理 OR」演算を表し、「&」が「論理 AND」演算を表す場合、前のオプションの指定は以下のように解釈できます。

```
NoDup(ImageFast(5,IMS(EXCLUDE(IMSTYPE(MPR | BMP) & IMSID(ABC1 | ABC2 | ABC3)))))
```

単一の INCLUDE または EXCLUDE サブオプションに指定する IMSTYPE 基準および/または IMSID 基準は、それぞれ 1 つずつに限定してください。そうでないと、マッチングは成立しません。

- INCLUDE または EXCLUDE サブオプションは任意の個数指定できます。単一の NoDup(ImageFast(*minutes*,IMS(...))) オプションの中に複数の INCLUDE または EXCLUDE サブオプションを指定する方法は、各サブオプションを別個のオプションとして指定する方法と同等です。

例えば、次のように指定することは

```
NoDup(ImageFast(5,IMS(INCLUDE(IMSTYPE(MPR),IMSID(ABC1)))))
```

次のように指定するのと同じ意味を持ちます。

```
NoDup(ImageFast(5,IMS(INCLUDE(IMSTYPE(MPR))))
NoDup(ImageFast(5,IMS(INCLUDE(IMSID(ABC1)))))
```



**注:** 値が異なっている場合、最後に指定された *mins* の値のみが有効です。

- ワイルドカード (\*) は値を完全置換する場合に使用できます。すなわち、アスタリスクの前に文字を付けることはできません。
- INCLUDE または EXCLUDE サブオプションの処理は、[オプション ページ 567](#)で定義されるオプション・ソースの階層に従って、指定された順序で行われます。例えば、IDICNFxx parmlib メンバーは、使用されたどの IDIOPTS ユーザー・オプション・ファイルよりも先に読み取られます。
- 最も一般的な基準を最初に指定し、その後により具体的な基準を指定するようにしてください。

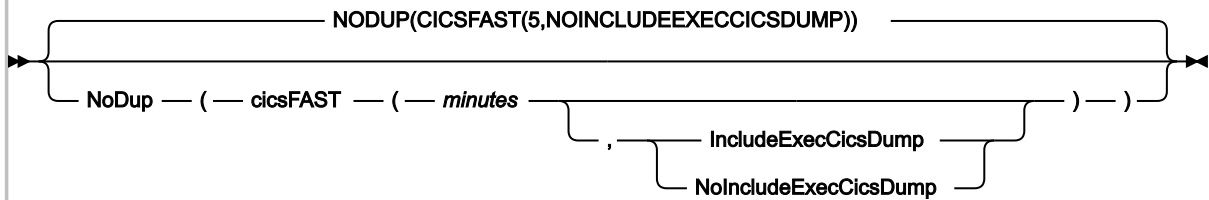
例えば、インストール済み環境で、ABC1 以外のサブシステム ID を使用する IMS™ MPR ジョブのみを対象に IMS™ 高速重複障害抑止を使用したい場合は、以下のオプションを指定します。

```
NoDup(ImageFast(5,IMS(EXCLUDE(IMSTYPE(*) /* Exclude everything */
INCLUDE(IMSTYPE(MPR)) /* Include only IMS MPR regions */
EXCLUDE(IMSID(ABC1)) /* Exclude subsystem ID ABC1 */
)))
```

障害特性に基づいて、ある障害が別の障害の重複であると見なされた場合、NoDup(ImageFast(...)) オプションが有効になっていると、障害分析はスキップされ、オリジナルの障害に関連した重複カウントが 1 増やされ、[IDI0121I ページ 722](#) メッセージが出されます。

## NoDup(CICSFAST(...))

図 321. Syntax



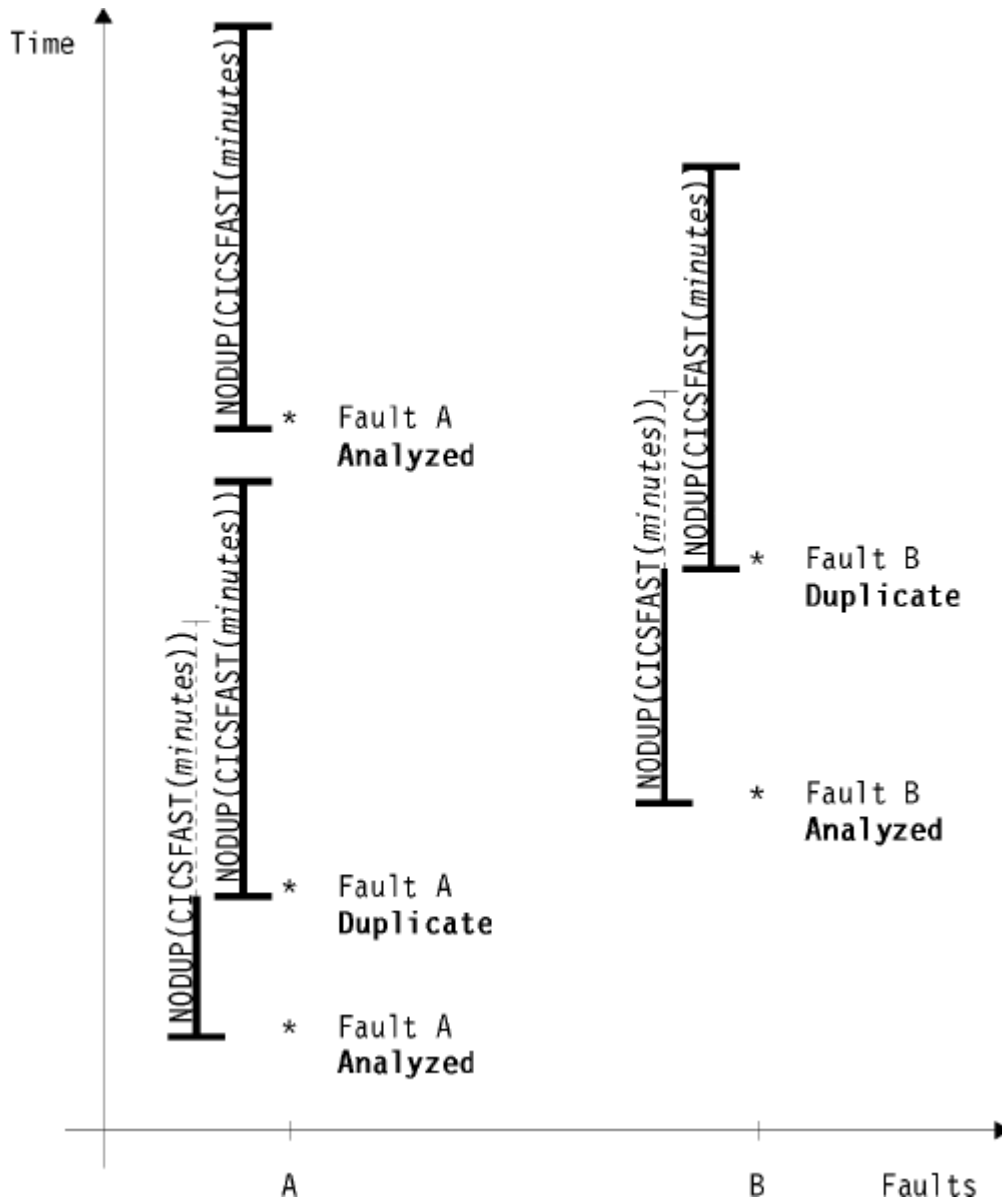
このサブオプションは、最後の障害の発生以降の経過時間を分数で指定する場合に使用します。この時間内に同一ジョブ・ステップで発生した後続の障害が、該当する障害特性基準を満たしている場合、それらの障害に対する以後の Fault Analyzer 呼び出しは、最後の障害の重複と見なされます。

図 322 : NoDup(CICSFAST) の図 ページ 617 には、以下の場合のこのタイプの処理が図示されています。

- “障害 A” のすべてのインスタンスは、障害特性のみに基づいて、互いの重複であると見なされる。
- “障害 B” のすべてのインスタンスは、障害特性のみに基づいて、互いの重複であると見なされる。
- “障害 A” の特性は、“障害 B” の特性と一致しない。



図 322. NoDup(CICSFAST) の図



このオプションは、短期間に発生する複数の同一トランザクション異常終了が、後にシステム・リソースを使い果たす危険性があるため、Fault Analyzer により分析されないようにするために、現在、CICS® で使用されます。

*minutes* の有効な範囲は 0 から 10080 (10080 は 1 週間に相当します) です。0 分を指定すると、「CICSFAST」重複障害は検出されないことを意味します。

デフォルトの経過時間は 5 分です。

CICS® 環境の場合、障害が別の障害の重複であると見なされるのは、指定された経過時間 (*minutes*) 内に障害が発生し、次の障害の詳細情報が同一である場合です。

- トランザクション ID
- CICS® 異常終了コード
- 障害が発生したプログラム名
- 要求 ID
- システムおよびユーザーのセンス・コード
- オペレーティング・システムの異常終了コード
- エラー点のオフセット
- 異常終了へのエントリーでの PSW

NoDup(CICSFAST) シグニチャーは CICS® 領域に保持されます。

トランザクション ID を除いて、すべての障害の詳細情報は トランザクション異常終了制御ブロック (TACB) から入手されます。両方の TACB で有効な TACB フィールドのみが、重複比較に含まれます。

デフォルトで、EXEC CICS® DUMP コマンドを使用した Fault Analyzer の呼び出しは、NoDup(CICSFAST) 重複判定の対象になりません。しかし、IncludeExecCicsDump サブオプションが指定された場合は、EXEC CICS® DUMP コマンドの使用時も NoDup(CICSFAST) 重複判定が実行されます。

- TACB が使用可能な場合、上に示した CICS® 異常終了の場合と同じ基準で重複判定が実行されます (CICS® 異常終了コードの代わりに CICS® ダンプ・コードが使用されます)。
- TACB が使用可能でない場合、重複判定は以下の基準に基づいて実行されます。
  - トランザクション ID
  - CICS® ダンプ・コード
  - 障害が発生したプログラム名

障害特性に基づいて、ある障害が別の障害の重複であると見なされたとき、NoDup(CICSFAST(...)) オプションが有効であると、履歴・ファイル項目の書き込みが抑止され、最後の障害に関連した重複カウントが 1 つ増え、メッセージ [ID10066I ページ 710](#) が出されます。このタイプの障害抑止は 「*CICS fast duplicate fault suppression*」 と呼ばれます。

Fault Analyzer は出口 IDINDFUE を提供します。これは異常終了コードおよび他の属性に基づいて、特定の障害の重複指定の指定変更で使用できます。詳しくは、[CICS NoDup\(CICSFAST\) オーバーライド・アセンブラー出口 \(IDINDFUE\) ページ 414](#)を参照してください。

Fault Analyzer が、新たに発生した異常終了を、既に削除された障害項目の重複として指定し続けるのを防止するには、NoDup(CICSFAST) 記録域を消去する必要があります。詳しくは、[NoDup\(CICSFAST\(...\)\) 記録域の消去 ページ 407](#)を参照してください。

## NoDup(CICSFAST) オプションの変更

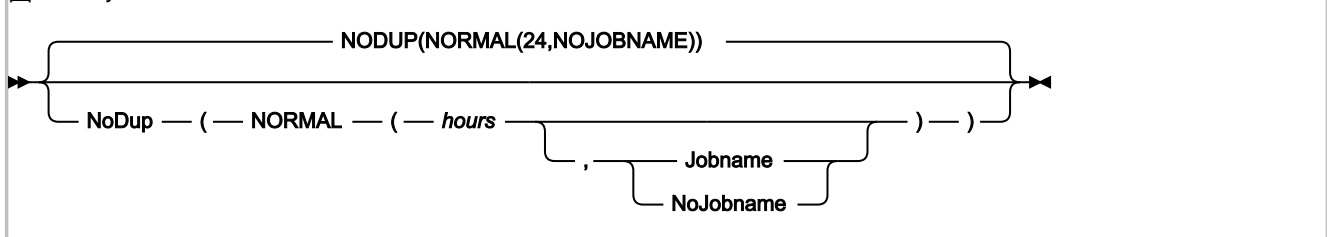
通常は、IDIOPTS DD ステートメントを CICS® プロシージャー内で使用して、オプションへの変更を容易にできるようにします。この際、CICS® を一度停止してから再始動する必要はありません。しかし、NoDup(CICSFAST) オプションに関する限り、制限が存在します。IDIOPTS データ・セットの実際の処理は、Fault Analyzer のメイン・モジュール IDIDA により実行されます。このモジュールは、MVS™ 生成サブタスクで実行されるメインプログラムです。NoDup(CICSFAST) 処理は、このサブタスクが生成される前に、IDIXCX53 出口コードで行われます。NoDup(CICSFAST) に使用される値

は、その CICS® 領域で直前にフル稼働 (Fault Analyzer タスク生成) した IDIDA で設定された値です。CICS fast duplicate fault suppression が行われる場合 (メッセージ [IDI0066I ページ 710](#))、IDIDA のタスク生成、およびオプション・データ・セットの読み取りは行われません。別の IDIDA タスク生成が発生しないと、NoDup(CICSFAST) オプションの変更は、NoDup(CICSFAST) アクションの出口まで反映されません。これは、別の固有の障害 (EXEC CICS® ABEND を実行するために CECI で新しい異常終了を作成することによって強制されます)、または NoDup(CICSFAST) タイムアウトの発生を待機することを意味します。

あるいは、CFA トランザクションを使用してすべての CICS 呼び出し出口をアンインストールおよび再インストールすることで、このオプションに対する変更をすぐに有効化することもできます。[CICS トランザクション異常終了分析の制御 ページ 405](#)を参照してください。

## NoDup(NORMAL(...))

図 323. Syntax



このサブオプションは、経過時間を時間数で指定する場合に使用されます。この時間内では、同じ履歴・ファイルを使用した Fault Analyzer の呼び出しは、以前に記録された障害の重複であると見なされます。このメカニズムは、例えば異常終了バッチ・ジョブに使用される標準の重複障害検出メカニズムです。

*hours* の有効な範囲は 0 から 168 です (168 は 1 週間に相当します)。0 分を指定すると、「NORMAL」重複障害は検出されないことを意味します。

デフォルトの経過時間は 24 時間です。

障害が、既に同じ履歴・ファイル内に記録されている別の障害の重複であると見なされるのは、指定された経過時間 (*hours*) 内に障害が発生し、障害点が判別されていて、以下の障害の詳細情報が同一である場合です。

- 初期異常終了コード
- CICS® トランザクション ID (CICS® トランザクション障害にのみ適用可能)
- 障害点モジュール名
- 障害点 CSECT (またはプログラム) 名
- CSECT (またはプログラム) の開始からの障害点オフセット
- 障害点モジュールのリンク・エディット日時 (重複障害の履歴・ファイル項目内だけでなく、現行の障害に使用可能な場合)
- ユーザー・タイトル
- ジョブ名 (Jobname サブオプションが指定されている場合のみ。それ以外の場合、ジョブ名は重複障害の判別には組み込まれません。)

NoDup(NORMAL) シグニチャーは、履歴・ファイル内の既存の障害項目に基づきます。

障害特性および有効な NoDup(NORMAL(*hours*)) オプションに基づいて、障害が別の障害の重複であると見なされる場合、デフォルトでは、履歴・ファイル項目と重複障害のダンプの両方の書き込みを抑制します。ただし、抑制に関する最終的な決定は、End Processingユーザー出口によりオーバーライドされます。

障害が他の障害の重複であると判断された場合は、既存の障害に関連する重複カウントに 1 が加算され、メッセージ [ID10044I ページ 705](#) が発行されます。

## トレース情報

IDITRACE DDname をジョブに追加すると、Fault Analyzer はトレース情報を提供できます。この情報は、リアルタイム分析時に、特定の障害が別の障害の「通常の」重複であると見なされるかどうかの理由を理解するのに役立ちます。例:

```
//IDITRACE DD SYSOUT=*
```

(CICS® 環境でこのトレースを活動化する別の方法については、[IDITRACE CICS](#) において [ページ 407](#) を参照してください。)



**注:** NoDup(NORMAL) トレース情報のみが使用可能です。NoDup(CICSFAST) はありません。

トレース情報は、IDITRACE DDname 宛先に書き込まれます。NoDup オプション処理トレースの例は、次のとおりです。

### 図 324. NoDup オプション処理トレースの例

```
NoDup(NORMAL(24)) option processing match values:
Abend Code . . . . . : S0C7
CICS Transaction ID. . . . . : n/a
Module Name. . . . . : IDISCBL1
CSECT Name . . . . . : IDISCBL1
Offset . . . . . : X'3D4'
Module Link-Edit Date. . . . : 2002/05/06
Module Link-Edit Time. . . . : 15:56:35
Fault ID F00615 module link-edit time 15:55:09 did not match
No duplicate fault exists
```

不一致 (上記の例では障害 ID F00615) の理由に関する情報は、NoDup オプションに対して有効な時間間隔内の障害項目にのみ提供され、また次の場合にも提供されます。

- リンク・エディットの日付が一致しない
- リンク・エディットの時刻が一致しない
- オフセットが一致しない

## PDTCCopts

### 図 325. Syntax

```
▶▶ PDTCCopts — ( — CONFIGDSN — ( — data-set-name — ) — )▶▶
```

PDTCCopts オプションは、Fault Analyzer Web ブラウザー・インターフェースで使用される Fault Analyzer 設定を IBM Application Delivery Foundation for z/OS Common Components に対して指定するために使用されます (詳しくは、Fault Analyzer Web インターフェースの使用を参照してください)。

**CONFIGDSN(*data-set-name*)**

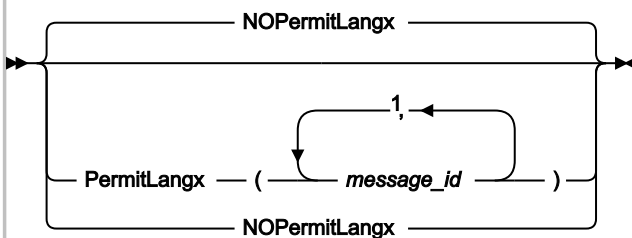
これは、ISPF 構成設定を含むデータ・セットの名前を指定します (また、オプションで、メンバー名を括弧で囲んで指定します)。Fault Analyzer Web ブラウザー・インターフェースに必要となるのは最後の 2 つのオプション ISPF\_PROF\_DSN および ISPF\_APPL のみですが、このファイルは、Fault Analyzer plug-in for Eclipse にも使用される構成ファイルと同じものにすることができます。詳しくは、[IBM Application Delivery Foundation for z/OS Common Components サーバーのカスタマイズ ページ 563](#)を参照してください。

指定の例:

```
CONFIGDSN(USER.PARMLIB(IDISRV03))
```

**PermitLangx**

図 326. Syntax



注:

<sup>1</sup> コンマまたは空白文字のいずれかを区切り文字に使用できます。

一部のコンパイラーでは、4 より大きい戻りコードとなるコンパイルでメッセージを出しますが、引き続き、有効なオブジェクト・モジュールを生成します。しかし、デフォルトの IDILANGX 処理では、4 より大きい戻りコードとなるすべてのメッセージをエラーとして扱っているため、このオプションを使用して、4 より大きい戻りコードで終了するコンパイルのリストで検出されたメッセージ ID のうち、無視すべきメッセージ ID のリストを指定することができます。

PermitLangx オプションは、COBOL または PL/I コンパイラー・リストにのみ適用できます。

PL/I メッセージ ID は、8 文字の長さで指定する必要があります。例えば、以下のメッセージがコンパイラー・リストに書き込まれたとします。

Message	Statement	Message Description
IBM1352I E	11	The statement element PUT is invalid. The statement will be ignored.

この場合、次のように PermitLangx オプションを指定することで、IDILANGX 処理を実行できます。

```
PermitLangx(IBM1352I)
```

OS/VS COBOL メッセージ ID は、8 文字の長さで指定する必要があります。例えば、以下のメッセージがコンパイラー・リストに書き込まれたとします。

CARD	ERROR MESSAGE
20	IKF1150I-C ILLEGAL CHARACTER IN COLUMN 7, BLANK ASSUMED.

この場合、次のように PermitLangx オプションを指定することで、IDILANGX 処理を実行できます。

```
PermitLangx(IKF1150I)
```

他の COBOL メッセージ ID (つまり OS/VS COBOL 以外) はすべて、9 文字の長さで指定する必要があります。例えば、以下のメッセージがコンパイラ・リストに書き込まれたとします。

LineID	Message code	Message text
1	IGYDS0017-E	"IDENTIFICATION" should begin in area "A". It was processed as if found in area "A".

この場合、次のように PermitLangx オプションを指定することで、IDILANGX 処理を実行できます。

```
PermitLangx(IGYDS0017)
```

## ワイルドカードの使用

指定の重大度のメッセージをすべて許可するよう指定する場合、以下の表記を使用できます。

```
XXX-C
```

それぞれの意味は以下のとおりです。

### XXX

メッセージ接頭語。

### C

メッセージの重大度レベル。

以下の例は、Enterprise PL/I (IBM®) のすべてのエラー・レベル・メッセージ (E) を許可する方法を示します。

```
PermitLangx(IBM-E)
```

指定されたオプションが適用されるメッセージ重大度は、コンパイラ・リストでメッセージ ID の後に表示される項目です。例えば、Enterprise PL/I メッセージ IBM1352I であれば、コンパイラ・リストは以下のように表示されます。

Message	Statement	Message Description
IBM1352I E	11	The statement element SKIP is invalid. The statement will be ignored.

この例のメッセージ重大度レベルは E であり、通常これは戻りコード 8 を意味します。

指定されたメッセージ重大度レベルのメッセージのみが影響を受けます。

以下の例は、Enterprise PL/I と Enterprise COBOL の両方のエラー・レベル・メッセージを無視するように指定する方法を示しています。

```
PermitLangx(IBM-E,IGY-E)
```

## 複数のオプション指定を結合

最後に指定した PermitLangx オプションのみが使用されます。つまり、複数指定しても累積されません。例えば、IDICNFxx parmlib メンバーにおいて以下のオプションが指定されているとします。

```
PermitLangx(IBM-E)
PermitLangx(IGYDS0017,IBM1352I)
PermitLangx(IGY-E)
```

この場合は、オプション

```
PermitLangx(IGY-E)
```

のみが有効になります。

指定されている 3 つのオプションをすべて保持するには、すべてのサブオプションを 1 つのオプション指定に結合します。以下に例を示します。

```
PermitLangx(IBM-E,
            IGYDS0017,IBM1352I,
            IGY-E)
```

## サブオプションの最大長

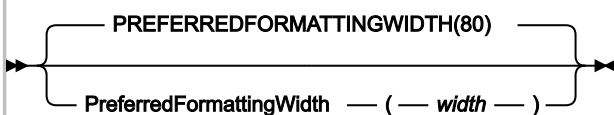
すべてのメッセージ ID の長さの合計にメッセージ ID の数を加算し、さらに 1 だけ減算した、「正規化された長さ」が 75 を超えてはなりません。この長さは、可能な限り最も短縮された (サブオプションを分離する単一の区切り文字のみを含み、不要な空白埋め込みが使用されていない) 指定に相当します。例えば、次の正規化された長さは、

```
PermitLangx(IKF1150I IBM-E,IGY-E )
```

20 です。この値は次の方法で計算されます。メッセージ ID 内の合計文字数は  $8 + 5 + 5 = 18$  です。3 つのメッセージ ID が指定されたため、正規化された長さの合計は  $18 + 3 - 1 = 20$  になります。

## PreferredFormattingWidth

図 327. Syntax



このオプションは、以下のパラグラフ・テキストおよび 16 進数ストレージ・フォーマットで使用される、リアルタイムまたはバッチ再分析レポートの行幅を指定します。レポートの他の部分は、このオプションに影響されません。

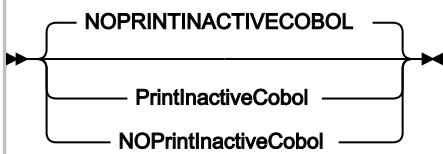
16 進数ストレージ・フォーマットは、1 行 16 バイトのデフォルトから 1 行 32 バイトに変更するため、128 文字の最小フォーマット幅を必要とします。

*width* の有効な範囲は 80 から 132 です。

このオプションは、対話式再分析には適用されません。

## PrintInactiveCOBOL

図 328. Syntax



PrintInactiveCOBOL オプションをリアルタイム再分析またはバッチ再分析で使用すると、非アクティブ COBOL プログラム (現在の保管域チェーンにないプログラム) のストレージを分析レポートに組み込むように要求できます。



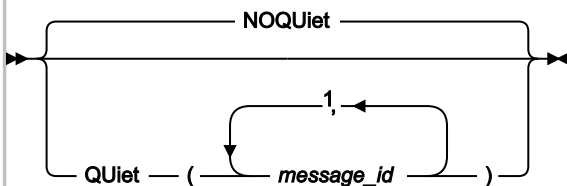
**注:** 非アクティブ COBOL プログラムの数とサイズによっては、このオプションによって、レポートのサイズが著しく大きくなる場合があります。

NoPrintInactiveCOBOL オプションが有効なときは (デフォルト)、非アクティブ COBOL プログラムのストレージは組み込まれません。

このオプションは、非アクティブ COBOL プログラムのストレージを常に表示可能な対話式再分析には適用されません。

## Quiet

図 329. Syntax



注:

<sup>1</sup> コンマまたは空白文字のいずれかを区切り文字に使用できます。

特定のオペレーター宛メッセージを抑制するには、これらのメッセージを Quiet オプションのサブオプションとして指定できます。メッセージ ID は、完全な 8 文字のメッセージ ID (例えば、[IDI0003I ページ 696](#)) である必要があります。

NoQuiet が指定される場合、すべてのオペレーター宛メッセージが発行されます。

最後に指定された Quiet または NoQuiet オプションが有効になります。例えば、IDICNFxx parmlib メンバーが Quiet を指定している場合でも、これは、ユーザー・オプション・ファイルの NoQuiet 指定によりオーバーライドされます。Quiet オプションを使用するメッセージ・リストの各指定により、前の指定は完全にオーバーライドされます。つまり、メッセージ番号は累積されません。

このオプションを使用することで、SYSLOG に書き込まれるメッセージの数は削減できますが、一般的には、このオプションの使用はお勧めしません。理由は、メッセージの欠落により、Fault Analyzer 処理の問題診断が難しくなることがあるた



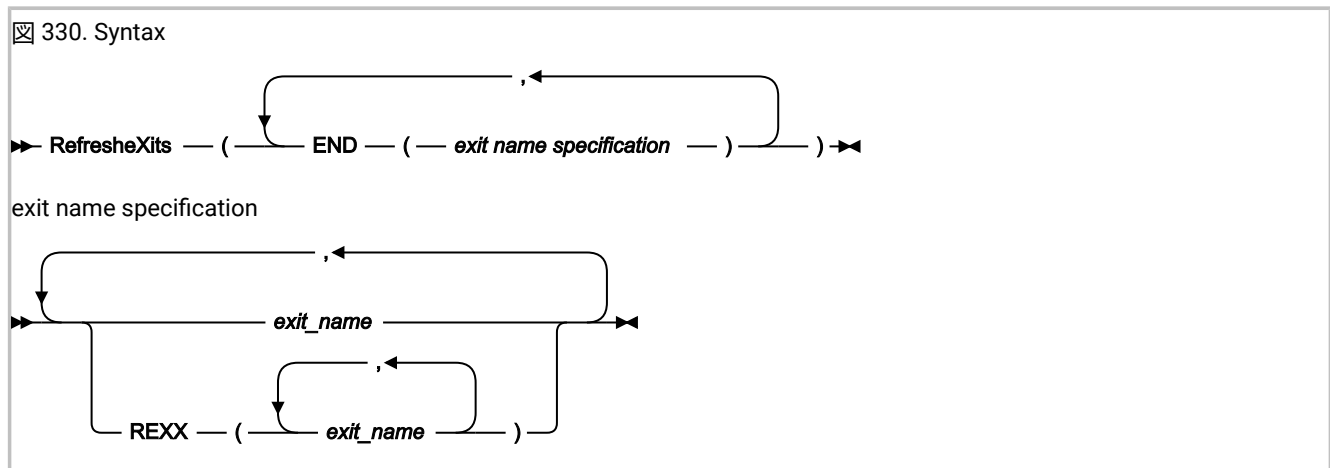
めです。また、報告された問題の資料を収集する際に、IBM® サポートが要求する特定の診断設定によって、このオプションがオーバーライドされることもあります。

メッセージ [IDI0001I ページ 695](#) などの一部のメッセージは、Quiet オプションを使用して抑止することはできません。

S レベル (重大) メッセージまたは Fault Analyzer IDIS サブシステムによって書き込まれるメッセージは抑止できません。

メッセージ番号 [IDI0034I ページ 703](#)、[IDI0066I ページ 710](#)、[IDI0118W ページ 721](#) に関連したこのオプションに対する変更を有効にするには、CFA トランザクションを使用してすべての CICS® 呼び出し出口をアンインストールおよび再インストールしてください。[CICS トランザクション異常終了分析の制御 ページ 405](#) を参照してください。

## RefreshExits



RefreshExits オプションは、ダンプ登録障害項目の最初のバッチ再分析中に呼び出されるユーザー出口のタイプおよび名前を指定します。指定した出口タイプについて、複数の出口名を指定でき、すべての出口の呼び出しが試行されます。

RefreshExits オプションの複数指定は累積されます。

出口は、REXX EXEC かロード・モジュールのどちらかです。

- REXX EXEC は次のように指定する必要があります。

```
REXX(exit_name_1, exit_name_2, ...)
```

また、IDIEEXEC DDname を介して使用可能である必要があります。

- ロード・モジュールは、標準システム検索パス (LPA、LINKLIST、または JOBLIB/STEPLIB JCL ステートメント) から使用可能である必要があります。

考えられる出口タイプは、以下のとおりです。

### END

End Processing ・ユーザー出口。このユーザー出口を使用して、Fault Analyzer ミニダンプの抑止、またはヒストリー・ファイル項目全体の更新を要求できます。詳しくは、[終了処理ユーザー出口 \(障害項目リフレッシュ\) ページ 498](#) を参照してください。

`exit_name` として指定する出口名は、有効な TSO/E REXX EXEC またはロード・モジュールの名前にすることができます。ただし、一部の名前は、特別な目的のために予約されています。

**NONE**

特殊名「NONE」は「ヌル」出口を表します。この出口は呼び出されません。結果、指定されたタイプの出口を呼び出すそれ以降の試みは終了します。

**-DROPCNF-**

特殊名「-DROPCNF-」を使用して、IDICNFxx parmlib メンバーから出口指定を廃棄します。詳しくは、[IDICNFxx parmlib メンバーのユーザー出口指定の廃棄](#) ページ 598を参照してください。

RefreshExits オプションによって1つ以上の出口が指定されている場合は、出口に関する情報が分析レポートの「有効になっているオプション」セクションに書き込まれます。このセクションでは、指定されたすべての出口がリストされ、それぞれのタイプの中で呼び出された出口がある場合、それらの出口が示されます。

分析レポートの「有効になっているオプション」セクションに書き込まれる情報の例は、以下のとおりです。

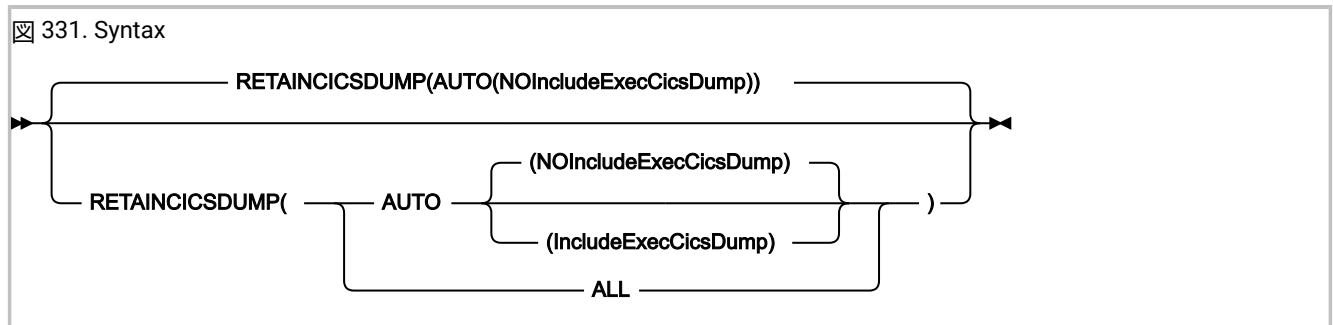
```
Exits:

The following user exits were specified via RefreshExits options.

Type      Name      Type Invoked
-----
END       ABC1      LMOD Yes
```

この例では、1つのEnd Processingユーザー出口 (ABC1) がロード・モジュールとして指定されました。このユーザー出口は呼び出されました。

**RetainCICSDump**



RetainCICSDump オプションは、分析の成功に基づくか (AUTO) または無条件に (ALL) に CICS® トランザクション・ダンプを保存することを指定します。

分析の成功 (AUTO) 設定とは、Fault Analyzer 分析が正常に完了した場合にトランザクション・ダンプが抑止されることを意味します。ただし、分析に重大な問題がある場合にはトランザクション・ダンプは保持されます。AUTO(NOIncludeExecCicsDump) は、分析が成功した場合でも、EXEC CICS® DUMP 呼び出しに対して発行された CICS® トランザクション・ダンプの抑止を防ぐために使用します。これは、デフォルトで

す。AUTO(IncludeExecCicsDump) は、分析が成功した場合に、EXEC CICS® DUMP 呼び出しを含め、すべての CICS® トランザクション・ダンプを抑止するために使用します。

このオプションは CICS® トランザクション障害のリアルタイム分析にのみ適用され、Fault Analyzer が XPCABND または XDUREQ グローバル・ユーザー出口のいずれかにインストールされる必要があります。このオプションに対する変更を有効にするには、CFA トランザクションを使用してすべての CICS 呼び出し出口をアンインストールおよび再インストールします。CICS トランザクション異常終了分析の制御 ページ 405 を参照してください。



**注:** EXEC CICS® DUMP コマンドの結果としてのトランザクション・ダンプは、デフォルトでは、Fault Analyzer によって抑止されません。この抑止の欠如は、発行側アプリケーション・プログラムに SUPPRESSED 応答が返されることによるものです。正しく処理されないと、AEXW 異常終了が発生するおそれがあります。

このオプションは、障害項目をヒストリー・ファイルに書き込む処理に影響を与えません。

このオプションは、CICS® トランザクション障害のリアルタイム実行にのみ有効なオプションを示す、障害分析レポートのセクションに含まれます。

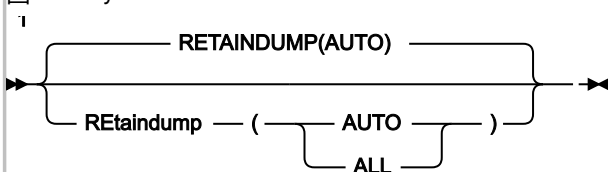


**注:**

1. End Processingユーザー出口を使用すると、有効な RetainCICSDump オプションが事実上指定変更されることがあります。
2. MVS™ ダンプ・データ・セットの保存を制御するには、代わりに RetainDump オプションを使用します。
3. ダンプ抑止の詳細については、[ダンプの抑止 ページ 33](#)を参照してください。

## RetainDump

図 332. Syntax



**注:**

- <sup>1</sup> コンマまたはブランク文字のいずれかを区切り文字に使用できます。

RetainDump オプションは、Fault Analyzer の IEAVTABX 変更オプション/抑止ダンプ出口 (IDIXDCAP) が、分析の成功に基づくか (AUTO) または無条件に (ALL)、異常終了ジョブ・ステップの SYSABEND ダンプ、SYSMDUMP、または SYSUDUMP を保存することを指定します。

分析の成功 (AUTO) 設定とは、Fault Analyzer 分析が正常に完了した場合には MVS™ ダンプが抑止されることを意味します。ただし、分析に重大な問題がある場合には MVS™ ダンプは保持されます。

このオプションは、非 CICS トランザクション障害のリアルタイム分析にのみ適用され、IEAVTABX 変更オプション/抑止ダンプ出口 IDIXDCAP にのみ適用できます。このオプションは、障害項目を履歴・ファイルに書き込む処理に影響を与えません。

IEAVTABX\_EXIT 出口リストが IDIXDCAP の前にインストールされており、その出口が(その戻りコードにより)ダンプ・オプションの変更またはダンプの抑止を要求した場合、Fault Analyzer は、オプション設定にかかわらず、その要求を受け入れます。

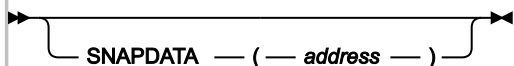
このオプションは、非 CICS トランザクション障害のリアルタイム分析にのみ有効なオプションを示す、障害分析レポートのセクションに含まれます。



1. 分析制御またはEnd Processingユーザー出口を使用すると、有効な RetainDump オプションを事実上オーバーライドします。
2. CICS® トランザクション・ダンプの保存を制御するには、代わりに RetainCICSDump オプションを使用します。
3. ダンプ抑止の詳細については、[ダンプの抑止 ページ 33](#)を参照してください。

## Snapdata

図 333. Syntax



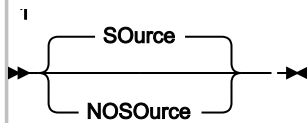
Snapdata オプションを使用して、ENV データ域の最終バージョンをユーザー指定データ域に保存する、プログラム SNAP インターフェイス (IDISNAP) のストレージのアドレスを指定します。*address* は、8 バイト文字形式でストレージ・アドレスを指定します。

このオプションを、IDICNFxx PARMLIB 構成メンバーに指定するか、または IDIOPTS DD ステートメントに指定することはできません。

IDISNAP に対する呼び出しでこのオプションを指定する方法については、[SNAPDATA オプションの使用 ページ 42](#)を参照してください。ENV データ域について詳しくは、[ENV - 共通出口環境情報 ページ 650](#)を参照してください。

## Source

図 334. Syntax



注:

<sup>1</sup> コンマまたは空白文字のいずれかを区切り文字に使用できます。

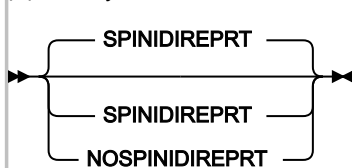
Source オプションは、リアルタイムのソース・コード分析が実行されるかどうかを制御するのに使用されます。

- Source が有効な場合、リアルタイム・ソース・コード分析が通常どおりに実行されます。この値はデフォルトです。
- NoSource が有効である場合、Fault Analyzer は、コンパイラ・リストにもサイド・ファイルにもアクセスせず、ソース・コード情報がリアルタイム・レポートで使用不可である可能性があります。ミニダンプまたは関連付けられた MVS ダンプ・データ・セットデータ・セットのどちらかが書き込まれていることを前提とすると、ヒストリー・ファイル項目に対して再分析を実行すると、ソース・コード情報を提供することが、引き続き可能です。

多数のコンパイラ・リストまたはサイド・ファイルのデータ・セットが使用されるインストール済み環境、および追加のソース・コード情報を提供するための再分析を実行することが受け入れ可能な代替方法であるインストール済み環境では、このオプションが、リアルタイム分析のパフォーマンスを改善する手段として役立つ場合があります。

## SpinIDIREPRT

図 335. Syntax



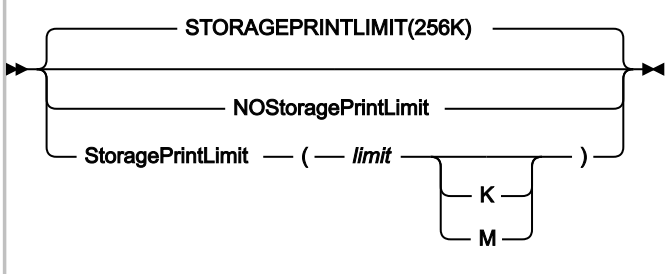
NoSpinIDIREPRT オプションをリアルタイム分析で使用すると、IDIREPRT DDname の動的割り振りを、処理の終了時、明示的に割り振り解除しないよう要求できます。このオプションは、例えば NJE を使用して、当該リアルタイム分析用のすべての JES スプール・データを別のシステムに単一ファイルとして送信できるようにするために利用できます。

SpinIDIREPRT オプションが有効なときは (デフォルト)、IDIREPRT DDname の動的割り振りは、処理の終了時、明示的に割り振り解除されます。

このオプションは、バッチまたは対話式再分析には適用されません。

## StoragePrintLimit

図 336. Syntax



このオプションは、作業用ストレージで大規模な配列を使用するプログラムなど、大容量の仮想記憶域がレポートに表示可能である場合、レポートのサイズを制限するのに使用できます。

このオプションはレポートに対して、イベントに関する関連ストレージ域で使用するフォーマットのスタイルを、以下のよう決定します。

表 16. StoragePrintLimit オプションの動作

コンパイラ・リストまたは サイド・ファイルが使用可能 <sup>1</sup>	StoragePrintLimit を超えていない	StoragePrintLimit を超えている
あり	イベントに関するソース・ビューの完全なフォーマット	イベント関連のソース・ビューのフォーマット
なし	イベントに関する 16 進ダンプの完全なフォーマット	イベントに関する 16 進ダンプのアドレス可能 <sup>2</sup> なフォーマット

指定された制限を超えているかどうかを判別するために、重複するストレージ範囲を考慮に入れながら、レポート対象である 16 進ダンプされたイベント関連のストレージ域と CICS® トランザクション・ストレージ域のすべての合計に対してテストされます。SystemWidePreferred(StorageAreas(Hex)) オプションは、その実際の設定にかかわらず、ストレージ範囲の累積に対して有効であると想定されます。

このオプションは、リアルタイムまたはバッチ再分析レポートにのみ影響します。対話式再分析レポートには影響しません。リアルタイムで使用される場合でも、ミニダンプの再分析から全データを入手できます。これは、StoragePrintLimit オプションによって、サイズが削減されることはありません。

*limit* の有効な指定は以下になります。

- バイト: 0 から 2147483648。
- キロバイト数: 0K-2097152K
- メガバイト数: 0M-2048M

NoStoragePrintLimit は、StoragePrintLimit(2147483648) を指定した場合に相当します。

1. これには、テスト・オプションの使用によりロード・モジュールから取得したソースが含まれます。
2. 汎用レジスター値の 4 キロバイト以内のアドレス可能ストレージ。

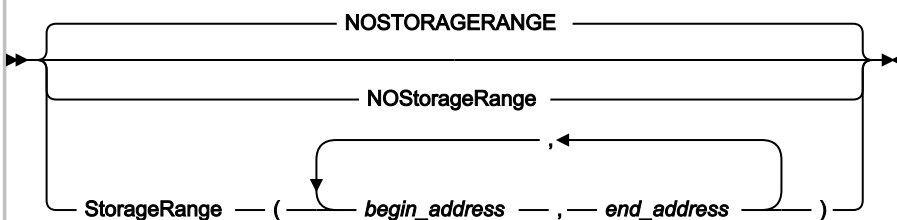
ストレージの有効な印刷制限を超えたかどうかを判断するのに使用する累積ストレージ量に関する情報は、IDITRACE に書き込まれます。

レポートの「Options in Effect」セクションで、このオプションは実際の指定にかかわらず、可能な限り最大の装置タイプを使用して表示されます。例えば、StoragePrintLimit(1024K) が指定された場合、オプションは StoragePrintLimit(1M) と表示されます。また、StoragePrintLimit(1025) が指定された場合、1025 が MB または KB の偶数ではないため、オプションは StoragePrintLimit(1025) と表示されます。

ストレージの印刷制限を超えたかどうかの情報は、レポートの「Options in Effect」セクションに含まれます。制限を超えた場合、制限を超えた部分のストレージ量も表示されます。

## StorageRange

図 337. Syntax



このオプションは、デフォルトのイベント関連またはシステム全体の 16 進ダンプ・ストレージの代わりに、特定の 1 つ以上のストレージ域を分析レポートに表示するように要求するために使用できます。

このオプションは基本的に、アプリケーション・プログラムからの IDISNAP 呼び出しとともに使用します。この場合、特定のストレージ域のみが関係します。例えば、COBOL プログラマーは、プログラムの作業用ストレージ・セクションに特定の変数を指定する StorageRange オプションを指定して、IDISNAP を呼び出すように選択できます。プログラムのすべての関連ストレージが含まれた分析レポートではなく、要求した領域のみが含まれた分析レポートになります。

IDISNAP に対する呼び出しでこのオプションを指定する方法については、[プログラム SNAP インターフェース \(IDISNAP\) の使用 ページ 39](#)を参照してください。

リアルタイム分析でこのオプションが有効になっていると、それ以降の障害項目の再分析時にも同じ指定が有効になります。このとき、オプションを指定変更することはできません。

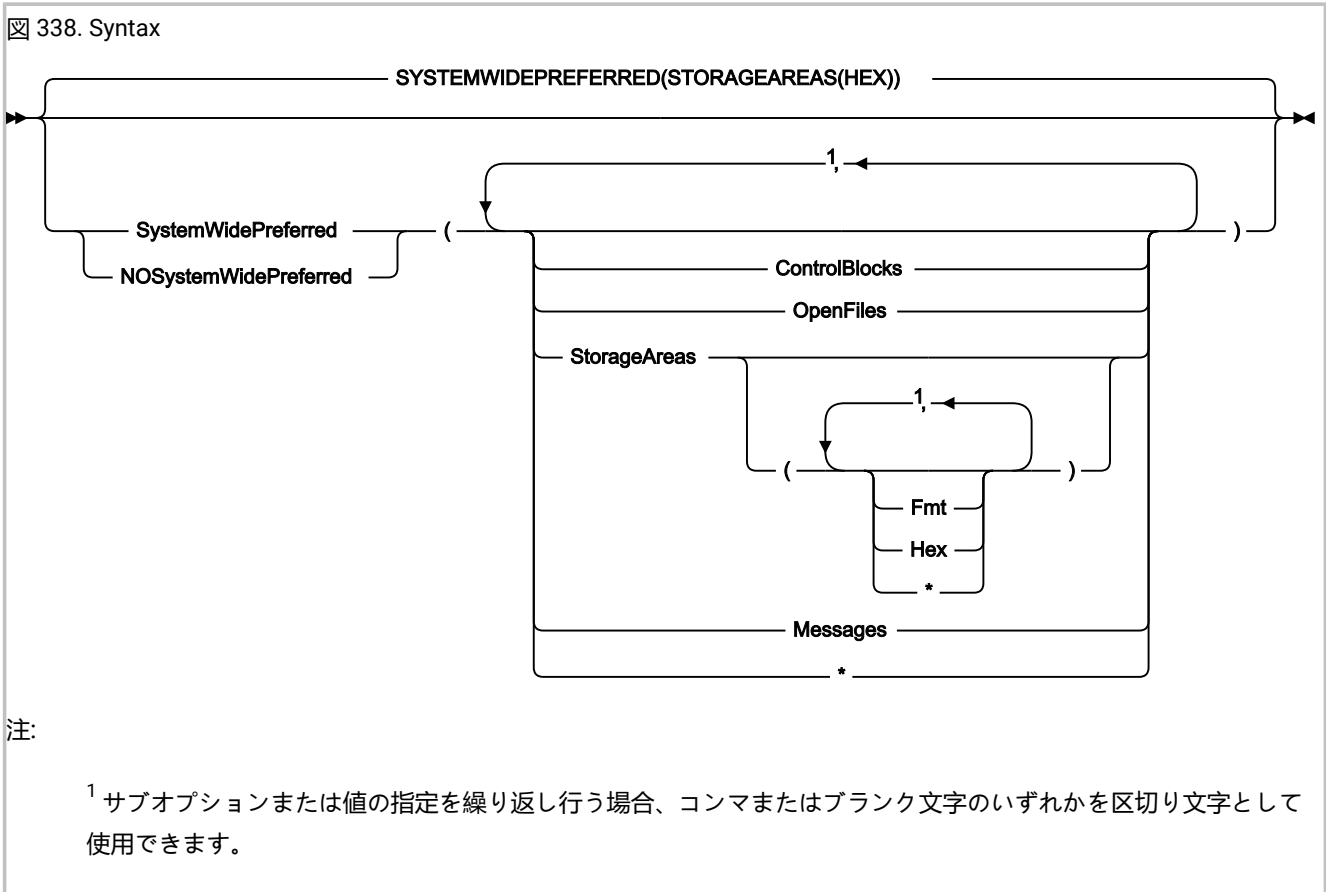
開始アドレスは、ストレージ域の最初のバイトのアドレスで、終了アドレスは、ストレージ域の最後のバイトの直後のバイトのアドレスです。つまり、終了アドレスから開始アドレスを減算すると、組み込まれるストレージ域の長さになります。

終了アドレスが開始アドレスより大きいという条件を満たしていれば、任意の順序で複数のストレージ範囲を指定できます。無効なストレージ範囲は無視されます。

StorageRange オプションを複数指定すると、それより前の指定はすべて置き換えられます。つまり、このオプションは累積型ではありません。

StorageRange オプションは、1 つ以上の有効なストレージ域についてリアルタイムで指定された場合にのみ、分析レポートの「有効なオプション (Options in Effect)」セクションに表示されます。

## SystemWidePreferred



このオプションは、リアルタイムまたはバッチ再分析レポートにのみ、適用できます。対話式再分析レポートには影響しません。

Fault Analyzer は、デフォルトで、イベントに関連すると考えられる情報を、そのイベントの「Event Details」セクションに配置します。ただし、SystemWidePreferred オプションを使用することで、Fault Analyzer はこのような情報をレポートの「System-Wide Information」セクションに配置するようになります。

以下のサブオプションを使用して、レポート内のイベントに関係するさまざまな領域の配置を変更できます。

### ControlBlocks

通常、レポートの「Event Details」セクションの見出し「Associated Control Blocks」の下に配置される、イベントに関係するすべてのデータが「System-Wide Information」セクションに配置されるように指定します。このデータには、さまざまな実行環境に対する以下の制御ブロックが含まれます。



表 17. 影響される制御ブロック

実行環境	影響される制御ブロック	「Target System-Wide Information」セクションの副見出し
DB2®	SQLCA	「DB2® 情報」セクションの「DB2® 制御ブロック」
IMS™	AIB、DIB、および UIB	「IMS™ 情報」セクションの「IMS™ 制御ブロック」

任意の制御ブロック・フィールド説明は、レポートの「System-Wide Information」セクション内の制御ブロックに続き、配置されます。

このサブオプションは、CB と省略できます。

### OpenFiles

通常、レポートの「Event Details」セクションの見出し「Associated Open Files」の下に配置される、イベントに関係するすべてのデータが「System-Wide Information」セクションに配置されるように指定します。

このサブオプションは OF に省略できます。

### StorageAreas

通常、レポートの「Event Details」セクションの見出し「Associated Storage Areas」の下に配置される、イベントに関係するすべてのデータが「System-Wide Information」セクションの副見出し「Storage Areas」に配置されるように指定します。

さらに、ストレージ域のタイプを以下の StorageAreas サブオプションを使用して選択できます。

#### Fmt

このサブオプションは、コンパイラー・リストまたはサイド・ファイルが使用可能な、高水準言語プログラムのストレージ域を指定するため、レポート内に 16 進ダンプの対としてフォーマットされて表示されます。これらの領域には、識別しやすいように「Storage Areas」セクション内で、別個の副見出しが指定されます。

このサブオプションは、F と省略できます。

#### 16 進

このサブオプションは、16 進ダンプ・フォーマットで表示されるストレージ域を指定します。ストレージ域のこれらのタイプは、すべてのイベントに対して結合され、発信元を示すアドレスのラベルが付けられて、アドレスのシーケンスの昇順で「Storage Areas」セクション内の共通副見出し「Hex-Dumped Storage」の下に配置されます。このサブオプションはデフォルトです。

このサブオプションは、H と省略できます。

\*

アスタリスク (\*) を指定することで、すべてのサブオプションの設定を暗黙的に指定できます。

このサブオプションは SA に省略できます。

### メッセージ

通常、レポートの「イベント詳細」セクションの見出し「関連メッセージ」の下に配置される、イベントに関係するすべてのメッセージが「システム全体情報」セクションに配置されるように指定します。

このサブオプションは、M と省略できます。

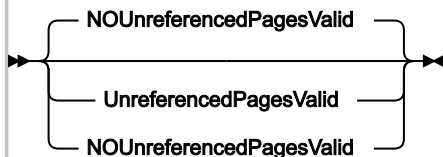
\*

アスタリスク (\*) を指定することで、すべてのサブオプションの設定を暗黙的に指定できます。

SystemWidePreferred オプションの各指定は累積されます。すべての SystemWidePreferred サブオプションを一度にリセットするには、NoSystemWidePreferred オプションを使用します。

## UnreferencedPagesValid

図 339. Syntax



デフォルトでは、Fault Analyzer 参照されていないすべてのストレージ・ページは無効なストレージとして処理されます。これは、ミニダンプのサイズを縮小し、アプリケーションが使用したことのない大量のストレージがレポートに表示されないようにすることで実行されます。ただし、PL/I TEST オプション・データが参照されていないストレージ域ページに存在する場合があります、これが無効と見なされると、Fault Analyzer がプログラムの作業用ストレージの完全なフォーマットを提供できなくなる可能性があります。

NoUnreferencedPagesValid オプションを使用して、参照されていないすべてのストレージ・ページを無効なストレージとして処理します。これは、デフォルトです。

UnreferencedPagesValid オプションを使用して、参照されていないすべてのストレージ・ページを有効なストレージとして処理します。

このオプションは、リアルタイム分析にのみ適用できます。他のすべてのモードの実行では、これは無視されます。

このオプションは、リアルタイム分析で UnreferencedPagesValid が有効である場合に、有効なオプションを示す障害分析レポートのセクションにのみ含まれます。

## 第 33 章. データ域

ユーザー出口に使用可能なデータ域について以下で説明します。ユーザー出口に関する詳細は、[ユーザー出口を使用した Fault Analyzer のカスタマイズ ページ 464](#)を参照してください。

以下のデータ域のソフトコピー・バージョンは、ソフトコピーのサンプル・データ・セット (IDI.SIDISAMP) 内にあります。これはアセンブラー、COBOL、C、および PL/I 用で、それぞれメンバー IDISXPLA、IDISXPLB、IDISXPLC および IDISXPLP です。

以下のデータ域に関する注意:

- 「アクセス権」列には、個々のフィールドを更新するための以下のようなユーザー出口の機能に関する情報が示されます。

### R/W

読み取り/書き込み。このフィールドは、ユーザー出口により更新できます。

### R/O

読み取り専用。

- 数値フィールドは「(nnn)」により識別され、ユーザー出口への入力時に、フィールドの幅全体が埋まるように必ず 0 が左側に埋め込まれます。



**注:** 通常の場合、先行ゼロは、REXX ユーザー出口に提供されるデータ域フィールドには含まれません。

- Fault Analyzer が初期値を指定しないフィールドはすべて、ブランクに初期化されます。



**注:** 通常の場合、REXX ユーザー出口に提供される文字フォーマット・データ域フィールドに 16 進文字が含まれる可能性がない限り、空白文字による埋め込みは、この文字フォーマット・データ域フィールドには含まれません。

- 特に指示されない限り、すべてのフィールドは Fault Analyzer により大文字に変換されます。
- 特に指示されない限り、すべての R/W フィールドは、ヌル文字 (X'00') を区切り文字として使用して切り捨てることができます。Fault Analyzer がパラメーター・リストを初期化するときに切り捨てが行われることはありません。
- COBOL の場合、すべての下線 ( \_ ) をダッシュ ( - ) に置き換えます。

異常終了ジョブ用フィールドのいずれかの初期値を参照するには、以下の JCL ステートメントを追加して、出口トレースを作成してください。

```
//IDITRACE DD SYSOUT=*  
//IDIOPTS DD *  
  Exits(exit_type(NONE))  
/*
```

`exit_type` には、必要とするデータ域に応じて CONTROL、LISTING、FORMAT、REPORT、MSGXPL、END、NOTIFY のいずれかを指定します。トレースの詳細については、[診断トレース ページ 469](#)を参照してください。Exits オプションについて詳しくは、[Exits ページ 595](#)を参照してください。

## 非 REXX ユーザー出口のバッファ・データ・フォーマット

このセクションに記載される情報は、REXX に書き込まれていないユーザー出口にのみ適用できます。

特定文字フィールドのデータ長が最大フィールド・サイズを超える場合、Fault Analyzer はすべてのフィールド・データに十分なサイズのバッファを割り振り、フィールドの相対オフセット・ゼロを起点にして3つのフルワードで以下のようにバッファに関する情報を提供します。

- 最初のフルワードの最初のバイトは X'FF' に設定され、このフィールドがバッファ情報を含むことを示します。他の3バイトは使用されませんが、Fault Analyzer により X'00' に設定されます。
- 2番目のフルワードは、データがそのフィールド自体に指定されたときと同じフォーマットでデータが含まれるバッファのアドレスです。
- 3番目のフルワードは、割り振られたバッファの長さを示します。

このバッファ・フォーマットが適用されるデータ・フィールドは、データ域の記述内で別個に識別されます。

分析制御ユーザー出口は、Fault Analyzer がそのフィールドに対して割り振ったバッファを解放できません。代わりに、フィールド・サイズやバッファ・サイズが不十分な場合、この出口は独自のバッファを割り振り、このフィールドの相対オフセット・ゼロを起点にしてアドレスおよび長さ情報を3つのフルワードで配置します。Fault Analyzer は、割り振られたストレージを後で解放する際に使用するために保存される、元のバッファ・アドレスに依存しません。

ユーザー出口によって割り振られたバッファはユーザーが解放しなければなりません。この解放を行うには、ENV.USER\_1 フィールドまたは ENV.USER\_2 フィールドを使用して、実行されたバッファ割り振りに関する情報が含まれているストレージ域を指し示します。End Processingユーザー出口のような後の出口は、割り振られたストレージの解放に使用できます。

REXX ユーザー出口の場合、Fault Analyzer は必要なバッファの割り振りと解放を自動的に処理し、フィールド・サイズを超えるデータ長に適應します。

## データ域の説明

さまざまなユーザー出口データ域について、以下で説明します。

## CTL - 分析制御ユーザー出口パラメーター・リスト

表 18. CTL データ域

オフセット		タイプ	アクセ	長さ	名前と説明
10進	16進		ス		
0	(0)	CHAR	R/O	4	<b>VERSION</b> パラメーター・リストのバージョン (現在は 0002)。

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
4	(4)	CHAR	R/W	1	<b>EXCLUDE</b>  分析からの除外 (Y/N)。このフィールドが「Y」に設定されている場合、現行の障害は 分析されず、ヒストリー・ファイルは更新されません。  リアルタイム処理およびダンプ登録に対してのみ適用可能。
5	(5)	CHAR	R/W	1	<b>DETAIL_OPT</b>  Detail オプション (S/M/L)。このオプションに関する詳細は、 <a href="#">Detail ページ 586</a> を参照してください。  対話式再分析には適用されません。
6	(6)	CHAR	R/W	1	<b>DEFERREDREPORT_OPT</b>  DeferredReport オプション (Y/N)。このオプションに関する詳細は、 <a href="#">DeferredReport ページ 584</a> を 参照してください。  リアルタイム処理にのみ適用できます。
7	(7)	CHAR	R/W	4	<b>DETAIL_OPT_EXTRA_SOURCE</b>  リアルタイムまたはバッチ再分析レポート内の、CSECT オフセット (nnnn) と一致するソース行またはソース・ステートメントの前後に 含めるべき追加ソース・コード行またはステートメントの数。追加の ソース行またはステートメントはレポート詳細セクションにのみ組み 込まれます。  デフォルトは 0005 です。  対話式再分析には適用されません。
8	(8)	CHAR	R/O	7	(予約済み)
18	(12)	CHAR	R/W	10	<b>RETAINDUMP_OPT</b>  RetainDump オプション。このオプションに関する詳細は、 <a href="#">RetainDump</a>

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
					<p>ページ 627を参照してください。(このフィールドに指定されているフォーマットは、通常の RetainDump オプション構文のフォーマットと同じ、すなわち「AUTO」または「ALL」です。)</p> <p>リアルタイム処理にのみ適用できます。</p>
28	(1C)	CHAR	R/O	44	(予約済み)
72	(48)	CHAR	R/O	120	<b>INCLUDE_CRITERION</b>
					<p>組み込み基準。最後に一致し処理された Include オプション基準。</p> <p>Fault Analyzer は、基準の長さがこのフィールドに指定可能な最大サイズを超える場合に使用される、バッファ・データ・フォーマットを許可します。詳しくは、<a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット</a> ページ 636を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。</p> <p>リアルタイム処理にのみ適用できます。</p>
192	(C0)	CHAR	R/O	120	<b>EXCLUDE_CRITERION</b>
					<p>除外基準。最後に一致し処理された Exclude オプション基準。</p> <p>Fault Analyzer は、基準の長さがこのフィールドに指定可能な最大サイズを超える場合に使用される、バッファ・データ・フォーマットを許可します。詳しくは、<a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット</a> ページ 636を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。</p> <p>リアルタイム処理にのみ適用できます。</p>
312	(138)	CHAR	R/O	5	(予約済み)
317	(13D)	CHAR	R/W	1	<b>SOURCE_OPT</b>

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ	長さ	名前と説明
10 進	16 進		ス		
					Source オプション (Y/N)。このオプションに関する詳細は、 <a href="#">Source ページ 629</a> を参照してください。  リアルタイム処理にのみ適用できます。
318	(13E)	CHAR	R/O	6	(予約済み)
324	(144)	CHAR	R/W	1	<b>PRINTINACTIVECOBOL_OPT</b>  PrintInactiveCOBOL オプション (Y/N)。このオプションに関する詳細は、 <a href="#">PrintInactiveCOBOL ページ 624</a> を参照してください。  対話式再分析には適用されません。
325	(145)	CHAR	R/O	51	(予約済み)

**フィールド・フォーマットおよび使用法:**

- 以下の情報は、IDIJAVA\_\*フィールドを除く、このデータ域のすべての IDI\*\_PRE、IDI\*\_JOB、および IDI\*\_sフィールドに適用されます。

各フィールドでは、データ・セット名は左寄せ、かつ 45 文字の境界にブランクが挿入されて指定されます。未使用のスペースは、すべてのブランクに指定されます。

**Fault Analyzer**

は、データ・セットの数がこのフィールドに指定可能な最大数を超える場合に使用される、バッファ・データ・フォーマットを許可します。バッファは Fault Analyzer によって割り振られるときに 25 個の追加データ・セット名を保持するだけの十分なサイズで割り振られます。詳しくは、[非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636](#)を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。

- 以下の情報は、このデータ域のすべての IDI\*\_PREフィールドに適用されます。

分析制御ユーザー出口は、いずれかのデータ・セットを割り振り解除すること、または追加データ・セットを割り振ることを選択できます。これらのフィールドは、事前割り振りされたデータ・セット名を含み、情報表示のみの目的で提供されます。このフィールドのパス名を変更しても、Fault Analyzer に認識されません。代わりに Fault Analyzer は、標準オペレーティング・システム・インターフェースを使用して、事前割り振りされたデータ・セットを再決定します。

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ	長さ	名前と説明
10 進	16 進		ス		
<p>• 以下の情報は、このデータ域のすべての IDI*_JOB または IDI*_CFG フィールドに適用されます。</p> <p>分析制御ユーザー出口は、任意またはすべてのデータ・セット名を追加、削除、または変更できます。出口から戻るとき、データ・セット名は 1 つ以上の空白またはコンマで区切られます。また、データ・セット名は境界に位置合わせされます。これらのパス名について、Fault Analyzer は戻された名前のリストを使用します。</p>					
376	178	CHAR	R/O	5400	<b>IDIADATA_PRE</b>  IDIADATA 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDIADATA により初期化されます。
5776	(1690)	CHAR	R/W	5400	<b>IDIADATA_JOB</b>  ユーザー・オプション・ファイル (IDIADATA) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDIADATA により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
11176	(2BA8)	CHAR	R/W	5400	<b>IDIADATA_CFG</b>  IDIADATA 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDIADATA データ・セットを組み込むために、IDICNFxx により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
16576	(40C0)	CHAR	R/O	5400	<b>IDILC_PRE</b>



表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
					<p>IDILC 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDILC により初期化されます。</p> <p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>
21976	(55D8)	CHAR	R/W	5400	<p><b>IDILC_JOB</b></p> <p>ユーザー・オプション・ファイル (IDILC) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDILC により初期化されま す。</p> <p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>
27376	(6AF0)	CHAR	R/W	5400	<p><b>IDILC_CFG</b></p> <p>IDILC 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDILC データ・セットを組み込むために、IDICNFxx により初期化されます。</p> <p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>
32776	(8008)	CHAR	R/O	5400	<p><b>IDILCOB_PRE</b></p> <p>IDILCOB 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDILCOB により初期化されます。</p> <p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
38176	(9520)	CHAR	R/W	5400	<b>IDILCOB_JOB</b>  ユーザー・オプション・ファイル (IDILCOB) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDILCOB により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
43576	(AA38)	CHAR	R/W	5400	<b>IDILCOB_CFG</b>  IDILCOB 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDILCOB データ・セットを組み込むために、IDICNFxx により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
48976	(BF50)	CHAR	R/O	5400	<b>IDILCOBO_PRE</b>  IDILCOBO 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDILCOBO により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
54376	(D468)	CHAR	R/W	5400	<b>IDILCOBO_JOB</b>  ユーザー・オプション・ファイル (IDILCOBO) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDILCOBO により初期化されます。

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
<p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>					
59776	(E980)	CHAR	R/W	5400	<b>IDILCOBO_CFG</b>  IDILCOBO 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDILCOBO データ・セットを組み込むために、IDICNFxx により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
65176	(FE98)	CHAR	R/O	5400	<b>IDILANGX_PRE</b>  IDILANGX 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDILANGX により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
70576	(113B0)	CHAR	R/W	5400	<b>IDILANGX_JOB</b>  ユーザー・オプション・ファイル (IDILANGX) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイ ル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDILANGX により初期化さ れます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
75976	(128C8)	CHAR	R/W	5400	<b>IDILANGX_CFG</b>  IDILANGX 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
					DataSets オプションを介して指定されたすべての IDILANGX データ・セットを組み込むために、IDICNFxx により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
81376	(13DE0)	CHAR	R/O	5400	<b>IDILPLI_PRE</b>  IDILPLI 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDILPLI により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
86776	(152F8)	CHAR	R/W	5400	<b>IDILPLI_JOB</b>  ユーザー・オプション・ファイル (IDILPLI) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDILPLI により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
92176	(16810)	CHAR	R/W	5400	<b>IDILPLI_CFG</b>  IDILPLI 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDILPLI データ・セットを組み込むために、IDICNFxx により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” ページ 639 を参照してください。
97576	(17D28)	CHAR	R/O	1024	(予約済み)
98600	(18128)	CHAR	R/O	5400	<b>IDILPLIE_PRE</b>

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
					<p>IDILPLIE 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDILPLIE により初期化されます。</p> <p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>
104000	(19640)	CHAR	R/W	5400	<p><b>IDILPLIE_JOB</b></p> <p>ユーザー・オプション・ファイル (IDILPLIE) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDILPLIE により初期化されます。</p> <p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>
109400	(1AB58)	CHAR	R/W	5400	<p><b>IDILPLIE_CFG</b></p> <p>IDILPLIE 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDILPLIE データ・セットを組み込むために、IDICNFxx により初期化されます。</p> <p>詳細については、“<a href="#">「フィールド・フォーマットおよび使用法」</a>” ページ 639 を参照してください。</p>
114800	(1C070)	CHAR	R/W	256	<p><b>LOCALE</b></p> <p>Locale オプションのロケール名。このオプションに関する詳細は、<a href="#">Locale ページ 609</a> を参照してください。</p>
115056	(1C170)	CHAR	R/W	1	<p><b>FADATE</b></p>

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
					Locale オプションの FADATE サブオプション (Y/N)。このサブオプションに関する詳細は、 <a href="#">Locale ページ 609</a> を参照してください。
115057	(1C171)	CHAR	R/O	5400	<b>IDISYSDB_PRE</b>  IDISYSDB 事前割り振りデータ・セット。このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのデータ・セットを組み込むために IDISYSDB により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” <a href="#">ページ 639</a> を参照してください。
120457	(1D689)	CHAR	R/W	5400	<b>IDISYSDB_JOB</b>  ユーザー・オプション・ファイル (IDISYSDB) に指定された IDIOPTS データ・セット。このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer データ・セットを組み込むために IDISYSDB により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” <a href="#">ページ 639</a> を参照してください。
125857	(1EBA1)	CHAR	R/W	5400	<b>IDISYSDB_CFG</b>  IDISYSDB 構成メンバーで指定された IDICNFxx データ・セット。このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDISYSDB データ・セットを組み込むために、IDICNFxx により初期化されます。  詳細については、“ <a href="#">「フィールド・フォーマットおよび使用法」</a> ” <a href="#">ページ 639</a> を参照してください。
131257	(200B9)	CHAR	R/W	56	<b>IDITRACE</b>  これは、IDITRACE トレースを動的に開始/停止するために使用しません。

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ	長さ	名前と説明
10 進	16 進		ス		
					<p><b>ONx</b></p> <p>SYSOUT クラス x へのトレースを開始します。x が指定されていない場合、デフォルトは '*' です。</p> <p><b>&lt;dsn&gt;</b></p> <p>指定のデータ・セット名のトレースを開始します。このデータ・セット名は完全修飾データ・セット名でなければなりません、単一引用符で囲むではありません。このデータ・セットが PDS または PDSE の場合は、メンバー名を括弧で囲んで指定する必要があります。</p> <p><b>OFF</b></p> <p>トレースを停止します。</p>
131313	(200F1)	CHAR	R/O	5400	<p><b>IDIJAVA_PRE</b></p> <p>IDIJAVA 事前割り振りされたパス名。</p> <p>このフィールドは、Fault Analyzer DDname に事前割り振りされたすべてのパス名を組み込むために IDIJAVA により初期化されます。</p> <p>パス名はブランク区切りリストとして指定されます。パス名に単一引用符、コンマまたはブランクが含まれる場合、名前は単一引用符で囲まれます。パス名内の単一引用符はすべて、2 つ重ねられます。</p> <p><b>Fault Analyzer</b></p> <p>では、パス名のリストがこのフィールドに指定可能な最大サイズを超えた場合、バッファ・データ・フォーマットを使用できます。詳しくは、<a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a>を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。</p> <p>分析制御ユーザー出口は、いずれかのパス名を割り振り解除すること、または追加パス名を割り振ることを選択できます。このフィールドのパス名を変更しても、Fault Analyzer に認識されません。代わりに</p>

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
<p>Fault Analyzer は、標準オペレーティング・システム・インターフェースを使用して、事前割り振りされたパス名を再決定します。</p>					
136713	(21609)	CHAR	R/W	5400	<p><b>IDIJAVA_JOB</b></p> <p>ユーザー・オプション・ファイル (IDIJAVA) に指定された IDIOPTS パス名。</p> <p>このフィールドは、ユーザー・オプション・ファイル内の DataSets オプションを介して指定されたすべての Fault Analyzer パス名を組み込むために IDIJAVA により初期化されます。</p> <p>パス名は空白区切りリストとして指定されます。パス名に単一引用符、コンマまたは空白が含まれる場合、名前は単一引用符で囲まれます。パス名内の単一引用符はすべて、2 つ重ねられます。出口が単一引用符、コンマまたは空白を含むパス名を提供する場合は、パス名内のすべての単一引用符が 2 つ重ねられた、単一引用符で囲まれたパス名を提供する必要があります。</p> <p>Fault Analyzer では、パス名のリストがこのフィールドに指定可能な最大サイズを超えた場合、バッファ・データ・フォーマットを使用できます。詳しくは、<a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a>を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。</p> <p>分析制御ユーザー出口は、任意またはすべてのパス名を追加、削除、または変更できます。パス名は 1 つ以上の空白またはコンマで区切り、上で説明したように単一引用符を保護する必要があります。</p> <p>これらのパス名について、Fault Analyzer は戻された名前のリストを使用します。</p>
142113	(22B21)	CHAR	R/W	5400	<p><b>IDIJAVA_CFG</b></p> <p>IDIJAVA 構成メンバーで指定された IDICNFxx パス名。</p>



表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ	長さ	名前と説明
10 進	16 進		ス		
					<p>このフィールドは、Fault Analyzer 構成メンバー内の DataSets オプションを介して指定されたすべての IDIJAVA パス名を組み込むために IDICNFxx により初期化されます。</p> <p>パス名は空白区切りリストとして指定されます。パス名に単一引用符、コンマまたは空白が含まれる場合、名前は単一引用符で囲まれます。パス名内の単一引用符はすべて、2 つ重ねられます。出口が単一引用符、コンマまたは空白を含むパス名を提供する場合は、パス名内のすべての単一引用符が 2 つ重ねられた、単一引用符で囲まれたパス名を提供する必要があります。</p> <p>Fault Analyzer では、パス名のリストがこのフィールドに指定可能な最大サイズを超えた場合、バッファードータフォーマットを使用できます。詳しくは、<a href="#">非 REXX ユーザー出口のバッファードータフォーマット ページ 636</a>を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。</p> <p>分析制御ユーザー出口は、任意またはすべてのパス名を追加、削除、または変更できます。パス名は 1 つ以上の空白またはコンマで区切り、上で説明したように単一引用符を保護する必要があります。</p> <p>これらのパス名について、Fault Analyzer は戻された名前を使用します。</p>
147513	(24039)	CHAR	R/O	5400	<p><b>STEPLIB</b></p> <p>STEPLIB データ・セットの連結 (STEPLIB が定義されていない場合は、JOB LIB データ・セットの連結)。</p> <p>これは、バッチ再分析以外の異常終了時の連結で、再分析ジョブの STEPLIB データ・セットです。</p> <p>データ・セット名は空白区切りリストとして指定されます。</p> <p>Fault Analyzer では、パス名のリストがこのフィールドに指定可能な最大サイズを超えた場合、バッファードータフォーマットを使用できます。詳しく</p>

表 18. CTL データ域

(続く)

オフセット		タイプ	アクセ	長さ	名前と説明
10 進	16 進		ス		
					くは、非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636 を参照してください。このフィールドのフォーマット は、REXX 出口のユーザーに対し透過的です。

## ENV - 共通出口環境情報

表 19. ENV データ域

オフセット		タイプ	アクセ	長さ	名前と説明	注
10 進	16 進		ス			
0	(0)	CHAR	R/O	4	<b>VERSION</b>  パラメーター・リストのバージョン (現在は 0005)。  変更の要約:  <b>0003</b>  6 桁の CICS® システム異常終了コードのサ ポート。  <b>0004</b>  障害項目の有効期限の制御をサポートする ための、LOCK_FLAG の 1 文字から 2 文字への変更。	
4	(4)	CHAR	R/O	1	<b>EXIT_CALL_TYPE</b>  以下のいずれかとして呼び出される出口のタイプを示す ユーザー出口呼び出しタイプ。  <b>C</b>  分析制御  <b>L</b>  コンパイラー・リスト読み取り	

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					<b>R</b>	
					バッチ・レポート調整	
					<b>M</b>	
					メッセージおよび異常終了コードの説明	
					<b>F</b>	
					フォーマット	
					<b>E</b>	
					終了処理	
					<b>N</b>	
					通知	
					<b>I</b>	
					IDIUTIL インポート	
					<b>D</b>	
					IDIUTIL 削除	
					<b>H</b>	
					IDIUTIL ListHF	
					<b>P</b>	
					IDIUTIL ListHFDup	
					<b>X</b>	
					ダンプ登録分析制御出口	
					<b>Y</b>	
					ダンプ登録通知出口	
					<b>Z</b>	
					障害項目リフレッシュ終了処理出口	
					この情報は、ユーザー出口が出口タイプ間で共通のものであることを示しています。	
5	(5)	CHAR	R/O	8	<b>FAULT_ID</b>	

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					障害 ID。  End Processing ユーザー出口と通知ユーザー出口の場合に、障害が重複 障害である (NFY.NFYTYPE='N' または 'F') と、このフィールドは重複障害 ID を含みます。  通知ユーザー出口の場合に、障害が重複障害でない (NFY.NFYTYPE='C' または 'R') と、このフィールドは割り当てられた障害 ID を含みます。  他のすべての出口では、このフィールドは初期設定され ません。	
13	(D)	CHAR	R/O	10	<b>ABEND_DATE</b>  フォーマット YYYY/MM/DD での、異常終了の日付。	
23	(17)	CHAR	R/O	8	<b>ABEND_TIME</b>  HH:MM:SS 形式の異常終了時刻 (24 時間クロック値)。	
31	(1F)	CHAR	R/O	1	<b>REALTIME</b>  リアルタイム実行 (Y/N)。	
32	(20)	CHAR	R/O	8	<b>SYSTEM_NAME</b>  システム名。  <ul style="list-style-type: none"> <li>• ENV.VERSION が 1 の場合、このフィールドは、CICS® トランザクション障害では APPLID を、その他の障害では MVS™ システム名を含み ます。</li> <li>• ENV.VERSION が 1 を超える場合、このフィールドは常に MVS™ システム名を含み、CICS®</li> </ul>	

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					トランザクションのアプリケーション ID は、このフィールドではなく ENV.APPLID に設定されます。	
40	(28)	CHAR	R/O	8	<b>JOB_NAME</b>  ジョブ/開始済みタスク名。	
48	(30)	CHAR	R/O	8	<b>EXEC_PGM_NAME</b>  EXEC プログラム名。	注 2 を参照
56	(38)	CHAR	R/O	8	<b>USER_ID</b>  ユーザー ID。	注 2 を参照
64	(40)	CHAR	R/O	4	(予約済み)	
68	(44)	CHAR	R/O	8	<b>ABEND_MODULE_NAME</b>  異常終了モジュール名。この値は、異常終了 (複数の異 常終了の場合は最初の異常終了) が発生したモジュール の名前を識別するものです。	
76	(4C)	CHAR	R/O	4	<b>CICS_TRANSACTION_ID</b>  CICS® トランザクション ID。	
80	(50)	CHAR	R/O	5	<b>CICS_TASK_NUMBER</b>  CICS® タスク番号。	
85	(55)	CHAR	R/O	1	<b>JOB_TYPE</b>  以下のいずれかとして分析されるジョブのタイプ。  <b>B</b>  バッチ・ジョブ、または [File] メ ニュー・オプション 5 によって対話式に分析される MVS™ ダン プ。	

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					<b>S</b> 開始済みタスク	
					<b>T</b> TSO	
					<b>C</b> CICS® トランザクション	
					<b>I</b> CICS® システムのダンプ分析 (CICS® シス テム・ダンプのダンプ登録を含む)	
					<b>D</b> ダンプ登録 (CICS® システム・ダンプ以外)	
86	(56)	CHAR	R/O	1	<b>JOB_CLASS</b> ジョブ実行クラス。	注 2 を参照
87	(57)	CHAR	R/O	3	<b>ACCOUNTING_FIELDS</b> JCT ACTJNFLD からの、ジョブ・アカウントING・フィールドの数 (nnn)。	注 2 を参照
90	(5A)	CHAR	R/O	144	<b>ACCOUNTING_INFO</b> JCT ACTACCNT からの、ジョブ・アカウントING情報。  このフィールドは、ジョブ・アカウントING用であ り、1 バイトの長さのフィールドに、 (ENV.ACCOUNTING_FIELDS フィールドが示す数値分の繰り返される) フィールド内容が続きます。	注 2 を参照

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
						<p>ステップ・アカウンティングの場合、このフィールドにはステップごとに以下の項目が(順番に)含まれます。</p> <ul style="list-style-type: none"> <li>• 最大ステップ実行時間 (3 バイト)。</li> <li>• ステップに含まれるフィールドの数 (1 バイト)。</li> <li>• 長さフィールド (1 バイト)。</li> <li>• フィールドの内容。</li> </ul> <p>最後 2 つのフィールドは、4 バイト目にあるフィールド数だけ繰り返されます。</p> <p>(2 進数の長さフィールドのような) 非印字文字はすべて、ピリオドで表されます。</p>
234	(EA)	CHAR	R/W	4	<b>USER_1</b>	<p>ユーザー・フィールド</p> <p>1. あるユーザー出口から別のユーザー出口に情報を渡すためにこのフィールドを使用できます。Fault Analyzer はユーザー出口への呼び出しと呼び出しの間にこのフィールドを再初期化しません。また、大文字変換は行われません。このフィールドのヌル文字 (X'00') による切り捨ては許可されていません。</p>
238	(EE)	CHAR	R/W	4	<b>USER_2</b>	<p>ユーザー・フィールド 2。USER_1 と同じです。</p>
242	(F2)	CHAR	R/O	1	(予約済み)	
243	(F3)	CHAR	R/W	1	<b>LOOPPROTECTION_OPT</b>	<p>LoopProtection オプション (Y/N)。</p>

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					<ul style="list-style-type: none"> <li>この値が Y に設定されると、LoopProtection オプションが有効になっているときと同じ効果があります。</li> <li>この値が N に設定されると、NoLoopProtection オプションが有効になっているときと同じ効果があります。</li> </ul> <p>このフィールドを N に設定して Fault Analyzer のループ/待機保護機構を非アクティブにすることのみが可能で す。このフィールドを Y に設定しても、その設定は無視 されます。</p> <p>このオプションについて詳しくは、<a href="#">LoopProtection ページ 609</a>を参照してください。</p> <p>このオプションは、ユーザー出口で変更できます。</p>	
244	(F4)	CHAR	R/O	4	(予約済み)	
248	(F8)	ADDRESS	R/O	4	<b>WRITE_ROUTINE_EP</b>	
					ルーチン・エンタリー・ポイント・アドレスの書き込 み。	
252	(FC)	CHAR	R/O	4	(予約済み。常に X'00000000'が含まれます)	
256	(100)	CHAR	R/O	1	<b>INVOCATION_EXIT</b>	
					Fault Analyzer をリアルタイム異常終了分析用に暗黙的に呼び出した り、IDISNAP で明示的に呼び出したりするために使用さ れる呼び出し出口のタイプ。	
					<b>C</b>	
					CICS® XPCABND 出口 (IDIXCX53)	
					<b>D</b>	
					CICS® XDUREQ 出口 (IDIXCX53)	



表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					<b>E</b> CICS® LE CEEEXTAN 出口、IDIXCCEE	
					<b>L</b> LE CEEEXTAN 出口、IDIXCEE	
					<b>M</b> MVS™ IEAVTABX 変更オプション/抑止ダンプ出 口、IDIXDCAP	
					<b>S</b> Fault Analyzer プログラム SNAP インターフェース (IDISNAP)	
					<b>P</b> MVS™ IEAVTSEL ポスト・ダンプ出口、IDIXTSEL(Fault Analyzer ダンプ登録)	
257	(101)	CHAR	R/O	8	<b>STEP_NAME</b>  ジョブ/開始済みタスク・ステップ名。	注 2 を参照
265	(109)	CHAR	R/O	8	<b>JOB_ID</b>  JES ジョブ ID。	注 2 を参照
273	(111)	CHAR	R/O	8	<b>IMS_PROGRAM_NAME</b>  IMS™ プログラム 名。この名前は、当該環境において DFSPRPX0 モジュールがロードされている場合に使用できます。	注 2 を参照
281	(119)	CHAR	R/W	8	<b>USER_NAME</b>  ユーザー名フィールド。	注 5 を参照
289	(121)	CHAR	R/W	40	<b>USER_TITLE</b>	注 5 を参照

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
						ユーザー・タイトル・フィールド。
329	(149)	CHAR	R/O	8	<b>APPLID</b>	
						アプリケーション ID。CICS® トランザクション障害 (ENV.JOB_TYPE = C) であり、さらに ENV.VERSION が 1 より大きい場合にのみ適用されます。この場合、関連する CICS® APPLID が含まれます。
337	(151)	CHAR	R/O	4	<b>TERMID</b>	
						CICS® 端末 ID。
341	(155)	CHAR	R/O	8	<b>NETNAME</b>	
						CICS® 端末ネット名。
349	(15D)	CHAR	R/O	8	<b>TCB_ADDRESS</b>	
						分析された TCB アドレス。
						 <b>注:</b> CICS® トランザクションの異常終了用の TCB アドレスは、QR TCB を指しています。QR TCB では、異常終了しているトランザクションを実行しなくなります。
357	(165)	CHAR	R/O	8	<b>CSA_ADDRESS</b>	
						CICS® CSA アドレス。
						このフィールドは、CICS® トランザクションの異常終了または CICS® システム・ダンプ分析においてのみ使用可能です。
365	(16D)	CHAR	R/O	8	<b>TCA_ADDRESS</b>	
						CICS® TCA アドレス。

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					このフィールドは、CICS® トランザクションの異常終了の分析においてのみ使用可能です。	
373	(175)	CHAR	R/W	44	<b>IDIHIST</b>  障害ヒストリー・ファイル名。  このフィールドは、IDIUTIL バッチ・ユーティリティー・ユーザー出口には使用されません。それ以外のユーザー出口の場合、このフィールドは Fault Analyzer により初期化され、IDIHIST DDname で指定されるヒストリー・ファイル名になります (この名前は事前割り振りされているか、または DataSets オプションを介して指定します)。  リアルタイム処理のとき、分析制御ユーザー出口または End Processing ユーザー出口は、このデータ・セット名を別のヒストリー・ファイルのデータ・セット名に変更することもでき、それ以後、そのデータ・セット名が代わりに使用されます。それ以外のすべてのユーザー出口タイプまたは処理モードの場合、このフィールドは読み取り専用です。  重複障害 (NFY.NFYTYPE='N' または 'F') の場合に呼び出される通知ユーザー出口に関しては、この名前は、ENV.FAULT_ID で識別される重複障害が検出されたヒストリー・ファイルの名前です。	
417	(1A1)	CHAR	R/O	6	<b>ABEND_CODE</b>  初期の (または唯一の) 異常終了コード:	

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					<ul style="list-style-type: none"> <li>ENV.JOB_TYPE = C の場合、このコードは 4 文字の CICS® トランザクション異常終了コードです。</li> <li>それ以外の場合、ENV.JOB_TYPE = I であれば、このコードは 6 文字の CICS® システム異常終了コードです。</li> <li>それ以外の場合は、システム異常終了コード (Sxxx) またはユーザー異常終了コード (Unnnn) のいずれかです。</li> </ul>	
423	(1A7)	CHAR	R/O	6	<b>CPU_HSECONDS</b>  分析レポート生成の終了時までに Fault Analyzer によって使用された合計 CPU 時間 (1/100 秒単位)。	
					 <b>注:</b> このフィールドは IDIUTIL バッチ・ユーティリティー・ユーザー出口でのみ使用可能です。また、PARM='UPDINDEX' オプション (ヒストリー・ファイル <a href="#">\$\$INDEX データのキャッシュ</a> ページ 320 を参照) を使用して、IDIS サブシステムによって管理される PDSE ヒストリー・ファイルの場合にのみ、データが提供されます。	
429	(1AD)	CHAR	R/O	9	<b>CICS_VRM</b>  CICS® リリース・レベル (VnnRnnMnn フォーマット)	注 3 を参照
438	(1B6)	CHAR	R/O	9	<b>DB2_VRM</b>  DB2® リリース・レベル (VnnRnnMnn フォーマット)	注 3 を参照
447	(1BF)	CHAR	R/O	9	<b>IMS_VRM</b>  IMS™ リリース・レベル (VnnRnnMnn フォーマット)	注 3 を参照
456	(1C8)	CHAR	R/O	9	<b>ZOS_VRM</b>	注 3 を参照

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
z/OS® リリース・レベル (VnnRnnMnn フォーマット)						
465	(1D1)	CHAR	R/W	2	<b>LOCK_FLAG</b>	注 5 を参照
<p>障害項目ロック・フラグこのフラグの目的は、現行障害項目が誤って削除されることを防ぐメカニズムを提供することです。障害項目有効期限制御の指定など、このフラグについて詳しくは、<a href="#">障害項目情報の表示 ページ 134</a>を参照してください。</p> <p>このフィールドへの変更内容がヒストリー・ファイル障害項目に反映されるのは、リアルタイム分析の処理時に分析制御ユーザー出口が呼び出される場合か、MVS ダンプ・データ・セットの対話式再分析から新規障害項目が作成される場合のみです。</p> <p>デフォルトでは、ロック・フラグはブランクに設定され、障害項目の削除は妨げられません。</p> <p>このフィールドには、次のように印刷可能文字を指定できます。</p> <ul style="list-style-type: none"> <li>• 非印刷可能文字が指定されると、[/] に変更されます。</li> <li>• 小文字が指定されると、大文字に変換されます。</li> </ul> <p>ロック・フラグの最終値は、IDIXLOCK ロック・フラグ制御出口によって行われた変更に応じて変化します。<a href="#">障害項目ロック・フラグ値の制御 ページ 133</a> を参照してください。</p>						
467	(1D3)	CHAR	R/O	5	<b>DUPLICATE_COUNT</b>	
<p>合計重複数 (現在の障害は含まれない)。</p> <p>この合計は、同じヒストリー・ファイル内に記録されている重複障害のすべてのインスタンスを、NoDup(Normal(...))</p>						

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					が有効であった期間にわたって集計することで決定されます。検出されて記録された障害のうち、重複基準が現在の障害と適合するものはいずれも、1 つのインスタンスとして計上されます。また、障害に対して重複が記録された場合は、その障害の重複カウントも追加されます。	
472	(1D8)	CHAR	R/O	8	<b>POF_MODULE_NAME</b>  障害点モジュール名。	
480	(1E0)	CHAR	R/O	10	<b>POF_MODULE_LKED_DATE</b>  障害点モジュールのリンク・エディット日付 (YYYY/MM/DD 形式)。	
490	(1EA)	CHAR	R/O	8	<b>POF_MODULE_LKED_TIME</b>  障害点モジュールのリンク・エディット時刻 (HH:MM:SS 形式)。	
498	(1F2)	CHAR	R/O	8	<b>POF_CSECT_NAME</b>  障害点プログラムまたは CSECT 名。プログラム名の長さが 8 文字を超える場合、このフィールドには最初の 8 文字のみが格納されます。この場合は、代わりに POF_LONG_PGM_NAME フィールドを使用します。	
506	(1FA)	CHAR	R/O	10	<b>POF_CSECT_OFFSET</b>  障害点エントリー・ポイント、プログラム/CSECT またはロード・モジュール・オフセット (10 進数)。	
516	(204)	CHAR	R/O	44	<b>POF_LOADED_FROM</b>  障害点プログラムのロード元のデータ・セット名。	
560	(230)	CHAR	R/O	44	<b>EXEC_LOADED_FROM</b>	

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					EXEC プログラムのロード元のデータ・セット名。	
604	(25C)	CHAR	R/O	10	<b>DUP_DATE</b>  最新の重複障害の日付 (YYYY/MM/DD 形式)。	
614	(266)	CHAR	R/O	8	<b>DUP_TIME</b>  HH:MM:SS 形式による最新の重複障害時刻 (24 時間クロック値)。	
622	(26E)	CHAR	R/O	8	<b>GROUP_ID</b>  セキュリティー・サーバー・デフォルト・グループ ID。	注 2 を参照
630	(276)	CHAR	R/O	6	<b>INVOCATION_ABEND_CODE</b>  最終の (または唯一の) 異常終了コード。Fault Analyzer が呼び出された際の異常終了コード。  <ul style="list-style-type: none"> <li>• ENV.JOB_TYPE = C の場合、このコードは 4 文字の CICS® トランザクション異常終了コードです。</li> <li>• それ以外の場合、ENV.JOB_TYPE = I であれば、このコードは 6 文字の CICS® システム異常終了コードです。</li> <li>• それ以外の場合は、システム異常終了コード (Sxxx) またはユーザー異常終了コード (Unnnn) のいずれかです。</li> </ul>	
636	(27C)	CHAR	R/O	10	<b>MINIDUMP_PAGES</b>  ミニダンプ・ページ数 (nnnnnnnnnn)。	



**注:** リアルタイム処理の場合、フォーマット設定  
ユーザー出口の実行時にストレージが参照され

表 19. ENV データ域

(続く)


オフセット	タイプ	アクセ	長さ	名前と説明	注	
10 進	16 進	ス				
				 ることでさらに生成されるミニダンプ・ページは、この値に含まれません。		
646	(286)	CHAR	R/O	8	<b>IDIRLOAD_DD</b>	注 1 を参照
				<p>IDIRLOAD DDname。Fault Entry List 画面 I または B 行コマンドによって開始された初期 RFR 障害項目の再分析中、STEPLIB または JOBLIB データ・セット連結が保存された場合、Fault Analyzer は STEPLIB または JOBLIB 連結をシステム生成 DDname (SYSnnnnn) に割り振って、このフィールドをその DDname で初期化します。このアクションの目的は、リアルタイム分析中に実行できなかった Binder CSECT マッピングを自動的に実行することです。</p> <p>それ以外の場合、このフィールドはデフォルトで IDIRLOAD に初期化されます。</p> <p>分析制御ユーザー出口は、次のいずれかのアクションを選択できます。</p> <ul style="list-style-type: none"> <li>• このフィールドをクリアして、初期 RFR 障害項目の再分析中、自動処理を無効にします。</li> <li>• 1 つ以上のロード・ライブラリーをユーザーが決定した DDname に割り振って、このフィールドをその DDname に変更します。</li> <li>• 1 つ以上のロード・ライブラリーを IDIRLOAD DDname に割り振ります。</li> </ul> <p>分析制御ユーザー出口からの戻り時に、Fault Analyzer はこのフィールドで指定された DDname に割り振られたデータ・セットを IDIRLOAD 割り振りとして使用します。詳しくは、<a href="#">「CSECT</a></p>		



表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明	注
10 進	16 進					
					<p>マッピングに <a href="#">IDIRLOAD DDname の使用 ページ 392</a>」を参照してください。</p> <p>分析制御以外のユーザー出口によってこのフィールドに加えられた変更は、無視されます。</p> <p>指定された DDname は、大/小文字の区別がありません。</p>	
654	(28E)	CHAR	R/O	8	<p><b>ABEND_REASON_CODE</b></p> <p>ENV.ABEND_CODE に関連付けられた 16 進理由コード。</p> <p><a href="#">図 223 : Fault Analyzer 分析の出口点 (IDIDA) ページ 465</a>にあるように後続の分析処理を実行するすべてのユーザー出口で使用できます。</p>	
662	(296)	CHAR	R/O	8	<p><b>LOCK_USERID</b></p> <p>ENV.LOCK_FLAG フィールドを最後に変更したユーザー ID。</p>	
670	(29E)	CHAR	R/O	10	<p><b>ORIGINAL_DATE</b></p> <p>元の障害の日付 (フォーマット YYYY/MM/DD)。</p>	注 4 を参照
680	(2A8)	CHAR	R/O	8	<p><b>ORIGINAL_TIME</b></p> <p>元の障害の時刻 (フォーマット HH:MM:SS、24 時間クロック値)。</p>	注 4 を参照
688	(280)	CHAR	R/O	1	<p><b>ASSOCIATED_DUMP_TYPE</b></p> <p>当該障害項目に関連付けられているダンプ・データ・セットのタイプ。次のいずれかです。</p> <p><b>P</b></p> <p>SLIP ダンプ</p>	

表 19. ENV データ域

(続く)

オフセット		タイプ	アクセ	長さ	名前と説明	注
10 進	16 進		ス			
					S SDUMP (SVC ダンプ)	
					T TDUMP (トランザクション・ダンプ)	
					X XDUMP (拡張ダンプ)	
689	(281)	CHAR	R/O	44	<b>ASSOCIATED_DUMP_DSN</b>  当該障害項目に関連付けられているダンプ・データ・セットの名前。	
733	(2DD)	CHAR	R/O	32	<b>POF_LONG_PGM_NAME</b>  障害時点の長いプログラム名 (32 文字まで)。このフィールドは、プログラム名の長さが 8 文字を超える場合にのみ初期化されます。それ以外の場合は、代わりに POF_CSECT_NAME フィールドを使用します。	
765	(2FD)	CHAR	R/O	775	(予約済み)	



注:

1

リアルタイム処理には適用されません。

2

ダンプ登録ユーザー出口では使用できません。

3

分析制御ユーザー出口の呼び出し時には使用できません。

4

障害が重複障害である場合にのみ使用できます。



5

更新は、レポートの生成前に起動された出口を使用するときのみ、リアルタイム処理中に障害項目に保存されます。

## EPC - End Processingユーザー出口パラメーター・リスト

表 20. EPC データ域

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
0	(0)	CHAR	R/O	4	<b>VERSION</b>  パラメーター・リストのバージョン (現在は 0002)。
4	(4)	CHAR	R/O	4	(予約済み)
8	(8)	CHAR	R/W	1	<b>IS_DUPLICATE</b>  重複する障害 (Y/N)。
9	(9)	CHAR	R/W	1	<b>SUPPRESS_MINIDUMP</b>  ミニダンプの抑止 (Y/N)。  このフラグは、MaxMinidumpPages オプションの設定とこの障害で予想されるミニダンプのページ数に基づいて Fault Analyzer によって設定されます。これは、End Processingユーザー出口でオーバーライドできます。
10	(A)	CHAR	R/O	1	(予約済み)
11	(B)	CHAR	R/W	1	<b>SUPPRESS_FAULT_ENTRY</b>  リアルタイムの場合はヒストリー・ファイル障害項目を抑止し、障害項目リフレッシュの場合はヒストリー・ファイル障害項目を更新しません (Y/N)。
12	(C)	CHAR	R/O	15	(予約済み)
26	(1A)	CHAR	R/O	5	<b>MINUTES_SINCE_LAST_DUP</b>  重複する最後の障害が記録されてから経過した分数 (nnnnn)。ブランクの場合は、重複する障害がなかったことを示します。

表 20. EPC データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
 <b>注:</b> 分数がこのフィールドの制限を超えるときは必ず 99999 という最大値が使用されます。					
31	(1F)	CHAR	R/O	1	<b>ANALYSIS_SUCCESSFUL</b>  正常な分析 (Y/N)。  正常に分析が行われたかどうかの基準は、次のとおりです。 <ul style="list-style-type: none"> <li>• エラー・メッセージが発行されていない。</li> <li>• 障害点についてコードのソース行が識別される。</li> </ul>
32	(20)	CHAR	R/O	88	(予約済み)
120	(78)	CHAR	R/W	1	<b>SUPPRESS_DUMP</b>  ダンプの抑止 (Y/N)。  このフラグは、MVS™ システム・ダンプまたは CICS® トランザク ション・ダンプの抑止に影響します。詳しくは、 <a href="#">ダンプの抑止 ペー            ジ 33</a> を参照してください。
121	(79)	CHAR	R/O	63	(予約済み)

## LST - Compiler Listing Readユーザー出口パラメーター・リスト

表 21. LST データ域

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
0	(0)	CHAR	R/O	4	<b>VERSION</b>  パラメーター・リストのバージョン (現在は 0001)。
4	(4)	CHAR	R/O	8	<b>MODULE_NAME</b>  モジュール名。

表 21. LST データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
					この名前は LST.CSECT_NAME で識別される CSECT が含まれているロード・モジュールの名前です。
12	(C)	CHAR	R/O	8	<b>CSECT_NAME</b>  CSECT 名。  この名前は LST.PROGRAM_NAME で識別されるプログラムが含まれている CSECT の名前です。
20	(14)	CHAR	R/O	256	<b>EP_NAME</b>  エンタリー・ポイント名 (256 文字までで切り捨て)。
276	(114)	CHAR	R/O	10	<b>COMPILE_DATE</b>  フォーマット YYYY/MM/DD でのコンパイル日付。
286	(11E)	CHAR	R/O	8	<b>COMPILE_TIME</b>  HH:MM:SS 形式によるコンパイル時間。
294	(126)	CHAR	R/O	1	<b>LISTING_TYPE</b>  コンパイラ・リストまたはアセンブラー SYSADATA ファイル (L)、またはサイド・ファイル (S)。
295	(127)	CHAR	R/O	12	<b>LANGUAGE_TYPE</b>  言語タイプ:  <ul style="list-style-type: none"> <li>• アセンブラー</li> <li>• C/C++</li> <li>• COBOL</li> <li>• OS/VS COBOL</li> <li>• PL/I</li> <li>• Enterprise PL/I</li> </ul>
307	(133)	CHAR	R/O	4	<b>RECFM</b>

表 21. LST データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
レコード・フォーマット。					
311	(137)	CHAR	R/O	5	<b>LRECL</b>  論理レコード長 (nnnnn)。
316	(13C)	CHAR	R/W	5	<b>DATA_LENGTH</b>  可変長レコードのデータ長 (nnnnn)。このフィールドは、DATA_BUFFER に配置されるレコードの長さを示します。
321	(141)	CHAR	R/W	1	<b>DISREGARD_EXIT_LISTING</b>  出口により指定されるコンパイラ・リストまたはサイド・ファイルの無視 (Y/N)。このフィールドは Fault Analyzer により必ず「N」に初期化されます。「Y」が戻されると、Fault Analyzer は指定されたデータを無視して、通常の検索パスを介してリストまたはサイド・ファイルの検索を続けます。
322	(142)	CHAR	R/O	8	<b>PROGRAM_NAME</b>  プログラム名。プログラム名の長さが 8 文字を超える場合、このフィールドには最初の 8 文字のみが格納されます。この場合は、代わりに LONG_PROGRAM_NAME フィールドを使用します。
330	(14A)	CHAR	R/O	10	<b>PROGRAM_LENGTH</b>  プログラムの長さ (バイト数) (10 進数)。プログラム名の長さが 8 文字を超える場合、このフィールドには最初の 8 文字のみが格納されます。この場合は、代わりに LONG_PROGRAM_NAME フィールドを使用します。
340	(154)	CHAR	R/W	1	<b>DATA_BUFFER_DSN</b>  データ・バッファにデータ・セット名が含まれている (Y/N)。
341	(155)	CHAR	R/O	44	<b>LOAD_MODULE_DSN</b>

表 21. LST データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
					ロード・モジュールのデータ・セット名。  この名前は LST.MODULE_NAME で識別されるロード・モジュールがロードされたときに含まれていた データ・セットの名前です。
385	(181)	CHAR	R/O	5	(予約済み)
390	(186)	CHAR	R/W	8188	<b>DATA_BUFFER</b>
					データ・バッファ。  このフィールドの内容が大文字に変換されることはありません。この フィールドのヌル文字 ('X'00') による切り捨ては許可されていません。可変長レコードの場 合、DATA_LENGTH フィールドに長さを指定する必要があります。固定長レコードの場 合、長さは LRECL に一致しなければなりません。  DATA_BUFFER_DSN が「Y」に設定される場合、このフィールドには、LISTING_TYPE に適したコンパイラ・リストまたはサイド・ファイルが入ってい る、データ・セットの名前 (区分データ・セットである場合、小括弧に入ったメンバー名が続く) を含むことが予想されます。データ・セット名の追加要件につい ては、 <a href="#">コンパイラ・リスト読み取りユーザー出口 ページ 480</a> を参照し てください。
8578	(2182)	CHAR	R/O	32	<b>LONG_PROGRAM_NAME</b>
					長いプログラム名 (32 文字まで)。  このフィールドは、プログラム名の長さが 8 文字を超える場合にのみ初期化されます。それ以外の場合 は、PROGRAM_NAME フィールドを使用します。

## NFY - 通知ユーザー出口パラメーター・リスト

表 22. Nfy データ域

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
0	(0)	CHAR	R/O	4	<b>VERSION</b>  パラメーター・リストのバージョン (現在は 0002)。
4	(4)	CHAR	R/O	45	(予約済み)
49	(31)	CHAR	R/O	1024	<b>SYNOPSIS</b>  障害分析の概要。SYNOPSIS の各行は、改行文字 (X'15') で区切られます。  Fault Analyzer は、概要のサイズがこのフィールドに指定可能な最大サイズを超える 場合に使用される、バッファ・データ・フォーマットを許可しま す。詳しくは、 <a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a> を参照してください。このフィールドのフォーマット は、REXX 出口のユーザーに対し透過的です。
1073	(431)	CHAR	R/O	1	<b>NFYTYPE</b>  <b>C</b> 障害が作成されました  <b>R</b> リカバリー障害記録  <b>N</b> NoDup(Normal) 重複  <b>F</b> NoDup(CICSfast) または NoDup(ImageFast) 重複
1074	(432)	CHAR	R/O	8	<b>DUPCOUNT</b>  NFYTYPE が [F] に設定されているときは、この 30 秒の記録期間における新規重複の数。NFYTYPE



表 22. NFY データ域

(続く)

オフセット		タイプ	アクセ ス	長さ	名前と説明
10 進	16 進				
が「N」に設定されているときは、常に 1。それ以外の NFYTYPE の値には適用されません。					
1082	(43A)	CHAR	R/O	55	(予約済み)

## UFM - ユーザー出口パラメーター・リストのフォーマット

表 23. UFM データ域

オフセット		タイプ	アク セス	長さ	名前と説明
10 進	16 進				
0	(0)	CHAR	R/O	4	<b>VERSION</b>  パラメーター・リストのバージョン (現在は 0001)。
4	(4)	CHAR	R/W	100	<b>USEROPTIONTITLE</b>  Exits オプションを使用して実行されたすべてのフォーマット設定ユーザー出口からの出力のレポート・セクション見出し。これは、以前に呼び出されたいずれかのフォーマット設定ユーザー出口によって設定された見出しに初期化されます。バッチ・レポートの初期デフォルトは「USER」であり、対話式再分析レポートの場合は「User」です。
104	(68)	CHAR	R/O	91	(予約済み)
195	(C3)	CHAR	R/O	5	<b>NUM_EVENTS</b>  イベントの総数(10 進数)。
これ以降のフィールドには、単一イベント用のデータのみが設定されます。他のイベント用のデータを設定するには、IDIEventInfo を使用します。					
200	(C8)	CHAR	R/W	5	<b>EVENT_NO</b>  Current@ イベント番号 (nnnnn)。

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
205	(CD)	CHAR	R/O	5	<b>NEXT_EVENT_NO</b>  次の使用可能なイベント番号 (10 進数)。
210	(D2)	CHAR	R/O	5	<b>PREVIOUS_EVENT_NO</b>  前の使用可能なイベント番号 (10 進数)。
215	(D7)	CHAR	R/O	1	<b>POF</b>  障害点 (Y/N)。
216	(D8)	CHAR	R/O	30	<b>EVENT_TYPE</b>  「Abend S0C7」など、分析レポートの Event Summary セクションに示されるものと同じフォーマットのイベント・タイプです。  このフィールド用のデータのサイズがフィールドのサイズを超える場合、バッファ・データ・フォーマットが使用されます。詳しくは、 <a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a> を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。  言語 (JPN) が有効になっている場合は、このフィールドに指定されたイベント・タイプ記述が日本語に変換されます。
246	(F6)	CHAR	R/O	12	<b>MODULE_NAME</b>  モジュール名。  このフィールド用のデータのサイズがフィールドのサイズを超える場合、バッファ・データ・フォーマットが使用されます。詳しくは、 <a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a> を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。
258	(102)	CHAR	R/O	8	<b>MODULE_ADDRESS</b>

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
					モジュール・アドレス。
266	(10A)	CHAR	R/O	8	<b>MODULE_LENGTH</b> モジュール長 (16 進数)。
274	(112)	CHAR	R/O	12	<b>PROGRAM_NAME</b> プログラム名。  このフィールド用のデータのサイズがフィールドのサイズを超える場合、バッファ・データ・フォーマットが使用されます。詳しくは、 <a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a> を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。
286	(11E)	CHAR	R/O	8	<b>PROGRAM_ADDRESS</b> プログラム・アドレス。
294	(126)	CHAR	R/O	8	<b>PROGRAM_LENGTH</b> プログラム長 (16 進数)。
302	(12E)	CHAR	R/O	12	<b>EP_NAME</b> エントリー・ポイント名。  このフィールド用のデータのサイズがフィールドのサイズを超える場合、バッファ・データ・フォーマットが使用されます。詳しくは、 <a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a> を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。
314	(13A)	CHAR	R/O	8	<b>EP_ADDRESS</b> エントリー・ポイント・アドレス。
322	(142)	CHAR	R/O	64	<b>EVENT_LOCATION</b>

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
					<p>[L#31 P+3D4] など、分析レポートの Event Summary セクションに示されるものと同じフォーマットのイベントの位置です。</p> <p>このフィールド用のデータのサイズがフィールドのサイズを超える場合、バッファ・データ・フォーマットが使用されます。詳しくは、<a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a>を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。</p>
386	(182)	CHAR	R/O	44	<p><b>LOADED_FROM</b></p> <p>データ・セット名など、分析レポートの Event Summary セクションに示されるものと同じフォーマットで示された、モジュールのロード元に関する情報です。</p> <p>このフィールド用のデータのサイズがフィールドのサイズを超える場合、バッファ・データ・フォーマットが使用されます。詳しくは、<a href="#">非 REXX ユーザー出口のバッファ・データ・フォーマット ページ 636</a>を参照してください。このフィールドのフォーマットは、REXX 出口のユーザーに対し透過的です。</p>
430	(1AE)	CHAR	R/O	8	<p><b>INSTRUCTION_ADDRESS</b></p> <p>イベントの命令アドレス。</p>
438	(1B6)	CHAR	R/O	2	<p><b>AMODE</b></p> <p>イベント・アドレッシング・モード (24/31/64)。</p>
440	(1B8)	CHAR	R/O	16	<p><b>PSW</b></p> <p>イベントの PSW。</p>
456	(1C8)	CHAR	R/O	8	<p><b>GPREG0</b></p> <p>汎用レジスタ 0。</p>
464	(1D0)	CHAR	R/O	8	<p><b>GPREG1</b></p>

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10進	16進				
					汎用レジスタ 1。
472	(1D8)	CHAR	R/O	8	<b>GPREG2</b> 汎用レジスタ 2。
480	(1E0)	CHAR	R/O	8	<b>GPREG3</b> 汎用レジスタ 3。
488	(1E8)	CHAR	R/O	8	<b>GPREG4</b> 汎用レジスタ 4。
496	(1F0)	CHAR	R/O	8	<b>GPREG5</b> 汎用レジスタ 5。
504	(1F8)	CHAR	R/O	8	<b>GPREG6</b> 汎用レジスタ 6。
512	(200)	CHAR	R/O	8	<b>GPREG7</b> 汎用レジスタ 7。
520	(208)	CHAR	R/O	8	<b>GPREG8</b> 汎用レジスタ 8。
528	(210)	CHAR	R/O	8	<b>GPREG9</b> 汎用レジスタ 9。
536	(218)	CHAR	R/O	8	<b>GPREG10</b> 汎用レジスタ 10。
544	(220)	CHAR	R/O	8	<b>GPREG11</b> 汎用レジスタ 11。

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
552	(228)	CHAR	R/O	8	<b>GPREG12</b> 汎用レジスター 12。
560	(230)	CHAR	R/O	8	<b>GPREG13</b> 汎用レジスター 13。
568	(238)	CHAR	R/O	8	<b>GPREG14</b> 汎用レジスター 14。
576	(240)	CHAR	R/O	8	<b>GPREG15</b> 汎用レジスター 15。
584	(248)	CHAR	R/O	8	<b>AREG_DATA_ADDRESS</b> アドレス・レジスターを含むストレージ域の 16 進フォーマットでのアドレス (AR0 から AR15)。
592	(250)	CHAR	R/O	122	(予約済み)
714	(2CA)	CHAR	R/O	16	<b>BEAR</b> Breaking Event Address Register。
730	(2DA)	CHAR	R/W	5	<b>DATA_LENGTH</b> データ長 (nnnnn)。このフィールドは、UFM.DATA_BUFFER に配置されるレコードの長さを示します。
735	(2DF)	CHAR	R/W	1024	<b>DATA_BUFFER</b> データ・バッファ。  このフィールドの内容が大文字に変換されることはありません。このフィールドのヌル文字 (X'00') による切り捨ては許可されていません。UFM.DATA_LENGTH フィールドで長さが提供される必要があります。

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
<p> <b>注:</b> このフィールドは、ENV.WRITE_ROUTINE_EP プログラムを使用してロード・モジュール・ユーザー出口から Fault Analyzer にレコードを戻す際に、レコード・バッファーとして使用されます。このバッファーの使用方法については、<a href="#">フォーマット・ユーザー出口 ページ 491</a>を参照してください。REXX ユーザー出口では、Fault Analyzer コマンドを使用して IDIWRITE に直接データを戻すことができるため、このフィールドを使用する必要はありません。</p>					
1759	(6DF)	CHAR	R/O	1	(予約済み)
1760	(6E0)	CHAR	R/O	16	<b>FPREG0</b> 浮動小数点レジスター 0。
1776	(6F0)	CHAR	R/O	16	<b>FPREG1</b> 浮動小数点レジスター 1。
1792	(700)	CHAR	R/O	16	<b>FPREG2</b> 浮動小数点レジスター 2。
1808	(710)	CHAR	R/O	16	<b>FPREG3</b> 浮動小数点レジスター 3。
1824	(720)	CHAR	R/O	16	<b>FPREG4</b> 浮動小数点レジスター 4。
1840	(730)	CHAR	R/O	16	<b>FPREG5</b> 浮動小数点レジスター 5。
1856	(740)	CHAR	R/O	16	<b>FPREG6</b> 浮動小数点レジスター 6。

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
1872	(750)	CHAR	R/O	16	<b>FPREG7</b> 浮動小数点レジスター 7。
1888	(760)	CHAR	R/O	16	<b>FPREG8</b> 浮動小数点レジスター 8。
1904	(770)	CHAR	R/O	16	<b>FPREG9</b> 浮動小数点レジスター 9。
1920	(780)	CHAR	R/O	16	<b>FPREG10</b> 浮動小数点レジスター 10。
1936	(790)	CHAR	R/O	16	<b>FPREG11</b> 浮動小数点レジスター 11。
1952	(7A0)	CHAR	R/O	16	<b>FPREG12</b> 浮動小数点レジスター 12。
1968	(7B0)	CHAR	R/O	16	<b>FPREG13</b> 浮動小数点レジスター 13。
1984	(7C0)	CHAR	R/O	16	<b>FPREG14</b> 浮動小数点レジスター 14。
2000	(7D0)	CHAR	R/O	16	<b>FPREG15</b> 浮動小数点レジスター 15。
2016	(7E0)	CHAR	R/O	8	<b>FPCR</b> 浮動小数点制御レジスター。
2024	(7E8)	CHAR	R/O	1	<b>GPREGS_64BIT</b>



表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
64 ビット汎用レジスタ使用可能 (Y/N)。					
2025	(7E9)	CHAR	R/O	16	<b>GPREG0_64BIT</b> 汎用レジスタ 0 (64 ビット)。
2041	(7F9)	CHAR	R/O	16	<b>GPREG1_64BIT</b> 汎用レジスタ 1 (64 ビット)。
2057	(809)	CHAR	R/O	16	<b>GPREG2_64BIT</b> 汎用レジスタ 2 (64 ビット)。
2073	(819)	CHAR	R/O	16	<b>GPREG3_64BIT</b> 汎用レジスタ 3 (64 ビット)。
2089	(829)	CHAR	R/O	16	<b>GPREG4_64BIT</b> 汎用レジスタ 4 (64 ビット)。
2105	(839)	CHAR	R/O	16	<b>GPREG5_64BIT</b> 汎用レジスタ 5 (64 ビット)。
2121	(849)	CHAR	R/O	16	<b>GPREG6_64BIT</b> 汎用レジスタ 6 (64 ビット)。
2137	(859)	CHAR	R/O	16	<b>GPREG7_64BIT</b> 汎用レジスタ 7 (64 ビット)。
2153	(869)	CHAR	R/O	16	<b>GPREG8_64BIT</b> 汎用レジスタ 8 (64 ビット)。
2169	(879)	CHAR	R/O	16	<b>GPREG9_64BIT</b> 汎用レジスタ 9 (64 ビット)。

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
2185	(889)	CHAR	R/O	16	<b>GPREG10_64BIT</b> 汎用レジスター 10 (64 ビット)。
2201	(899)	CHAR	R/O	16	<b>GPREG11_64BIT</b> 汎用レジスター 11 (64 ビット)。
2217	(8A9)	CHAR	R/O	16	<b>GPREG12_64BIT</b> 汎用レジスター 12 (64 ビット)。
2233	(8B9)	CHAR	R/O	16	<b>GPREG13_64BIT</b> 汎用レジスター 13 (64 ビット)。
2249	(8C9)	CHAR	R/O	16	<b>GPREG14_64BIT</b> 汎用レジスター 14 (64 ビット)。
2265	(8D9)	CHAR	R/O	16	<b>GPREG15_64BIT</b> 汎用レジスター 15 (64 ビット)。
2281	(8E9)	CHAR	R/O	32	<b>VFREG0</b> ベクトル機能レジスター 0。
2313	(909)	CHAR	R/O	32	<b>VFREG1</b> ベクトル機能レジスター 1。
2345	(929)	CHAR	R/O	32	<b>VFREG2</b> ベクトル機能レジスター 2。
2377	(949)	CHAR	R/O	32	<b>VFREG3</b> ベクトル機能レジスター 3。
2409	(969)	CHAR	R/O	32	<b>VFREG4</b>

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
					ベクトル機能レジスタ 4。
2441	(989)	CHAR	R/O	32	<b>VFREG5</b> ベクトル機能レジスタ 5。
2473	(9A9)	CHAR	R/O	32	<b>VFREG6</b> ベクトル機能レジスタ 6。
2505	(9C9)	CHAR	R/O	32	<b>VFREG7</b> ベクトル機能レジスタ 7。
2537	(9E9)	CHAR	R/O	32	<b>VFREG8</b> ベクトル機能レジスタ 8。
2569	(A09)	CHAR	R/O	32	<b>VFREG9</b> ベクトル機能レジスタ 9。
2601	(A29)	CHAR	R/O	32	<b>VFREG10</b> ベクトル機能レジスタ 10。
2633	(A49)	CHAR	R/O	32	<b>VFREG11</b> ベクトル機能レジスタ 11。
2665	(A69)	CHAR	R/O	32	<b>VFREG12</b> ベクトル機能レジスタ 12。
2697	(A89)	CHAR	R/O	32	<b>VFREG13</b> ベクトル機能レジスタ 13。
2729	(AA9)	CHAR	R/O	32	<b>VFREG14</b> ベクトル機能レジスタ 14。

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
2761	(AC9)	CHAR	R/O	32	<b>VFREG15</b> ベクトル機能レジスター 15。
2793	(AE9)	CHAR	R/O	32	<b>VFREG16</b> ベクトル機能レジスター 16。
2825	(B09)	CHAR	R/O	32	<b>VFREG17</b> ベクトル機能レジスター 17。
2857	(B29)	CHAR	R/O	32	<b>VFREG18</b> ベクトル機能レジスター 18。
2889	(B49)	CHAR	R/O	32	<b>VFREG19</b> ベクトル機能レジスター 19。
2921	(B69)	CHAR	R/O	32	<b>VFREG20</b> ベクトル機能レジスター 20。
2953	(B89)	CHAR	R/O	32	<b>VFREG21</b> ベクトル機能レジスター 21。
2985	(BA9)	CHAR	R/O	32	<b>VFREG22</b> ベクトル機能レジスター 22。
3017	(BC9)	CHAR	R/O	32	<b>VFREG23</b> ベクトル機能レジスター 23。
3049	(BE9)	CHAR	R/O	32	<b>VFREG24</b> ベクトル機能レジスター 24。
3081	(C09)	CHAR	R/O	32	<b>VFREG25</b>

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
					ベクトル機能レジスタ 25。
3113	(C29)	CHAR	R/O	32	<b>VFREG26</b> ベクトル機能レジスタ 26。
3145	(C49)	CHAR	R/O	32	<b>VFREG27</b> ベクトル機能レジスタ 27。
3177	(C69)	CHAR	R/O	32	<b>VFREG28</b> ベクトル機能レジスタ 28。
3209	(C89)	CHAR	R/O	32	<b>VFREG29</b> ベクトル機能レジスタ 29。
3241	(CA9)	CHAR	R/O	32	<b>VFREG30</b> ベクトル機能レジスタ 30。
3273	(CC9)	CHAR	R/O	32	<b>VFREG31</b> ベクトル機能レジスタ 31。
以下の GPREGn_VALID フィールドは、GPREGn と汎用レジスタ値 GPREGn_64BIT 両方に適用されます。つまり、レジスタ n の値が「N」の場合、GPREGn も GPREGn_64BIT も無効です。					
3305	(CE9)	CHAR	R/O	1	<b>GPREG0_VALID</b> 汎用レジスタ 0 が有効 (Y/N)。
3306	(CEA)	CHAR	R/O	1	<b>GPREG1_VALID</b> 汎用レジスタ 1 が有効 (Y/N)。
3307	(CEB)	CHAR	R/O	1	<b>GPREG2_VALID</b> 汎用レジスタ 2 が有効 (Y/N)。

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
3308	(CEC)	CHAR	R/O	1	<b>GPREG3_VALID</b> 汎用レジスタ 3 が有効 (Y/N)。
3309	(CED)	CHAR	R/O	1	<b>GPREG4_VALID</b> 汎用レジスタ 4 が有効 (Y/N)。
3310	(CEE)	CHAR	R/O	1	<b>GPREG5_VALID</b> 汎用レジスタ 5 が有効 (Y/N)。
3311	(CEF)	CHAR	R/O	1	<b>GPREG6_VALID</b> 汎用レジスタ 6 が有効 (Y/N)。
3312	(CF0)	CHAR	R/O	1	<b>GPREG7_VALID</b> 汎用レジスタ 7 が有効 (Y/N)。
3313	(CF1)	CHAR	R/O	1	<b>GPREG8_VALID</b> 汎用レジスタ 8 が有効 (Y/N)。
3314	(CF2)	CHAR	R/O	1	<b>GPREG9_VALID</b> 汎用レジスタ 9 が有効 (Y/N)。
3315	(CF3)	CHAR	R/O	1	<b>GPREG10_VALID</b> 汎用レジスタ 10 が有効 (Y/N)。
3316	(CF4)	CHAR	R/O	1	<b>GPREG11_VALID</b> 汎用レジスタ 11 が有効 (Y/N)。
3317	(CF5)	CHAR	R/O	1	<b>GPREG12_VALID</b> 汎用レジスタ 12 が有効 (Y/N)。
3318	(CF6)	CHAR	R/O	1	<b>GPREG13_VALID</b>

表 23. UFM データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10進	16進				
					汎用レジスタ 13 が有効 (Y/N)。
3319	(CF7)	CHAR	R/O	1	<b>GPREG14_VALID</b> 汎用レジスタ 14 が有効 (Y/N)。
3320	(CF8)	CHAR	R/O	1	<b>GPREG15_VALID</b> 汎用レジスタ 15 が有効 (Y/N)。

## UTL - IDIUTIL バッチ・ユーティリティー・ユーザー出口パラメーター・リスト

表 24. UTL データ域

オフセット		タイプ	アクセス	長さ	名前と説明
10進	16進				
0	(0)	CHAR	R/O	4	<b>VERSION</b> パラメーター・リストのバージョン (現在は 0002)。
4	(4)	CHAR	R/O	44	(予約済み)
48	(30)	CHAR	R/W	1	<b>PERFORM_ACTION</b> この障害項目に関するユーティリティー・アクションの実行 (Y/N)。
49	(31)	CHAR	R/W	100	<b>IMPORT_DUMP_DSN</b> この値は、IDIROBOT exec を使用して作成されたステージング・データ・セットから単一障害項目をインポートするときのみ適用されます。  インポート対象の障害項目が、関連ダンプ・データ・セットとともにエクスポートされたものである場合、関連付けられた元のダンプ・データ・セット名が ENV.ASSOCIATED_DUMP_DSN

表 24. UTL データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
					<p>で指定されます。その後、IDIUTIL ユーザー出口は次の方法で応答できます。</p> <ul style="list-style-type: none"> <li>• UTL.IMPORT_DUMP_DSN フィールドを「NULLFILE」に設定することで、関連するダンプ・データ・セットを無視できます。</li> <li>• UTL.IMPORT_DUMP_DSN を ENV.ASSOCIATED_DUMP_DSN で指定された元のダンプ・データ・セット名に初期化できます。❶</li> <li>• UTL.IMPORT_DUMP_DSN を別のダンプ・データ・セット名に初期化できます。❶</li> <li>• UTL.IMPORT_DUMP_DSN フィールドを空白のままにしておくことができます。この場合、IDIOPTLM 構成オプション・モジュールの RFRDSN、SDUMPDSN または XDUMPDSN オプションに指定した適切なデータ・セット名が使用されます。</li> </ul> <p>UTL.IMPORT_DUMP_DSN で指定された名前は、インポートされた障害項目に割り振られて関連付けられるダンプ・データ・セットに使用されます。</p>
83	(131)	CHAR	R/O	1	<p><b>DUP_TYPE</b></p> <p>このフィールドは、ListHFDup 出口を使用している場合にのみ適用できます。</p> <p>考えられる値は以下のとおりです。</p> <p><b>空白</b> 基本障害項目</p> <p><b>N</b> 通常重複</p> <p><b>F</b> CICSfast 重複</p>



表 24. UTL データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
Imagefase 重複					



注:

①

データ・セット名は、障害項目がインポートされるシステムおよび IDIUTIL が実行されるユーザー ID に適したものである必要があります。

## XPL - Message and Abend Code Explanationユーザー出口パラメーター・リスト

表 25. XPL データ域

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
0	(0)	CHAR	R/O	4	<b>VERSION</b> パラメーター・リストのバージョン (現在は 0001)。
4	(4)	CHAR	R/O	125	<b>MESSAGE_TEXT1</b> 単一行 WTO または最初の複数行 WTO メッセージ・テキスト。
129	(81)	CHAR	R/O	70	<b>MESSAGE_TEXT2</b> 複数行 WTO メッセージ・テキスト行 2。
199	(C7)	CHAR	R/O	70	<b>MESSAGE_TEXT3</b> 複数行 WTO メッセージ・テキスト行 3。
269	(10D)	CHAR	R/O	70	<b>MESSAGE_TEXT4</b> 複数行 WTO メッセージ・テキスト行 4。

表 25. XPL データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
339	(153)	CHAR	R/O	70	<b>MESSAGE_TEXT5</b> 複数行 WTO メッセージ・テキスト行 5。
409	(199)	CHAR	R/O	70	<b>MESSAGE_TEXT6</b> 複数行 WTO メッセージ・テキスト行 6。
479	(1DF)	CHAR	R/O	70	<b>MESSAGE_TEXT7</b> 複数行 WTO メッセージ・テキスト行 7。
549	(225)	CHAR	R/O	70	<b>MESSAGE_TEXT8</b> 複数行 WTO メッセージ・テキスト行 8。
619	(26B)	CHAR	R/O	70	<b>MESSAGE_TEXT9</b> 複数行 WTO メッセージ・テキスト行 9。
689	(2B1)	CHAR	R/O	70	<b>MESSAGE_TEXT10</b> 複数行 WTO メッセージ・テキスト行 10。
759	(2F7)	CHAR	R/O	4	(予約済み)
763	(2FB)	CHAR	R/O	8	<b>ABEND_REASON_CODE</b> ABEND 理由コード。
771	(303)	CHAR	R/O	8	<b>ABEND_MODULE_NAME</b> 異常終了モジュール名。
779	(30B)	CHAR	R/O	1	<b>ABEND_TYPE</b> ABEND タイプ。

**C**

CICS® トランザクション異常終了

表 25. XPL データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
					<b>D</b> CICS® ダンプ・コード
					<b>S</b> システム
					<b>U</b> ユーザー
780	(30C)	CHAR	R/W	5	<b>DATA_LENGTH</b>  データ長 (nnnnn)。このフィールドは、DATA_BUFFER に配置されるレコードの長さを示します。
785	(311)	CHAR	R/W	256	<b>DATA_BUFFER</b>  データ・バッファ。  このフィールドの内容が大文字に変換されることはありません。このフィールドのヌル文字 (X'00') による切り捨ては許可されていません。IDIWRITE コマンドで REXX EXEC 変数名を使用しない場合は、DATA_LENGTH フィールドに長さを指定する必要があります。
					 <b>注:</b> このフィールドは、Fault Analyzer (REXX) または ENV.WRITE_ROUTINE_EP プログラム (非 REXX) を使用して、ユーザーの終了時に IDIWRITE にレコードを戻す際に、レコード・バッファとして使用されます。詳細については、 <a href="#">メッセージおよび異常終了コード説明 ユーザー出口 ページ 485</a> を参照してください。
1041	(411)	CHAR	R/O	1	<b>EXPLANATION_AVAILABLE</b>  Fault Analyzer がメッセージまたは異常終了コードの説明を示すことができたかどうかを示すフラグ (Y/N)。
1042	(412)	CHAR	R/O	6	<b>ABEND_CODE</b>

表 25. XPL データ域

(続く)

オフセット		タイプ	アクセス	長さ	名前と説明
10 進	16 進				
異常終了コード。					
1048	(418)	CHAR	R/O	58	(予約済み)

## 第 34 章. 戻りコード

このトピックでは、Fault Analyzer ユーティリティーによって発行される戻りコードについて説明します。

### バッチ再分析 (IDIDA)

バッチ障害再分析を実行するときに受け取る可能性がある戻りコードは、次のとおりです。

**RC**

**意味**

**0**

- 1 つ以上の通知メッセージが発行された可能性があります (メッセージ番号にサフィックス「I」が付きます)。

**2 または 4**

- 1 つ以上の警告メッセージが発行されました (メッセージ番号にサフィックス「W」が付きます)。
- 通知メッセージも発行された可能性があります。

**8**

- 1 つ以上のエラー・メッセージが発行されました (メッセージ番号にサフィックス「E」が付きます)。
- 通知および警告メッセージも発行された可能性があります。

**12**

- 1 つ以上の重大メッセージが発行されました (メッセージ番号にサフィックス「S」が付きます)。
- 通知、警告およびエラー・メッセージも発行された可能性があります。

### IDIUTIL バッチ・ユーティリティー

以下の戻りコードが、IDIUTIL バッチ・ユーティリティーによって出されます。

**RC**

**意味**

**0**

正常終了。

**4**

1 つ以上の警告メッセージが SYSPRINT ddname に書き込まれます。

**8**

1 つ以上のエラー・メッセージが SYSPRINT ddname に書き込まれます。

## **IDILANGX**

IDILANGX によって発行される戻りコードについては、「*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*」を参照してください。

## 第 35 章. メッセージ

この章では、Fault Analyzer および関連ユーティリティーによって発行されるメッセージについて説明します。

### IDILANGX メッセージ

前に IDISF\* が付いているメッセージは、Fault Analyzer が内部で使用する際や、サイド・ファイルを作成する際にユーザーが起動する IDILANGX プログラムにより発行されます。それらについては、“*IBM Application Delivery Foundation for z/OS Common Components: Customization Guide and User Guide*”の「IPVLANGX メッセージ」に記載されています。

### Fault Analyzer メッセージ

Fault Analyzer により、以下のメッセージが発行されます。

---

#### IDI0001I

Fault Analyzer *version-info* invoked by *exit-name* using *config-member*

#### 説明

Fault Analyzer がリアルタイム分析で呼び出されました。メッセージ・テキスト内の各項目の意味は以下のとおりです。

- *version-info* は、使用する Fault Analyzer のバージョンと現在の保守レベルに関する情報を示します。
- *exit-name* は、Fault Analyzer を呼び出した出口を示します。
- *config-member* は、parmlib 構成オプションを含むデータ・セットとメンバーの名前です。ただし、構成メンバーが見つからなかった場合は、デフォルト・オプションの使用を示します。

**システムアクション** 通常の処理は継続されます。

**ユーザーの応答** なし

---

#### IDI0002I

*analysis-summary*

#### 説明

障害分析が完了しました。メッセージ・テキスト、*analysis-summary* に問題の要約が示されます。分析サマリーは通常、以下のフォーマットで表示されます。

```
[point-of-failure:] symptom
```

各部の意味は以下のとおりです。

#### **point-of-failure**

ユーザー・アプリケーション内のエラーの発生点、またはエラーの発生前のユーザー・アプリケーションの制御点。使用可能であれば、モジュール名、プログラムまたは CSECT 名、オフセット、およびソース行番号またはコンパイラー・リストのステートメント番号が示されます。

#### **symptom**

発生したエラーのタイプ (例: 初期異常終了コード)。

**システムアクション** 通常の処理は継続されます。

**ユーザーの応答** なし**IDI0003I**

Fault ID *faultid* assigned in history file *history-file-name*

**説明** Fault Analyzer がリアルタイム分析を完了し、ヒストリー・ファイル *faultid* 内の障害 ID *history-file-name* をこの異常終了に割り当てました。

**システムアクション** 処理が終了しました。

**ユーザーの応答** なし**IDI0004S**

入力ダンプ・データ・セット *data-set-name* が、次に示す原因のために、オープンできませんでした。 *reason*

**説明** *data-set-name* で識別されるダンプ・データ・セットが、*reason* に示された原因のために、再分析用にオープンできませんでした。

**システムアクション** 処理は終了します。

**ユーザーの応答** データ・セット名を訂正して、ジョブを再実行依頼します。

**IDI0005S**

*module-name:line-number* Storage allocation for *dec-count* (*X'hex-count'*) bytes failed - processing terminated

**説明** ストレージ不足条件が発生しています。

**システムアクション** 処理は終了します。

**ユーザーの応答** より大きな領域サイズを指定して、ジョブを再実行依頼します。詳しくは、[推奨されるストレージ ページ 297](#) を参照してください。TSO を使用している場合は、ログオン・パネルで、より大きい領域サイズを指定します。Fault Analyzer は、このメッセージを出す前に、使用可能な 31 ビット・ストレージを計画的にすべて使用しないようにします。すべてのストレージが使い果たされると、MVS™ 領域などの非終端アドレス・スペースの場合に重大な結果を引き起こす可能性のある 24 ビット・ストレージを、CICS® が代わりに使用することがあるためです。

**IDI0006E**

次に示された理由により、*context* データ・セット *data-set-name* のオープンに失敗しました。 *reason*

**説明** *data-set-name* で識別されるデータ・セットが、*reason* に示された原因のために、オープンできませんでした。データ・セットのオープンが試みられたコンテキストは、*context* で提供されます。このコンテキストは、関連付けられた ddname であるか、またはそれ以外の、関与するデータ・セットのタイプの記述であると考えられます。

*reason* が

```
DYNALLOC error=error-code info=info-code
```

というフォームの場合、エラー・コードと情報コードの詳細は「z/OS®: MVS™ Programming: Authorized Assembler Services Guide」を参照してください。

いくつかのより一般的な DYNALLOC 情報コードの理由は次のとおりです。

**情報コード****理由**



210

データ・セットは別のジョブに割り当てられています。

1708

データ・セットが見つからない。

**システムアクション** このデータ・セットを使用せずに処理の継続が試行されます。

**ユーザーの応答** オープンが失敗した原因を判別し、ジョブを再実行依頼します。

---

IDI0007S

GETMAIN failed *resource-name* rc=*return-code*

**説明** ストレージ不足条件が発生しています。

**システムアクション** 処理は終了します。

**ユーザーの応答** より大きい領域サイズを指定して、ジョブを再実行依頼します。

---

IDI0008E

CSVINFO error, rc=*return-code*

**説明** CSVINFO サービスを使用中にエラーが発生しました。

**システムアクション** 処理は終了します。

**ユーザーの応答** 内部エラーである可能性があります。IBM サポートに連絡してください。

---

IDI0009E

IEWBUFF error

**説明** バインドを使用中にエラーが発生しました。

**システムアクション** 処理は終了します。

**ユーザーの応答** 内部エラーである可能性があります。IBM サポートに連絡してください。

---

IDI0010E

IEWBIND error *function module-name* rc=*reason-code*

**説明**

MVS™ バインド・プログラムの呼び出しが失敗しました。メッセージ・テキストで、*function* は、要求される関数のタイプを識別します。*module-name* は、バインドが試行されたモジュールであり、*reason-code* は、バインドにより戻された理由コードです。



**注:** *reason-code* の値は、バインド・プログラム API の理由コードです。バインド・プログラム API の理由コードは、「z/OS®: MVS™ プログラム管理: 拡張機能」に記載されています。

**システムアクション** 処理を続行しますが、分析が完全でない可能性があります。

**ユーザーの応答** 内部エラーである可能性があります。IBM サポートに連絡してください。

---

#### IDI0011S

Fault Analyzer 分析で異常終了 *abend-code*が発生しました。

**説明** Fault Analyzer で異常終了が発生しました。

**システムアクション** 処理は終了します。

**ユーザーの応答** これは内部エラーです。IBM サポートに連絡してください。

---

#### IDI0012S

Abend *abend-code* occurred in abend exit processing,

**説明** Fault Analyzer で異常終了が発生しました。このメッセージの後にメッセージ [IDI0013S ページ 698](#) が続きます。

**システムアクション** 処理は終了します。

**ユーザーの応答** これは内部エラーです。IBM サポートに連絡してください。後で分析するため、Fault Analyzer により書き込まれた SVC ダンプを保持しておいてください。

---

#### IDI0013S

R15=*r15-value* PSW=*program-status-word* DCAPSUB=*base-reg-value*

**説明** このメッセージはメッセージ [IDI0012S ページ 698](#) の後に続きます。

**システムアクション** メッセージ [IDI0012S ページ 698](#) を参照してください。

**ユーザーの応答** メッセージ [IDI0012S ページ 698](#) を参照してください。

---

#### IDI0014E

MTRACE error calling IEEMB879 for buffer accumulation, rc=*rc*.

**説明** MTRACE 処理中にエラーが発生しました。

**システムアクション** MTRACE の処理が終了し、障害ヒストリー・ファイル内で、コンソール・メッセージに関する情報が欠落している可能性があります。

**ユーザーの応答** 内部エラーである可能性があります。IBM サポートに連絡してください。

---

#### IDI0015W

表示するメッセージ *message-number* は見つかりませんでした

**説明** *message-number* で示されたメッセージの発行を試行しましたが、失敗しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** これは内部エラーです。IBM サポートに連絡してください。

**IDI0016E**

Abend *abend-code* during fault history file OPEN/READ processing

**説明** リアルタイム処理中に、ヒストリー・ファイルをオープンまたは読み取ろうとした際に異常終了しました。

**システムアクション** 処理は継続されますが、障害項目は作成されませんし、重複カウントは更新されません。

**ユーザーの応答** 異常終了の理由を判別してください。

**IDI0018W**

Parmlib member read error: *reason*

**説明** IDICNFxx parmlib 構成メンバーのオープンまたは読み取り試行中に問題が発生しました。エラーの原因が *reason* に示されます。

**システムアクション** IDICNFxx parmlib メンバーを使用せずに処理を継続します。

**ユーザーの応答** IDICNFxx メンバーが論理 parmlib 連結、または IDIOPTLM 構成オプションのモジュール CNFDSN 設定を介して指定された別の parmlib データ・セットに存在していることを確認してください。詳しくは、[Fault Analyzer 構成オプション・モジュールを使用した IDIOPTLM のカスタマイズ ページ 338](#)を参照してください。

**IDI0019W**

*options-source* syntax error on line *line-number* column *column-number*: *reason*

**説明**

*options-source* に指定されたオプションの処理中に構文エラーが検出されました。ここで、*options-source* は以下のいずれかです。

**Analysis Control user exit**

このオプションは分析制御ユーザー出口により提供されました。

**環境変数 \_IDI\_OPTS**

このオプションは \_IDI\_OPTS 環境変数を介して提供されました。

**Options line for interactive reanalysis**

このオプションは、アクション・バーの「オプション」プルダウン・メニューから「対話式再分析オプション...」を選択すると表示される「対話式再分析オプション」画面の「対話式再分析のオプション行」フィールドに指定されました。

**PARM フィールド**

このオプションは、以下のどちらかの場所に指定されました。

- バッチ再分析ジョブにおける PGM=IDIDA のための EXEC カードの PARM フィールド。
- Batch Reanalysis Options 画面の **Options line for batch reanalysis** フィールド。このフィールドは、アクション・バーの「オプション」プルダウン・メニューから **Batch Reanalysis Options...** を選択すると表示されます。

**Parmlib config member**

このオプションは、IDICNFxx parmlib メンバーに指定されたか、または IDICNFUM ユーザー・オプション・モジュールによって識別されるデータ・セットおよびメンバーに指定されました。

### User options file

このオプションは IDIOPTS DDname を通して指定されました。

**システムアクション** 処理を続行します。

**ユーザーの応答** エラーを訂正して、ジョブを再実行依頼します。

---

### IDI0020W

*options-source* contained invalid option *option*

**説明** *options-source* に指定されたオプションの処理中に無効なオプションが検出されました。 *options-source* の可能な値については、メッセージ [IDI0019W ページ 699](#) を参照してください。

**システムアクション** オプション *option* が無視され、処理を続行します。

**ユーザーの応答** エラーを訂正して、ジョブを再実行依頼します。

---

### IDI0021W

割り振りが失敗しました: DD=*ddname* DSN=*data-set-name* RC=*return-code* Error=*error-code* Info=*info-code*

**説明** 動的データ・セット割り振りが失敗しました。

**システムアクション** *data-set-name* に示されるデータ・セットを使用せずに処理を続行します。

**ユーザーの応答** 「z/OS®: MVS™ Programming: Authorized Assembler Services Guide」で、戻りコード、エラー・コード、および情報コードについて参照してください。

---

### IDI0022W

Suboption(s) missing for *options-source* option *option*

**説明** *options-source* に指定されたオプションの処理中に、必須サブオプションが欠落したオプションが検出されました。 *options-source* の可能な値については、メッセージ [IDI0019W ページ 699](#) を参照してください。

**システムアクション** オプション *option* のデフォルト値を使用して処理を続行します。

**ユーザーの応答** エラーを訂正して、ジョブを再実行依頼します。

---

### IDI0023W

Suboption(s) ignored for *options-source* option *option*

**説明** サブオプションが許可されないときに、*options-source* のオプションで1つまたは複数のサブオプションを指定しました。 *options-source* の可能な値については、メッセージ [IDI0019W ページ 699](#) を参照してください。

**システムアクション** オプション *option* のデフォルト値を使用して処理を続行します。

**ユーザーの応答** エラーを訂正して、ジョブを再実行依頼します。

---

### IDI0024W

Concatenation failed for DSN=*data-set-name* to DD=*ddname* RC=*return-code* Error=*error-code*Info=*info-code*

**説明** 動的データ・セット連結が失敗しました。

**システムアクション** *data-set-name* に示されるデータ・セットを使用せずに処理を続行します。

**ユーザーの応答** 「z/OS®: MVS™ Programming: Authorized Assembler Services Guide」で、戻りコード、エラー・コード、および情報コードについて参照してください。

---

#### IDI0025W

Suboption *sub-opt* invalid for *options-source* option *option*

**説明** *options-source* のオプションが、1 つまたは複数の無効なサブオプションを指定しました。 *options-source* の可能な値については、メッセージ [IDI0019W ページ 699](#) を参照してください。

**システムアクション** オプション *option* のデフォルト値を使用して処理を続行します。

**ユーザーの応答** エラーを訂正して、ジョブを再実行依頼します。

---

#### IDI0026W

Information retrieval failed for DD=*ddname* RC=*return-code* Error=*error-code* Info=*info-code*

**説明** 動的割り振り情報の検索が失敗しました。

**システムアクション** *data-set-name* に示されるデータ・セットを使用せずに処理を続行します。

**ユーザーの応答** 「z/OS®: MVS™ Programming: Authorized Assembler Services Guide」で、戻りコード、エラー・コード、および情報コードについて参照してください。

---

#### IDI0027E

Fetch/load failed for module *module-name*

**説明** モジュール *module-name* の取り出しまたはロードを試行しましたが、失敗しました。 *module-name* が IPVLANGX または IPVLANGP (あるいはそれらの別名 IDILANGX または IDILANGP) である場合、これらのモジュールは ADFz Common Components によって提供され、LINKLIST または LPA を介して使用できるようにする必要があります。詳細については、[Fault Analyzer モジュールの有効化 ページ 307](#) を参照してください。

**システムアクション** *module-name* が IPVLANGX または IPVLANGP (あるいはそれらの別名 IDILANGX または IDILANGP) ではない場合、もしくは IBM Fault Analyzer プラグイン (Eclipse 用) の使用中にこのメッセージが発せられていなかった場合、モジュール *module-name* を使用せずに処理を続行します。ただし、サブジェクト・ロード・モジュールの機能によっては、処理が完了しない場合があります。 *module-name* が IPVLANGX または IPVLANGP (あるいはそれらの別名 IDILANGX または IDILANGP) であり、IBM Fault Analyzer プラグイン (Eclipse 用) の使用中にこのメッセージが発せられた場合、ADFz サーバーの Fault Analyzer プロセスが終了し、最終的に IBM Fault Analyzer プラグイン (Eclipse 用) を使用できなくなります。

**ユーザーの応答** システム・プログラマーに連絡してください。

---

#### IDI0028W

Error reading user options file

**説明** DDname IDIOPTS によりユーザー・オプション・ファイルを読み取る際にエラーが発生しました。

**システムアクション** ユーザー・オプション・ファイルを使用せずに処理を続行します。

**ユーザーの応答** 内部エラーである可能性があります。IBM® サポートに連絡してください。

---

#### IDI0029W

*options-source options syntax error at offset offset. reason*

##### 説明

*options-source* に指定されたオプションの処理中に構文エラーが検出されました。ここで、*options-source* は以下のいずれかです。

##### PARM フィールド

このフィールドは、プログラム IDIDA を実行するジョブで指定される JCL EXEC ステートメントの PARM フィールドです。

##### 環境変数 \_IDI\_OPTS

この環境変数は、異常終了アプリケーションによって設定されます。

**システムアクション** 処理を続行します。

**ユーザーの応答** エラーを訂正して、ジョブを再実行依頼します。

---

#### IDI0030W

Allocation to SYSOUT=(*class,form-name*) failed: DD=*ddname* RC=*return-code* Error=*error-code* Info=*info-code*

**説明** JES スプール・データ・セットの動的割り振りが失敗しました。

**システムアクション** *ddname* に示される DDname を使用せずに処理を続行します。

**ユーザーの応答** [z/OS®: *MVS™ Programming: Authorized Assembler Services Guide*] で、戻りコード、エラー・コード、および情報コードについて参照してください。

---

#### IDI0031W

DD= *ddname* のオープンに失敗しました: *reason*

**説明** DDname *ddname* のオープンを試行しましたが、失敗しました。

**システムアクション** *ddname* に示される DDname を使用せずに処理を続行します。

**ユーザーの応答** 考えられ得る説明が *reason* に示されます。

---

#### IDI0032W

I/O error writing report: *reason*

**説明** 分析レポートの書き込み中に入出力エラーが発生しました。

**システムアクション** 処理を続行しますが、分析レポートの情報が欠落する場合があります。

**ユーザーの応答** 考えられ得る説明が *reason* に示されます。

**IDI0033E**

*history-file-dsn* への書き込み中に I/O エラーが発生しました: システム異常終了 *abend-code* *reason-code* *reason-text*

**説明** *history-file-dsn* に示されているヒストリー・ファイル・データ・セットに障害ヒストリー・レコードを書き込み中に入出力エラーが発生しました。

**システムアクション** 処理を続行しますが、ヒストリー・ファイルには現行ジョブの情報は含まれません。

**ユーザーの応答** 考えられ得る説明が *abend-code*、*reason-code*、および *reason-text* に示されます。

**IDI0034I**

Fault analysis skipped due to: *reason*

**説明**

以下のいずれかとして *reason* で示される原因により、ジョブが障害分析から除外されました。

- **Exclude オプションの指定 (高速)** 高速 Exclude オプション処理で、現行の障害との一致が検出されました ([高速 Exclude オプション処理 ページ 375](#)を参照)。
- **parmlib 構成メンバー Exclude オプションの指定** IDICNFxx 構成メンバー内で指定された Exclude オプションが現行の障害属性に一致しました。ユーザー・オプション・ファイル内に後から指定した Include または Exclude オプションは一致しませんでした。また、分析制御ユーザー出口により除外されたものがオーバーライドされることもありませんでした。
- **ユーザー・オプション・ファイル Exclude オプションの指定** IDIOPTS ユーザー・オプション・ファイル内で指定された Exclude オプションが現行の障害属性に一致しました。分析制御ユーザー出口により除外されたものはオーバーライドされませんでした。
- **分析制御ユーザー出口要求** IDICNFxx 構成メンバーまたは IDIOPTS ユーザー・オプション・ファイル内で指定された Exclude オプションが現行の障害属性に一致しませんでした。代わりに、分析制御ユーザー出口が除外を要求しました。
- **オプションの処理中に異常終了** *abend-code* リアルタイム・オプションの処理中に異常終了しました。

**システムアクション** 障害分析は実行されません。

**ユーザーの応答** 異常終了が発生し、問題が解決しない場合は、IBM サポートにご連絡ください。

**IDI0035W**

No dump records found in *data-set-name*

**説明** 再分析中にダンプ・データ・セットの読み取りを試行しましたが、ダンプ・データ・セットにデータが含まれていませんでした。

**システムアクション** ダンプ・データ・セットを使用せずに Fault Analyzer が処理を継続します。

**ユーザーの応答** Fault Analyzer に適切なダンプ・データ・セットが指定されていることを確認します。

**IDI0036E**

Abend *abend-code* during processing of *exit-type* user exit *exit-name*—refer to message explanation for problem determination information

**説明** ユーザー出口の実行中に異常終了しました。メッセージ・テキストの *abend-code* は、発生した異常終了のタイプ、*exit-type* は、異常終了したユーザー出口のタイプ、*exit-name* はユーザー出口の名前です。

**システムアクション** 処理を続行しますが、この出口に対する以後の呼び出しは行われません。

#### ユーザーの応答

デバッグ目的でこの状態のダンプを取得するには、例えば次の JCL ステートメントを使用して、DDname IDITRACE の割り振り先を DUMMY 以外にして、ダンプ・データ・セットを異常終了ジョブに割り振ります。

```
//IDITRACE DD SYSOUT=*
//SYSMDUMP DD DISP=SHR,DSN=my.dumpdsn
```

#### IDI0037E

不完全な障害ヒストリー・ファイル・レコードが検出されました。

**説明** 再分析のために障害エントリーをオープンしようとしているときに、エラーが発生しました。

**システムアクション** 再分析は終了します。

**ユーザーの応答** 問題が解決しない場合は、IBM® サポートに連絡してください。

#### IDI0038W

I/O error writing to softcopy book cache: *reason*

**説明** ソフトコピー・ブックのキャッシュ・データ・セットへの書き込みを試行した際に、エラーが発生しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** 考えられ得る説明が *reason* に示されます。問題が解決しない場合は、データ・セットを削除して、再割り振ります。

#### IDI0042W

Dump data set *data-set-name* with timestamp *timestamp* might not match current history file fault entry

**説明** ダンプ・データ・セット名のダンプ・ヘッダー・レコード・タイム・スタンプが、現行のヒストリー・ファイル障害項目に記録されているタイム・スタンプより 5 分以上前か、これより 35 分を超えて後です。

**システムアクション** 処理を続行します。

**ユーザーの応答** ダンプ・データ・セットが、他の障害のダンプの内容により再使用または復元されていないことを確認します。

#### IDI0043W

Fetch failed for message module *module-name* - using language *old-language-option* instead of *new-language-option*

**説明** 多文化サポート・メッセージ・モジュールをストレージに保管しようとしたのですが、失敗しました。 *module-name* に示されるロード・モジュールの名前は、IDIHM とこれに続く有効な言語オプションから構成されています。言語オプションが、サポートされない言語 ID を指定しているか、モジュールが常駐するロード・モジュール・ライブラリーが、MVS™ 検索パスでは検出されませんでした。

**システムアクション** *old-language-option* に示されている現行の言語オプション設定が保持されます。

**ユーザーの応答** Language オプションが、サポートされる言語 ID を指定していることと ([Language ページ 607](#)を参照)、通常の MVS™ 検索パスにより Fault Analyzer がロード・モジュールを使用できることを確認してください。



**IDI0044I**

Current® fault is a duplicate of fault ID *faultid* - the duplicate count is *count*

**説明** 有効な NoDup(NORMAL(*hours*)) オプションを使用した結果、Fault Analyzer は、現行の障害が、同じ履歴・ファイル内の既存の障害と重複するものであると判別しました。既存の障害の重複カウントは1つずつ増やされ、メッセージに *count* として表示されます。重複する障害を判別するために使用される条件については、[NoDup ページ 611](#)を参照してください。

**システムアクション** 処理を続行します。End Processing ユーザー出口を使用して、重複する障害処理の通常の動作を変更しない限り、システム・ダンプ (存在する場合) と履歴・ファイル項目の両方が抑止されます。

**ユーザーの応答** なし

**IDI0046I**

MVS™ SVC dump registration skipped due to: *reason*

**説明**

*reason* に示された以下のいずれかの理由により、Fault Analyzer IDIXTSEL ポスト・ダンプ出口による MVS™ SVC ダンプ登録が実行されませんでした。

- **parmlib 構成メンバー Exclude オプションの指定** IDICNFxx 構成メンバー内で指定された Exclude オプションが現行の障害属性に一致しました。ユーザー・オプション・ファイル内に後から指定した Include または Exclude オプションは一致しませんでした。また、分析制御 (ダンプ登録) ユーザー出口により除外されたものがオーバーライドされることもありませんでした。
- **分析制御ユーザー出口要求** IDICNFxx 構成メンバー内で指定された Exclude オプションが現行の障害属性に一致しませんでした。代わりに、分析制御 (ダンプ登録) ユーザー出口が除外を要求しました。
- **オプションの処理中に異常終了** *abend-code* オプションの処理中に異常終了しました。

**システムアクション** 障害項目は作成されません。

**ユーザーの応答** なし

**IDI0047S**

IBM Fault Analyzer internal abend *abend-code*

**説明** Fault Analyzer が、示されている異常終了コードで異常終了しました。このメッセージは [IDI0120S ページ 722](#) メッセージに相当しますが、発行される事情が異なります。

**システムアクション** 処理は終了します。

**ユーザーの応答** IBM サポートに連絡してください。

IDI0047S メッセージの発行時に生成されたダンプは、問題分析に必要です。既存のリカバリー障害記録ダンプが使用できない場合、問題を再現する前に次の SLIP トラップを設定して取得できることがあります。

```
SL SET, ID=xxxx, MSGID=IDI0047S, A=SVCD, END
```

**IDI0048W**

不完全なミニダンプ: *expected-pages* ページが予期され、*read-pages* ページが読み取られました

**説明** リアルタイム分析中、ヒストリー・ファイルに保管されるべきミニダンプ・ページの数 *expected-pages* ページでした。ただし、再分析中には *read-pages* ページ以外検出されませんでした。

**システムアクション** 処理を続行します。

**ユーザーの応答** *read-pages* が *expected-pages* より小さい場合、IBM サポートに連絡してください。

---

#### IDI0049S

Fault ID *faultid* not found in history file

**説明** 存在しない障害 ID が FaultID オプションに指定されました。

**システムアクション** 処理は終了します。

**ユーザーの応答** 使用するヒストリー・ファイルに、FaultID オプションに指定された障害 ID が含まれていることを確認してください。

---

#### IDI0050S

Fault reanalysis attempted without FaultID or DumpDSN option specified

**説明** 障害の再分析が行われましたが、FaultID オプションも DumpDSN オプションも指定されていません。これらのオプションのどちらかが指定されないと、Fault Analyzer は障害再分析を実行できません。

**システムアクション** 処理は終了します。

**ユーザーの応答** FaultID または DumpDSN オプションのいずれかを指定して、再分析を再試行してください。

---

#### IDI0052I

*count* page minidump suppressed from the fault entry being created

**説明** 障害項目を作成中のため *count* 4K ページで構成されているミニダンプが、Fault Analyzer により抑止され、ヒストリー・ファイルに書き込まれませんでした。ただし、障害項目の残りの部分については書き込みを試行します。この抑止は、ミニダンプ・ページが最大数を超えたためか、あるいは呼び出された End Processing ユーザー出口によりそのように決定されていたためであると考えられます。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

#### IDI0053I

Fault history file entry suppressed due to: *reason*

##### 説明

ヒストリー・ファイルは、現行の障害について更新されませんでした。抑止の理由が、以下のいずれかとして *reason* に示されます。

##### DUMMY ヒストリー・ファイルの指定

IDIHIST DDname が DUMMY と指定されました

##### History file ENQ timeout

Fault Analyzer は、設定された時間間隔内でヒストリー・ファイルを逐次化できませんでした。

**History file access error**

ヒストリー・ファイルのオープン/読み取り障害であれば、多くの場合、このメッセージの前にメッセージ [IDIO016E ページ 699](#) または [IDIO078E ページ 712](#) が発行されています。

**Duplicate fault or End Processing user exit**

障害の検出が重複したことが理由の場合、メッセージ [IDIO044I ページ 705](#) がこのメッセージの前に発行されています。それ以外の場合には、End Processing ユーザー出口が抑止を要求しました。

**End Processing user exit**

障害項目リフレッシュ処理の間、End Processing ユーザー出口が抑止を要求しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** 障害項目の抑止の理由が、重複と見なされるものではなく、ユーザー出口による抑止の要求でもない場合で、問題が解決しないときは、抑止の原因を調査する必要があります。前に発行されたメッセージに追加情報が提供されているかを確認するか、または別のユーザーやジョブが引き続き履歴ファイルに対する ENQ を保持しているかを確認してください。

**IDIO055E**

*num* MB の 31 ビットのストレージが現在使用できないため、Fault Analyzer プロセスが除外されました

**説明** 分析が実行されるのに十分な境界以上のストレージが、領域内で使用できませんでした。

**システムアクション** 処理は終了します。

**ユーザーの応答** Fault Analyzer により分析される異常終了ジョブに、少なくとも *num* MB の境界以上のストレージが、領域内で使用可能であることを確認してください。必要なストレージは、JOB または EXEC JCL ステートメントの REGION パラメーターを使用することで、使用可能になります。[推奨されるストレージ ページ 297](#)を参照してください。

**注:**

- REGION パラメーターの MVS™ インプリメンテーションが原因で、既に指定済みの REGION パラメーター内の値に、必要な量のストレージを単独に追加するだけでは十分でない可能性があります。
- CICS® の場合、このストレージ全体の量は、EDSALIM パラメーターからではなく、JOB または EXEC JCL ステートメントの REGION パラメーターから提供されなければなりません。

ストレージの制限により、Fault Analyzer はリアルタイム分析を実行できませんでしたが、リカバリー障害記録 ([リカバリー障害記録 ページ 57](#)[リカバリー障害記録 ページ 57](#)を参照) は非 CICS 環境で試行されます。

**IDIO056E**

REXX environment initialization failed, IRXINIT rc=*rc* reason=*reason*

**説明** REXX 環境を初期化しようとしたのですが、失敗しました。メッセージ・テキスト内の *rc* は戻りコードであり、*reason* は REXX 初期化ルーチン IRXINIT によって戻される理由コードです。戻りコードおよび理由コードについては、「z/OS®: TSO/E REXX 解説書」を参照してください。

**システムアクション** 処理を続行します。ただし、REXX exec ユーザー出口は呼び出されず、診断情報は IDITRACE DDname に書き込まれません。

**ユーザーの応答** 戻りコードおよび理由コードについては、「z/OS®: TSO/E REXX 解説書」を参照してください。

#### IDI0058W

一時作業データ・セットの割り振りに失敗しました: Type=*type* RC=*return-code* Error=*error-code* Info=*info-code*

**説明** 一時作業データ・セットの動的割り振りに失敗しました。

**システムアクション** データ・セットなしで処理を続行します。

**ユーザーの応答** 「z/OS@: MVS™ Programming: Authorized Assembler Services Guide」で、戻りコード、エラー・コード、および情報コードについて参照してください。

---

#### IDI0059W

論理 parmlib 連結リスト・エラー: RC=*return-code* Reason=*reason*

**説明** 論理 parmlib 連結の中のデータ・セットを判別するために論理 parmlib リスト・サービスの使用が試みられましたが失敗しました。

**システムアクション** IDICNFxx parmlib メンバーを使用せずに処理を続行します。

**ユーザーの応答** 「z/OS@: MVS™ Programming: Assembler Services Reference」で、戻りコードおよび理由コードの説明を参照してください。

---

#### IDI0061E

Data set organization of history file *data-set-name* is not PDS or PDSE

**説明** Fault Analyzer は、無効なフォーマットのヒストリー・ファイルのデータ・セットを使用して呼び出されました。

**システムアクション** 処理を続行しますが、障害エントリーを書き込むことはできません。

**ユーザーの応答** ヒストリー・ファイルが PDS または PDSE のいずれかであることを確認してください。

---

#### IDI0062W

ファイル *data-set-name* 内にエラーがあります。 *reason*

**説明** データ・セット *reason* の読み取り時に *data-set-name* で示されるエラーが発生しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** メッセージ IDI0062W は、特定の障害項目について、または \$\$INDEX メンバーについて発行される可能性があります。破損した HI セグメントが原因で、\$\$INDEX メンバーについてメッセージ IDI0062W が発行された場合は、ヒストリー・ファイルの次回の更新時(例えば、新しい障害項目が書き込まれるとき)に、その HI セグメントに対し、\$ \$BACKUP メンバーからのリカバリーが自動的に試みられます。示されたデータ・セットが有効なヒストリー・ファイル PDS または PDSE であることを確認し、必要に応じてエラーのメンバーを削除してください。ただし、問題が解消されない場合を除いて、\$\$INDEX メンバーを削除しないように注意してください。その削除の結果として、重複障害情報が失われる可能性があるからです。

---

#### IDI0063W

ファイル *data-set-name* の削除に失敗しました。 *reason*

**説明** データ・セット *data-set-name* を削除しようとしたましたが、失敗しました。原因は *reason* で示されています。

システムアクション 処理を続行します。

ユーザーの応答 障害について示された原因を参照して、データ・セットを削除してください。

#### IDI0064W

CICS® Exit *command* failed. Response=*response* reason=*reason*

説明 CICS® コマンドが失敗しました。考えられる応答コードを以下の表にリストします。

表 26. 応答コード

応答コード	意味
1	OK
2	例外
3	災害
4	無効
5	カーネル・エラー
6	ページ済み

各コマンドに関連する理由コードは以下のとおりです。

表 27. 理由コード

理由コード	意味
<b>FREEMAIN</b>	
(適用可能な理由コードなし)	
<b>GETMAIN</b>	
1	ストレージ不足
<b>INQ_APPLICATION_DATA</b>	
1	DPL プログラム
2	トランザクション環境なし
3	トランザクション・ドメイン・エラー
4	無効な機能
5	異常終了
6	ループ
7	照会失敗
<b>INQUIRE_TRANSACTION</b>	
1	トランザクション環境なし
3	バッファが小さすぎる
7	無効なトランザクション・トークン
37	異常終了

表 27. 理由コード

(続く)

理由コード	意味
39	ループ
<b>START_PURGE_PROTECTION</b>	
(適用可能な理由コードなし)	
<b>STOP_PURGE_PROTECTION</b>	
(適用可能な理由コードなし)	
<b>WAIT_MVS</b>	
3	タスク取り消し
4	タイムアウト

**システムアクション** 処理を続行します。

**ユーザーの応答** 障害については、示された原因を参照してください。

**IDI0065W**

CICS® Exit *action* failed. Return code=*rc*

**説明** CICS® 呼び出し出口内で実行されたアクションが失敗しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** 障害については、示されたアクションと戻りコードを参照してください。

**IDI0066I**

CICS® Fast No Dup processing for task *task-number* found duplicate for *program-name abend-code fault-id*

**説明** NoDup(CICSFAST(...)) オプションが友好であるため、Fault Analyzer は、CICS® トランザクション異常終了が、前の CICS® トランザクション異常終了の重複であると判別しました。重複する障害を判別するために使用される条件については、[NoDup ページ 611](#)を参照してください。メッセージ・テキストの *fault-id* は、重複と見なされた現行障害の障害 ID を示します。

**システムアクション** 異常終了 CICS® トランザクション用のヒストリー・ファイル項目を作成せずに処理を継続します。

**ユーザーの応答** なし。

**IDI0068I**

*token* の Fault Analyzer SubTask が *num* 分間実行されています。

**説明** このメッセージが CICS® で発生するのは、タスク異常終了が 10 分以上、Fault Analyzer SubTask を実行していたときに、新しい CICS® タスク異常終了も、Fault Analyzer SubTask 処理の実行を必要としている場合です。システム上の CPU 使用率が非常に高く、妥当な時間で分析が行われない場合に、この状態が発生します。または、Fault Analyzer SubTask の処理に問題があることを示しています。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0069E**

*command-name* failed. Resp=*response-code* field=*value*

**説明** 不明なエラーが発生しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** IBM サポートに連絡してください。可能な場合はジョブ・ログを送信してください。

---

**IDI0071I**

*function-name* of program *program-name* (*api,concurrency*) completed successfully

**説明** このメッセージは、実行されたコマンドが完了したことの確認です。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

**IDI0072I**

Program *program-name* is *status*

**説明** このメッセージは、現行の出口プログラムの名前と状況を示します。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0073I**

Usage: CFA <Install|Uninstall>

**説明** CFA トランザクションに渡された引数が誤っていました。このメッセージは、正しい使用法をユーザーに知らせます。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0074E**

User not authorized to issue command *command-name*

**説明** 実行中のタスクに関連したユーザーに、このコマンドを使用する権限がないか、このコマンドが必要とする方法でリソースにアクセスする権限がありません。

**システムアクション** コマンドの試行が失敗しました。

**ユーザーの応答** 必要なコマンド権限があることを確認してください。

---

**IDI0075E**

IDIXCCEE CICS 出口が異常終了しています。CICS SOS pending.

**説明** 分析を開始するのに十分なストレージが使用できませんでした。このストレージの欠如は、高い CICS® システム・アクティビティーの場合、または複数の CICS® トランザクション異常終了が、Fault Analyzer により並行して分析されている場合に発生する可能性があります。

**システムアクション** 処理は終了します。

**ユーザーの応答** この問題の原因が、高い CICS® システム・アクティビティー、または複数の並行した CICS® トランザクション異常終了の分析ではなかった場合、CICS® DSA サイズの増設をお勧めします。

---

**IDI0076E**

Fault Analyzer CSVINFO MIPR ルーチンの異常終了が発生しました

**説明** ロード・モジュールについての情報を取り出すために CSVINFO マクロの処理中に、異常終了が発生しました。

**システムアクション** ロード・モジュールに対する CSVINFO 処理が終了します。

**ユーザーの応答** IBM サポートに連絡してください。

---

**IDI0077I**

Fault Analyzer 内部診断: *text*

**説明** このメッセージは、特定の診断情報を表示する場合に使用される内部 Fault Analyzer メッセージです。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0078E**

次に示す原因のために、障害ヒストリー・ファイル *data-set-name* をオープンできませんでした。 *reason*

**説明** *data-set-name* で識別される履歴ファイルが、*reason* に示された原因のために、オープンできませんでした。

*reason* が [DDfopenw() rc=X'800sss'rc'] の形式で表示される場合、この理由は、オープン処理の間の異常終了を示します。説明:

**sss**

16 進 MVS™ システム異常終了コード。

**rc**

関連の 16 進理由コード。

**システムアクション** 障害項目がヒストリー・ファイルに書き込まれていない可能性があります。

**ユーザーの応答** オープンが失敗した原因を判別し、ジョブを再実行依頼します。ファイル定義用の属性が VB LRECL 10000 PDS または PDSE であることを確認してください。履歴ファイルが、シスプレックス全体で共用される PDSE である場合、追加情報について、[シスプレックス全体でのヒストリー・ファイルの共用 ページ 364](#)を参照してください。



---

**IDI0081I**IEWBIND unusual condition *function module-name rc=reason-code*

**説明** バインドを使用中にエラーが発生しました。 *reason-code* が 83000526 の場合、入力モジュールに異常な状態が検出されたことを示します。この状態は、独自の管理情報をロード・モジュールに追加するサード・パーティー製品を使用する場合に当てはまります。このメッセージを抑止するには、`Quiet(IDI0081I)` を使用してください。

**システムアクション** 処理を続行します。

**ユーザーの応答** *reason-code* の値は、バインド・プログラム API の理由コードです。バインド・プログラム API の理由コードは、「z/OS®: MVS™ プログラム管理: 拡張機能」に記載されています。

---

**IDI0082E**DB2® *subsys-id* Call Level Interface error: *reason*

**説明** *subsys-id* で識別された DB2® サブシステムの DB2® コール・レベル・インターフェース (CLI) を使用した DB2® 情報の取得が失敗しました。このエラーの詳細については、*reason* を参照してください。

**システムアクション** 処理を続行しますが、分析が完全でない可能性があります。

**ユーザーの応答** DB2® コール・レベル・インターフェースがインストールされ、必要なセットアップを実行済みであることを確認してください。DBRM をパッケージにバインドするアプリケーション・プランが存在しなければなりません。デフォルトのアプリケーション・プランは、DB2® SDSNSAMP データ・セットのメンバー DSNTIJCL 内のサンプル・ジョブを使用して作成できます (詳しくは、[DB2 のバインディング ページ 422](#) を参照してください)。 *reason* が「SQL return code -1 for SQLAllocConnect to DB2® system *system-name*」である場合、Fault Analyzer IDIS サブシステムを [Fault Analyzer IDIS サブシステムの使用 ページ 319](#) の説明に従って開始することで、問題が解決する可能性があります。メッセージ *reason* に組み込まれるのは通常は SQLERROR ODBC 関数呼び出しの結果です。これにより、エラーに関する詳細情報が提供されます。特に注意するものは、REASON= キーワードに続く 8 桁の 16 進理由コードです (例えば、REASON=00f30034)。この理由コードの説明は、次を参照してください:「[DB2® for z/OS®: コード](#)」

---

**IDI0083E**Fault Analyzer SVC エラー: *reason*

**説明** Fault Analyzer SVC 実行時にエラーが発生しました。このエラーの詳細については、*reason* を参照してください。このメッセージが断続的に出される場合は、Fault Analyzer がロックを保持せずにサブシステム・アドレス・スペースを全探索している可能性があります。

**システムアクション** 処理を続行しますが、分析が完全でない可能性があります。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。 *reason* が、Fault Analyzer SVC 実行時の異常終了を示している場合、SVC ダンプが常に書き込まれます。診断のために、このダンプを保存しておいてください。

---

**IDI0084E**

Fault Analyzer 終了とメイン・タスク・レベルの不一致

**説明** 呼び出し出口が、Fault Analyzer メイン・モジュール IDIDA とは異なるバージョンまたは保守レベルであることが判明しました。

**システムアクション** 処理を続行しますが、エラーが発生する可能性があります。

**ユーザーの応答** 使用されるすべての Fault Analyzer モジュールが、同じレベルのバージョンと保守であることを確認してください。

---

#### IDI0085E

Fault Analyzer パージ済み、切り離し中のサブタスク TCB

**説明** 異常終了した CICS® タスクが、オペレーターによってパージされたときに、Fault Analyzer を実行していました。Fault Analyzer は、分析サブタスク TCB を消去します。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

#### IDI0086E

size k の 24 ビットのストレージが現在使用できないため、Fault Analyzer プロセスが除外されました

**説明** 分析が実行されるのに十分な境界以下のストレージが、使用できませんでした。size は、必要なストレージ量 (KB) を示します。

**システムアクション** 現行の障害について、障害分析が除外されます。

**ユーザーの応答** 境界以下の十分なストレージが使用可能であることを確認してください。詳しくは、[推奨されるストレージ ページ 297](#)を参照してください。ストレージの制限により、Fault Analyzer はリアルタイム分析を実行できませんでした。リカバリー障害記録 ([リカバリー障害記録 ページ 57](#)を参照) は非 CICS 環境で試行されます。

---

#### IDI0087I

size Meg of 31 bit storage could be provided by SETPROG LPA,ADD,MOD(module-list),DSN=LNKLST

**説明** 境界以上のストレージが不足しているため分析が実行できなかった場合、相当なサイズがあるすべての必要なモジュールが、まだ LPA にロードされていないときに、これらのモジュールを LPA にロードした場合に実現した領域サイズ・スペースの節約量を示すために、このメッセージが発行されます。詳しくは、[推奨されるストレージ ページ 297](#)を参照してください。メッセージ [IDI0055E ページ 707](#) がこのメッセージに先行し、取得できなかった境界以上の必要な領域サイズを示します。

**システムアクション** 処理は終了します。

**ユーザーの応答** メッセージ [IDI0055E ページ 707](#) に示されている境界以上のストレージ量が使用可能であることを確認してください。このメッセージのテキストに記載されている SETPROG コマンドを実行すると、size で示されている MB 数を使用可能にすることができます。

---

#### IDI0088E

Fault Analyzer が IEWBIND 異常終了 *abend-code* を取り込みました。処理は続行します

**説明** CSECT のロード・モジュールへのマップを試行中、Fault Analyzer が、異常終了コードを表示して異常終了したバインド・プログラムを呼び出しました。異常終了コードのフォーマットは xxxyyy です。ここで、xxx は 3 桁の 16 進システム異常終了コードで、yyy は 4 桁の 10 進ユーザー異常終了コードの 3 桁の 16 進表現です。

**システムアクション** 処理を続行しますが、モジュール情報は不完全である可能性があります。

**ユーザーの応答** バインド障害の原因を判別してください。

**IDI0089I**

Fault Analyzer Subsystem *task name* RC=*rc*, Rsn=*rsn* *description*

**説明** 表示可能な場合、Fault Analyzer サブシステムのタスク名が *task-name* に表示されます。このメッセージは、予期しない状態が Fault Analyzer IDIS サブシステム・インターフェースで発生した場合に発行されます。メッセージ *description* に SQLError ODBC 関数呼び出しの結果が含まれる場合があります。これにより、エラーに関する詳細情報が提供されます。特に注意するものは、REASON= キーワードに続く 8 桁の 16 進理由コードです (例えば、REASON=00f30034)。この理由コードの説明は、次を参照してください: DB2® for z/OS®: コード

**システムアクション** 処理を続行しますが、分析が完了していない可能性があります。

**ユーザーの応答** IDI0082E ページ 713 など、状況の解明に役立つ他のメッセージが発行されていないか確認してください。

**IDI0090E**

Execution of REXX exec *exec-name* failed, IRXEXEC rc=*return-code*

**説明** このメッセージは、REXX ユーザー出口の実行で障害が発生した場合に発行されます。

**システムアクション** 処理を続行しますが、その出口での実行は失敗します。

**ユーザーの応答** IDITRACE を使用して問題について説明する REXX メッセージを参照します。「z/OS®: TSO/E REXX 解説書」で、IRXEXEC からの戻りコードの説明を参照してください。rc=20 の場合には、*exec-name* に示す *exec* が IDIEXEC 連結で使用可能かどうか確認してください。

**IDI0091S**

GETMAIN が IDIXTSEL 初期エンタリーに失敗しました

**説明** IDIXTSEL MVS™ ダンプ後出口の実行に使用可能なストレージが不足しています。

**システムアクション** ダンプ登録処理は終了します。

**ユーザーの応答** なし。

**IDI0092S**

*condition* exceeded, the subtask is canceled

**説明**

メッセージ・テキストの中の *condition* は、以下のいずれかになります。

**Short term interval**

環境検査が、指定された制限時間内に完了しませんでした。この検査がリアルタイム障害分析を始める前に実行され、制限時間を超過しているのは、環境の問題の兆候であるか、分析中に重大な遅延またはハングが発生した可能性があります。

**IDIDA 実行の時間**

リアルタイム分析が、許可された最大時間内に完了しませんでした。制限時間は、分析中のさまざまなチェックポイントにおける経過時間を基に自動的に調整されます。

**システムアクション** 処理は終了します。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

問題分析には、IDI0092S メッセージが発行されたときに生成されたダンプが必要です。既存のリカバリー障害記録ダンプが使用できない場合、問題を再現する前に次の SLIP トラップを設定して取得できることがあります。

```
SL SET, ID=xxxx, MSGID=IDI0092S, A=SVCD, END
```

「Short term interval」以外の状態で (上の「Short term interval」を参照してください)、Fault Analyzer 待機/ループ保護機構を無効にするには、NoLoopProtection オプションを使用できます。詳しくは、[LoopProtection ページ 609](#)を参照してください。

---

#### IDI0093W

Binder processing stopped because *num* meg of 31 bit storage is not currently available

**説明** バインダーを正常に実行するために使用可能な境界以上のストレージが、十分に存在しません。

**システムアクション** バインダー処理は終了しますが、Fault Analyzer 分析は継続します。

**ユーザーの応答** 可能な場合、使用可能な 31 ビット領域のサイズを増加させ、再試行してください。

---

#### IDI0094W

Binder processing stopped because *num* k of 24 bit storage is not currently available

**説明** バインダーを正常に実行するために使用可能な境界以下のストレージが、十分に存在しません。

**システムアクション** バインダー処理は終了しますが、Fault Analyzer 分析は継続します。

**ユーザーの応答** 可能な場合、使用可能な 24 ビット領域のサイズを増加させ、再試行してください。

---

#### IDI0095W

Unexpected condition found in *source-location description*

**説明** IBM サポートに報告する必要がある状況が発生しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** IBM サポートに連絡してください。

---

#### IDI0096S

CICS® Task *task-id* was force purged by an operator while *state* fault analysis

**説明** CICS® トランザクションの障害分析がオペレーター介入により強制終了されました。メッセージ・テキストで、*task-id* は強制的にページされたタスクを示します。*state* は、強制終了された障害分析のアクティビティーを示して「waiting for」または「performing」のいずれかになります。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

#### IDI0097W

Unsupported REXX execution environment detected - no REXX services available

**説明** ジョブ・ステップ TCB の保護キーが 8 以外のキーに設定されています。この環境は REXX でサポートされない環境であるため、障害分析で REXX サービスは使用できません。このサービスの欠如には、REXX ユーザー出口に対するすべての呼び出し、および IDITRACE DDname へのすべての出力が含まれます。

**システムアクション** 処理は REXX サービスを使用せずに継続されます。

**ユーザーの応答** なし。

---

#### IDI0098S

出口 *exit-name* がインストールされていません。Fault Analyzer SVC が見つかりません

**説明** *exit-name* で示された Fault Analyzer 呼び出し出口で、インストールを試みました。しかし、Fault Analyzer SVC がインストールされていないため、処理は終了します。

**システムアクション** 処理は終了します。

**ユーザーの応答** Fault Analyzer SVC (ステップ 5 ページ 308 を参照) をインストールし、再試行してください。

---

#### IDI0099S

*transaction-id* transaction cannot be used, Fault Analyzer SVC not found

**説明** Fault Analyzer CFA トランザクション (*transaction-id* に示された、別の名前呼び出されている可能性があります) の使用が試みられました。しかし、Fault Analyzer SVC がインストールされていないため、処理は終了します。

**システムアクション** 処理は終了します。

**ユーザーの応答** Fault Analyzer SVC (ステップ 5 ページ 308 を参照) をインストールし、再試行してください。

---

#### IDI0100S

Fault Analyzer を使用できません。Fault Analyzer SVC が見つかりません

**説明** Fault Analyzer SVC がインストールされていない状態で、Fault Analyzer の呼び出しが試みられました。

**システムアクション** 処理は終了します。

**ユーザーの応答** Fault Analyzer SVC (ステップ 5 ページ 308 を参照) をインストールし、再試行してください。

---

#### IDI0101I

Fault Analyzer CICS DUMPTABLEEXCLUDE オプションのために処理がスキップされました。異常終了コード *abend-code*

**説明** CICS DumpTableExclude オプションを使用した結果、CICS® トランザクション障害の分析が除外されました。このオプションの詳細については、[CICS DumpTableExclude ページ 574](#)を参照してください。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

#### IDI0102S

Fault Analyzer 実行環境が不適切なために処理が終了しました

**説明** 障害分析を始める前の実行環境の検査中に、異常終了が発生しました。

**システムアクション** 処理は終了します。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

#### IDI0103I

Binder processing of member *member-name* from *data-set-name* received a 'not found' condition.

**説明** MVS™ バインド・プログラムの呼び出しが失敗しました。

**システムアクション** 処理を続行しますが、分析が完了していない可能性があります。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

#### IDI0104S

IDINDFUE 作業域の上書き前または後 (またはその両方) のストレージ - 処理が終了しました。

**説明** Fault Analyzer が IDINDFUE プログラム出口を呼び出すときには、256 バイトの作業域を受け渡します。IDINDFUE から戻るとき、Fault Analyzer はこの作業域を囲むストレージが上書きされたことを検出しました。

**システムアクション** Fault Analyzer 処理は即時に終了します。(processing is terminated immediately.)

**ユーザーの応答** IDINDFUE プログラム出口コードを調べて、ストレージが上書きされた理由を判別してください。

---

#### IDI0105S

*module-name:line-number* Storage allocation for *dec-count* (*X'hex-count'*) bytes failed - processing terminated

**説明** ストレージの無効な要求が発生しました。このメッセージはメッセージ [IDI0005S ページ 696](#) と似ていますが、負の長さが含まれているすべてのストレージ割り振り要求に使用されます。

**システムアクション** 処理は終了します。

**ユーザーの応答** IBM サポートに連絡してください。

IDI0105S メッセージの発行時に生成されたダンプは、問題分析に必要です。既存のリカバリー障害記録ダンプが使用できない場合、問題を再現する前に次の SLIP トラップを設定して取得できることがあります。

```
SL SET, ID=xxxx, MSGID=IDI0105S, A=SVCD, END
```

---

#### IDI0106E

ENQ timed out for *major-name minor-name* held by *jobname* on *system-name*

**説明** 新規障害項目の作成前に、ヒストリー・ファイルに対して ENQ に設定された制限時間を超えました。制限時間は約 1 分です。ENQ リソースは、*jobname* システム上のジョブ *system-name* に保持されています。障害項目は作成されませんが、分析レポートは IDIREPRT に書き込まれます。

**システムアクション** 処理を続行します。

**ユーザーの応答** ENQ を満たすことができなかった理由を判別してください。この理由は、履歴ファイル・メンバーを編集している TSO/ISPF ユーザーである可能性があります。

**IDI0107I**

*transaction-id date time exit status*

**説明** このメッセージは、Fault Analyzer Control Transaction (*transaction-id*) を使用して、1 つ以上の出口 (CICS® で識別される XPCABND、XDUREQ、LE Exit、または *exit* 出口) の状況を変更すると必ず発行されます。*status* フィールドには、「Uninstalled」または「Installed」と表示されます。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

**IDI0108I**

IDIS サブシステムは、PDSE ヒストリー・ファイルではないため、*data-set-name* を処理しません。

**説明** このメッセージは、PARM='UPDINDEX' を使用して、PDSE でないヒストリー・ファイルを開こうとすると、Fault Analyzer IDIS サブシステムによって発行されます。メッセージ・テキスト *data-set-name* は、IDIS サブシステムでキャッシュできないヒストリー・ファイルを示します。IDIS サブシステムのキャッシングの詳細については、[ヒストリー・ファイル \\$\\$INDEX データのキャッシュ ページ 320](#)を参照してください。

**システムアクション** 処理は継続されますが、ヒストリー・ファイルの \$\$INDEX メンバーに対するすべての入出力がリクエスター (リアルタイム分析、バッチまたは対話式再分析、あるいは Fault Analyzer ISPF インターフェース) によって実行されます。

**ユーザーの応答** ヒストリー・ファイルの \$\$INDEX メンバーを、Fault Analyzer IDIS サブシステムに管理させることで、できるだけパフォーマンスの向上を図るには、PDSE ヒストリー・ファイルを使用する必要があります。

**IDI0109E**

PC recovery entered psw *program-status-word* abend=*abend-word*

**説明** このメッセージは、Fault Analyzer サブシステムのプログラム呼び出しインターフェースでエラーが発生した場合に発行されます。

通常は、このメッセージの直後に、エラーの詳細情報を示す追加のメッセージが表示されます。例:

```

IDISSPC PC recovery IDISSPC=16703250 IDISAMAN=16700000
IDISSPC R0-R3      00000001 171EAE20 80FF28FE 00000002
IDISSPC R4-R7      008A6770 008A69E0 008FD0C8 00F7DA00
IDISSPC R8-R11     00000000 00FF24AC 00000C80 00F7DA00
IDISSPC R12-R15    00000000 00000000 80FF2A2A 808A69E0
IDISSPC AR0-AR3    00000000 00000000 00000000 00000000
IDISSPC AR4-AR7    00000000 00000001 00000000 00000000
IDISSPC AR8-AR11  00000000 00000001 00000000 00000000
IDISSPC AR12-AR15 00000000 00000000 00000000 00000000

```

**システムアクション** 処理は継続されますが、失敗したサブシステムのサービスによって、予期しない結果が生じる可能性があります。S102 異常終了の場合、この問題の原因としては、Fault Analyzer サブシステムにサービスを要求していたジョブまたはタスクが、サブシステムの要求が完了する前に取り消されたことが考えられます。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

**IDI0111W**

IDI0092S のタイマーの有効期限が切れましたが、NoLoopProtection オプションが設定されているため、実行を続行します

**説明** このメッセージは、ループ/待機保護機構インターバル・タイマーの有効期限が切れたが、NoLoopProtection オプションが有効な場合に発行されます。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0112W**

*db2\_name* SYSIBM.SYSDBRM access time took *num\_secs* seconds, further IDIS requests for this table suspended for 30 minutes—create index on SYSIBM.SYSDBRM for improved performance

**説明** このメッセージは、IDIS サブシステムが DB2® サブシステム名 *db2\_name* の SYSIBM.SYSDBRM DB2® カタログ テーブルを照会しようとしたときに、*num\_secs* 値で示される最大予想時間を超えた場合に発行されます。識別された DB2® サブシステム上の SYSIBM.SYSDBRM テーブルへのアクセス試行は、その後 30 分間行われません。

**システムアクション** 処理は継続されますが、このメッセージが発行される原因となった障害、および同じ DB2® サブシステムが関係するそれ以降の障害についての分析レポートから、一部の DB2 関連情報が欠落する可能性があります。

**ユーザーの応答** パフォーマンスを向上させるために、SYSIBM.SYSDBRM に索引を作成してください。詳しくは、[Fault Analyzer DB2 パフォーマンスの向上 ページ 422](#)を参照してください。

---

**IDI0113W**

Subtask *tcb-addr(description)* return ECB=*ecb-value* during problem reanalysis under CICS®

**説明** CICS® で対話式再分析の実行中にエラーが発生しました。このエラーは、異常終了した、または CICS® での対話式再分析コンポーネントのセットアップが完全でなかった ([以下の環境での対話式再分析の実行 CICS ページ 286](#)を参照) ことが原因と考えられます。

**システムアクション** 対話式再分析は終了します。

**ユーザーの応答** 問題の説明に役立つ他のメッセージがないか確認してください。問題が解決しない場合は、IBM サポートに連絡してください。

---

**IDI0114W**

XMEM POST ABEND S102, requester termination

**説明** この問題は、Fault Analyzer サブシステムにサービスを要求していたジョブまたはタスクが、サブシステムの要求が完了する前に取り消された場合に発生することがあります。

**システムアクション** 分析は既に取り消されています。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

**IDI0115E**

LE エンクレーブの異常終了 *system-abend-code user-abend-code*。実行を続行します



**説明** 障害分析中に異常終了しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

#### IDI0116E

IPVLANGX の異常終了 *abend-code*。実行を続行します

**説明** IDILANGX プログラム内で異常終了しました。

**システムアクション** 処理を続行しますが、ソース・レベル情報は不完全である可能性があります。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

#### IDI0117E

ABEND<*abend-code*> during return POST, request from task *jobnamejob-id* may have been canceled

**説明** Fault Analyzer サブシステム POST 処理で異常終了が発生しました。この異常終了は、*jobname* および *job-id* によって示されるタスクからの要求の CANCEL が原因で起きた可能性があります。

**システムアクション** 処理を続行します。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

#### IDI0118W

CICS® task *task-number* abend *abend-code* analysis skipped due to max waiting exceeded.  
CICSFAtasks(*max\_slots,max\_waits*)

**説明** 別の障害が発生した時点で、既に分析の待機を許可される障害の最大数に達していました。この限度は、DeferredReport オプションの CICSFAtasks サブオプションによって制御され、有効な現行値がメッセージ・テキスト内に提供されます。

**システムアクション** 分析はスキップされるので、通常の (すなわち、Fault Analyzer を使用しない) CICS® トランザクション・ダンプ分析の実行が必要になります。

**ユーザーの応答** 既に最大値を指定している場合を除き、DeferredReport オプションの CICSFAtasks サブオプションを使用して、使用する並列実行タスクの数を増やすか、または分析の待機を許可される障害の数を増やすことができます。[DeferredReport ページ 584](#)を参照してください。

---

#### IDI0119E

IDIS subsystem request *server-id resource-id* by *jobname job-id* has hung and will be canceled

**説明** Fault Analyzer IDIS サブシステムが、要求が予想どおりに終了しなかったことを検出しました。

**システムアクション** 要求は取り消されます。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

**IDI0120S**IBM Fault Analyzer internal abend *system-abend-code* user-abend-code

**説明** Fault Analyzer が、システム異常終了コード *system-abend-code* またはユーザー異常終了コード *user-abend-code* を出して異常終了しました。このメッセージは [IDI0047S ページ 705](#) メッセージに相当しますが、発行される事情が異なります。

**システムアクション** 処理は終了します。

**ユーザーの応答** 問題の原因を説明している可能性のある重大な (S レベルの) Fault Analyzer メッセージ (例えば、メッセージ [IDI0005S ページ 696](#)。これはストレージ不足の問題を示しています) が以前に発行されていないかどうかを確認してください。S レベルのメッセージが以前に発行されていない場合は、IBMサポートに連絡してください。

---

**IDI0121I**ImageFast NoDup processing found duplicate of *ims\_pgm fault\_id histfile*

**説明** Fault Analyzer は、現在の障害が IMS™ NoDup(ImageFast(...)) の重複であると判断しました。(IMS™ NoDup(ImageFast(...)) 重複検出については、[NoDup ページ 611](#)を参照してください。)現在の障害が重複と見なされる元になった、先に発生した障害は IMS™ プログラム名 (*ims\_pgm*)、割り当てられた障害 ID (*fault\_id*)、およびヒストリー・ファイル・データ・セット名 (*histfile*) によって示されます。

**システムアクション** 処理を続行します。障害分析は実行されませんが、先に発生した障害の重複カウントが1つ増え、後で Fault Analyzer ISPF インターフェースで表示できるように ([障害項目の重複ヒストリーの表示 ページ 139](#)を参照してください)、重複の詳細が記録されます。

**ユーザーの応答** なし。

---

**IDI0122W**ユーザー出口 IDIALLOC コマンドが失敗しました: DD=*ddname* DSN=*data-set-name* RC=*return-code* Error=*error-code*  
Info=*info-code*

**説明** REXX ユーザー出口が、データ・セットを動的に割り振る IDIALLOC コマンドを発行しましたが、割り振りに失敗しました。

**システムアクション** *data-set-name* に示されるデータ・セットの割り振りを行わずに処理を続行します。

**ユーザーの応答** [z/OS®: *MVS™ Programming: Authorized Assembler Services Guide*] で、戻りコード、エラー・コード、および情報コードについて参照してください。

---

**IDI0123S**Processing of abend *abend-code* terminated due to unsupported execution environment: *reason***説明**

*abend-code* に示された異常終了の分析前または分析中に、Fault Analyzer が、次のいずれかの理由で処理を続行できないと判断しました。

- **SUB=MSTR アドレス・スペース**アドレス・スペースは SUB=MSTR で開始されました。したがって SYSOUT の割り振りは許可されません。
- **全機能アドレス・スペースでない**現行アドレス・スペースは限定的な機能アドレス・スペースであることが分かりました。このアドレス・スペースはデータ・セットの割り振りなどの、必要なシステム・サービスをサポートしません。
- **システム・アドレス・スペース**開始タスクは既知のシステム・アドレス・スペースでした。Fault Analyzer の分析はこのアドレス・スペースに対して不適切であり、失敗する可能性があります。
- **システム・タスクの異常終了**異常終了が、IEFIIC イニシエーター TCB などのシステム・タスクの RB の下で発生しました。
- **JES が使用不可**基本サブシステム (JES) が使用不可であることが検出されました。
- **Fault Analyzer が JES シャットダウン時に起動された**Fault Analyzer は、異常終了の処理中に JES が使用不可になったことを検出しました。
- **動的割り振りが使用不可**現行タスクに対しては、必要な動的割り振りサービス (SVC 99) は使用不可であることが検出されました。
- **所有ジョブ・ステップ TCB が異常終了中**現行 TCB がジョブ・ステップ TCB ではなく、これを所有するジョブ・ステップ TCB が終了中の場合、ロードの障害が発生する可能性があります。
- **リソース・マネージャーでの異常終了**リソース・マネージャーは、現行 TCB に対してアクティブでした。
- **アドレス・スペース内の同時分析**システム・リソースの消費を防ぐために、与えられた 1 つのアドレス・スペースでは一時点で 1 つの TCB のみが Fault Analyzer の異常終了分析の実行を許可されます。現行異常終了に対して Fault Analyzer が起動されたとき、別の Fault Analyzer 異常終了分析がすでにアクティブであると判断されました。現行の分析はキャンセルされます。
- **TCB により SYSZTIOT がエンキュー済み***tcb-address* Fault Analyzer が大分類名 SYSZTIOT で異常終了の分析に呼び出されましたが、この名前は既に *tcb-address* 内のアドレスで識別される TCB によってキュー済みでした。この状態が発生する例は、Fault Analyzer が U0002 異常終了で呼び出されたときに IMS™ がシャットダウン中の場合です。
- **TCB の保護キーが 8 でも 9 でもない**Fault Analyzer が、保護キー 8 (非 CICS) でも 9 (CICS®) でも実行されていないタスクの異常終了を分析するために呼び出されました。

**システムアクション** 処理は終了します。

**ユーザーの応答** このような条件のもとでは、リアルタイム分析は可能ではありません。Fault Analyzer は、MSGID=IDI0123S および A=SVCD パラメーターを使用して SLIP トラップを設定し、次に Fault Analyzer ISPF インターフェイス File メニューのオプション 5 を使用した結果の SVC ダンプを分析することにより、依然として問題判別の補助に使用されることもあります。

#### IDI0124E

IDIS subsystem subtask *server-id resource-id* has terminated with abend code *abend-code*

**説明** Fault Analyzer サブシステム内のサブタスクが異常終了しました。

**システムアクション** Fault Analyzer サブシステム・サブタスクは終了しますが、必要に応じて再始動されます。リクエスト処理は続行しますが、失敗したサブシステム要求のタイプによっては予期しない結果になる可能性があります。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

#### IDI0125W

IDIS subsystem subtask *server-id resource-id* has terminated with return code *rc*

**説明** Fault Analyzer IDIS サブシステム内のサブタスクが異常終了しました。

**システムアクション** IDIS サブシステム・サブタスクは終了しますが、必要に応じて再始動されます。リクエスト処理は続行しますが、失敗したサブシステム要求のタイプによっては予期しない結果になる可能性があります。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

**IDI0126I**Recovery fault recording fault ID *faultid* assigned in history file *history-file-name*

**説明** Fault Analyzerは、通常のリアルタイム分析を完了することができませんでしたが、リカバリー障害記録障害項目 *faultid* をヒストリー・ファイル *history-file-name* に作成しました。この障害項目の再分析を行うと、Fault Analyzer が終了時に分析していた障害の情報を入手できます。

**システムアクション** 処理が終了しました。

**ユーザーの応答** なし

---

**IDI0127W**リカバリー障害記録が *job-id* に対して失敗しました。 *reason***説明**

*job-id* に示されたジョブに対する Recovery fault recording 処理が、以下のいずれかとして *reason* に示された理由で失敗しました。

- **SDUMP suppressed, capture phase of another SVC dump was in progress.** SDUMP が、戻りコード 8 および理由コード 2 で失敗しました。これは、SVC ダンプが、別の SVC ダンプの取り込みフェーズが進行中のため、抑止されたことを示しています。
- **SDUMP DASD space or overload error.** SDUMP が、戻りコード 8 および理由コード 3E で失敗しました。これは、SVC ダンプが、他のダンプを処理するために既に最大量の仮想ストレージ (CHNGDUMP コマンドの MAXSPACE パラメーターを使用して、インストール済み環境によって定義) を使用していることを示しています。一般には、この状態になる理由は、DASD スペースの不足です。
- **SDUMP rc=return-code reason=reason-code error.** SDUMP が、戻りコード *return-code* および理由コード *reason-code* で失敗しました。 [z/OS®: MVS™ Programming: Authorized Assembler Services Reference, Volume 3 (LLACOPY-SDUMPX)] で、これらのコードの説明を参照してください。
- **RFR requires the IDIS subsystem to be functioning correctly.** リカバリー障害記録が通常では実行される状態が発生しました。しかし、IDIS サブシステムが正しく機能していなかったため、リカバリー障害記録を実行できませんでした。

**システムアクション** 終了を続行します。

**ユーザーの応答** 示された理由に従って、適切なアクションを実行してください。リカバリー障害記録を使用可能にするためには、IDIS サブシステムが始動しており、機能している必要があります。

---

**IDI0129W**Recovery fault recording IEATDUMP failure, rc=*return-code* reason=*reason-code* dump=*data-set-name*

**説明** リカバリー障害記録処理時に IEATDUMP データ・セットを書き込もうとしたときに、エラーが発生しました。このメッセージ・テキストの *return-code* と *reason-code* は、それぞれ IEATDUMP サービスからの戻りコードと理由コードで、*data-set-name* は書き込もうとしたダンプ・データ・セットの名前です。この問題の説明については、 [z/OS®: MVS™ Programming: Assembler Services Reference] の IEATDUMP 戻りコードおよび理由コードに関するセクションを参照してください。

**システムアクション** 終了を続行します。

**ユーザーの応答** 該当するアクションを行って、IEATDUMP エラーを解決してください。

---

**IDI0130E**

IDIS サブシステム *task-id1 task-id2* からの応答が 2 分以内に返されなかったため、要求がキャンセルされました

**説明**

**システムアクション** 処理は続行されますが、取り消されたサブシステム要求によって予期された結果が影響されることがあります。

**ユーザーの応答** 発行されている可能性がある他のメッセージを確認して、IDIS サブシステムが応答しない理由を判別してください。また、IDIS サブシステムの優先順位は、このサブシステムが呼び出される可能性のあるどの異常終了するタスクよりも低く設定しないでください。引き続き応答が欠落し、理由が判別できない場合は、IBM サポートに連絡してください。

---

**IDI0131W**

Waiting *mins* minutes for *dsn(mbr)* SPFEDIT ENQ *owner*

**説明** *dsn* および *mbr* により識別されたヒストリー・ファイル・データ・セット・メンバーへの排他的アクセスの取得での問題がありました。それまで待機した時間は、*mins* 値に分数で示されます。可能な場合、ENQ の現在の所有者が *owner* に表示されます。例: "held by IDIS on SYS8".

**システムアクション** IDIS サブシステムは、要求されたデータ・セット・アクセスの待機を続行します。

**ユーザーの応答** 示されたデータ・セット・メンバーが更新に使用できない理由を判別してください。考えられる原因は、TSO/ISPF ユーザーがメンバーを編集していることです。

---

**IDI0132W**

MaxWaitSeconds of *seconds* exceeded for transaction *transaction* (task *task*), analysis will be skipped

**説明** DeferredReport(CICS(FATasks(...))) オプションに有効な MaxWaitSeconds 値を超過しました。このオプションの詳細については、[DeferredReport ページ 584](#)を参照してください。

**システムアクション** 示された CICS® トランザクションの分析は実行されません。

**ユーザーの応答** なし。

---

**IDI0133W**

DeferredReport option overridden due to MaxMinidumpPages(*max\_pages*) exceeded by *num\_pages* pages

**説明** XDUMP は有効ではなく、DeferredReport オプションは有効ですが、MaxMinidumpPages オプションの限度を超えました。

**システムアクション** ミニダンプは障害項目に書き込まれませんでした。分析は実行され、レポートが障害項目と IDIREPRT の両方に書き込まれました。

**ユーザーの応答**

このような状況を防ぐには、以下のようなことが役立つ可能性があります。

- XDUMP データ・セットの使用を有効にします。詳細については、[拡張ミニダンプ・データ・セット \(XDUMP\) ページ 59](#)を参照してください。
- MaxMinidumpPages オプション設定値を十分な大きさにします。

---

**IDI0134E**

*size k* の 24 ビットの LSQA ストレージが現在使用できないため、Fault Analyzer プロセスが除外されました

**説明** *size* (キロバイト単位) で示されたストレージの最小必要量が、境界以下の LSQA に使用できません。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

**IDI0135E**

Recovery fault recording terminating. Severe private storage shortage and no SVC dump access.

**説明** IEATDUMP ダンプ・タイプを使用してリカバリー障害記録処理を実行するための十分なストレージが専用領域で使用できません。また、アクセス権限が不十分なため、Fault Analyzer は SDUMP ダンプ・タイプを使用できません。SDUMP は、IDIS サブシステムから書き込まれます。したがって、通常は、異常終了した領域に重大なストレージの不足がある場合に使用できます。これで、IEATDUMP が書き込まれるのを防止します。

**システムアクション** リカバリー障害記録の障害項目なしで処理は終了します。

**ユーザーの応答** 可能な場合には、より大きな領域サイズで再実行依頼するか、または代わりに、SDUMP ダンプ・タイプを使用するのに必要なアクセス権限を提供します ([SDUMP リカバリー障害記録データ・セット ページ 311](#) を参照)。

---

**IDI0136W**

Recovery fault recording IEATDUMP not taken because NULLFILE has been selected for the DSN

**説明** Recovery fault recording 適格なデータ・セット名が判別されなかったため、処理で IEATDUMP データ・セットを書き込みできませんでした。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

**IDI0137W**

I/O Error

**説明** データ・セットの入出力操作中にエラーが発生しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** このエラーに関連した追加メッセージがあるかどうか調べてください。

---

**IDI0138S**

No minidump or MVS™ dump data set is available for reanalysis of history file *hist-file* fault ID *fault-id*

**説明** バッチ再分析を試みた障害項目には、ミニダンプもなく、既存の MVS™ ダンプ・データ・セットにも関連付けられていませんでした。それらの一方または両方がなければ、再分析は不可能です。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

**IDI0140S**

Processing terminated due to data set open error for DDname *ddname*

**説明** 必須 *ddname* のデータ・セットのオープンが失敗しました。通常、このメッセージの前に1つ以上の別のメッセージがあり、エラーに関してさらに詳細な情報を提供します (例えば、メッセージ [IDI0006E ページ 696](#))。

**システムアクション** 処理は終了します。

**ユーザーの応答** 示された *ddname* に関連するオプションの指定を確認してください。

---

**IDI0141W**

IDIS サブシステムには MODIFY-STOP を使用してください

**説明** このメッセージは、IDIS サブシステムを停止するために CANCEL コマンドを使用した場合に発行されます。

IDIS サブシステムを停止する正しい方法は、次のように、MODIFY または STOP コマンドを使用することです。

```
F name,STOP
```

または

```
P name
```

**システムアクション** IDIS サブシステムは停止します。

**ユーザーの応答** なし。

---

**IDI0142W**

Dispatch delay in IDIS subsystem (Priority=*idis-priority*) exceeded *num* seconds for *jobname job-id* (Priority=*job-priority*)

**説明** このメッセージは、IDIS サブシステムの応答時間が、示されているリクエスターの所定の値を超えた場合に発行されます。

**システムアクション** 処理を続行します。

**ユーザーの応答** IDIS サブシステムのディスパッチ優先順位を検討して、Fault Analyzer サブシステム・サービスを要求する、*jobname* および *job-id* により識別されるジョブの優先順位より低くないことを確認してください。

---

**IDI0143W**

Binder processing stopped because *num* k of 24 bit LSQA storage is not currently available

**説明** バインド・プログラムを起動するために使用できる 24 ビット LSQA ストレージが、十分ではありませんでした。

**システムアクション** 処理は続行されますが、分析レポートからソース行情報が欠落している場合があります。

**ユーザーの応答** なし。

---

**IDI0144E**

IDIS subsystem TCB *tcb-address detection-location* abended *abend-code*

**説明** IDIS サブシステムの TCB または機能が、異常終了したかエラー条件を検出しました。

**システムアクション** 処理は続行されますが、分析レポートから情報が欠落している場合があります。

#### ユーザーの応答

問題が解決しない場合は、IBM サポートに連絡してください。修正が利用可能でない場合は、以下のように SLIP トラップを設定し、分析用のダンプを提供してください。

```
SL SET, ID=xxxx, MSGID=IDI0144E, A=SVCD, END
```

#### IDI0145I

*message-text*

**説明** このメッセージは、IDIS サブシステムが発行するすべての状況メッセージに使用されます。

これらのメッセージの例には、次のものがあります。

```
IDI0145I IDISXCFA TCB XCF startup
IDI0145I IDIS subsystem, IDISMAIN Started. V9R1M0
(MVS 2009/02/03)
IDI0145I Starting Termination.
```

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

#### IDI0146I

IDIS サブシステムのストレージ使用量は、JCL REGION サイズの *percent-value%* です。 *number-of-tasks* タスクを実行しています

**説明** このメッセージは、使用されたストレージの量が、JCL REGION パラメーターに指定された、またはデフォルトで設定された使用可能な最大ストレージの 80% を超えるたびに、IDIS サブシステムによって発行されます。このメッセージは、ストレージ使用量が高い間は、間隔を置いて再発行されますが、ストレージ使用量が再度 80% を下回ると発行されません。

**システムアクション** 処理を続行します。

**ユーザーの応答** このメッセージが定期的に発行される場合には、使用可能なストレージの不足によって IDIS サブシステムが終了するのを回避するため、領域サイズを増大します。

#### IDI0147I

IDI\_SDUMP\_ACCESSSDUMP には XFACILIT Fault Analyzer への ALTER アクセス権が必要

**説明** このメッセージは、RFR ダンプが書き込まれ、異常終了したジョブに関連付けられているユーザー ID に XFACILIT IDIS リソース・クラスに対する ALTER アクセス権がないと判断されるたびに、IDI\_SDUMP\_ACCESS サブシステムによって発行されます。

**システムアクション** RFR ダンプを SDUMP ではなく IEATDUMP として書き込むように試行されます。

**ユーザーの応答** SDUMP を使用してリカバリー障害記録のパフォーマンスを改善する方法については、[SDUMP リカバリー障害記録データ・セット ページ 311](#)を参照してください。



**IDI0148E**

IDIS サブシステム NOUPDINDEX オプションが有効であるために、RFR 障害エントリーを作成できません。

**説明** このメッセージは、RFR ダンプが書き込まれようとしているときに、IDIS サブシステムが NOUPDINDEX オプションで起動されていた場合、そのサブシステムによって出されます。

**システムアクション** この問題の SDUMP はとられますが、履歴ファイルの障害エントリーには記録されません。

**ユーザーの応答** 該当する場合、SDUMP の分析を実行し、問題を識別してください。UPDINDEX オプションを有効にした状態でサブシステムを再始動してください。

**IDI0149W**

IDIMAPS *dsname* には build YYMMDD=*build-date* がありますが、必要なレベルは *required-date* です。Execution may be incorrect.

**説明** 不正なバージョンの IDIMAPS データ・セットが使用されています。*dsname* で示されるデータ・セットは、*build-date* に示される日付に作成されていますが、この日付は、*required-date* 内の Fault Analyzer のインストール・レベルに対して必要な日付に一致しません。

**システムアクション** 処理は続行しますが、結果は正しくない可能性があります。

**ユーザーの応答** IDIMAPS DDname に対して指定したデータ・セット名が正しく (この名前は通常、IDICNFxx parmlib メンバーの DataSets オプションを使用して指定されます)、またこのデータ・セットに SMP/E ターゲット・ライブラリーからの現行データが含まれるようにします。

**IDI0150W**

No READ access to DDname *ddname* data set name *dsname*. This data set will not be used.

**説明** 明示的または暗黙的に Fault Analyzer に対して提供されたデータ・セットは、セキュリティー・サーバーの READ アクセス権がないため、アクセス不能でした。

**システムアクション** 処理は続行しますが、データ・セットがクリティカルな場合には、エラーが発生する可能性があります。

**ユーザーの応答** データ・セットに対する適切なアクセス権限を提供します。

**IDI0151W**

SDUMP failure *reason* [Job *job\_id*]

**説明**

*reason* で特定された理由で Recovery fault recording 処理または Java™ 分析が失敗しました。関連するジョブ ID が使用可能な場合、そのジョブ ID は *job\_id* としてメッセージに含まれます。例:

- SDUMP rc=8 rsn=2 for job *jobname*. SVC dump suppressed because capture phase of another SVC dump was in progress.
- SDUMP rc=8 rsn=3E for job *jobname*. DUMPSERV has used virtual storage MAXSPACE because of dump DASD space shortage or high activity.
- SDUMP rc=0 rsn=04 for job *jobname*. DUMPSERV は部分的なダンプしか作成できませんでした。Examine associated IEA\* DUMP messages for more detail.

- **SDUMP rc=return-code rsn=reason-code for job jobname.**
- **SDUMP requires the IDIS subsystem to be functioning.**

SDUMP の戻りコードと理由コードについては、メッセージ [IDI0127W ページ 724](#) を参照してください。  
このメッセージは、IDIS サブシステムによって発行されます。

**システムアクション** IDIS サブシステム処理を続行します。

**ユーザーの応答** 示された理由に従って、適切なアクションを実行してください。

---

#### IDI0152I

*jobname(history-file)* にジョブ *fault-id* SDUMP が要求されました

**説明** SDUMP リカバリー障害記録ダンプ・データ・セットが、*jobname* に示されるジョブ用に要求されましたが、これは、ヒストリー・ファイル *fault-id* 内で障害 ID *history-file* に関連付けられます。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

#### IDI0153I

Binder processing terminated for member *module-name* because it was created with the LINK=NO option

**説明** ロード・モジュール *module-name* に関するバインド・プログラムの呼び出しが、リンク・エディット・オプション LINK=NO を使用したことにより rc=83000505 で失敗しました。

**システムアクション** 示されたロード・モジュールに関するバインド・プログラム情報なしで処理を続行します。

**ユーザーの応答** なし。

---

#### IDI0154W

Configuration-options module *module-name* found in non-authorized load library and has been ignored

**説明** Fault Analyzer の構成オプションの提供に使用できる、*module-name* で示されているロード・モジュールが、APF 許可のないロード・ライブラリー内で検出されました。ロード・モジュール名は、IPVOPTLM または IDIOPTLM のいずれかです。このロード・モジュールを使用するためには、APF 許可ロード・ライブラリー内に配置する必要があるため、これは無視されます。

**システムアクション** 処理を続行します。

**ユーザーの応答** ロード・モジュールを APF 許可のあるロード・ライブラリーに置いてください。

---

#### IDI0155W

IDIS サブシステムに使用するユーザー ID には、HOME パスを使用する OMVS セグメントがありません

**説明** IDIJLIB DDname が IDIS サブシステム JCL に指定されていない場合、Fault Analyzer は、代わりに、作業ファイル用に IDIS サブシステムのユーザー ID OMVS セグメント HOME パスを使用します。この場合、IDIJLIB DDname および HOME パスは使用できません。

**システムアクション** 処理は続行しますが、Java™ 分析サポートは使用されません。

**ユーザーの応答** IDIJLIB DDname が IDIS サブシステム JCL に指定されているか、または IDIS サブシステムのユーザー ID 用に OMVS セグメント HOME パスが存在することを確認してください。

---

#### IDI0156W

GETMAIN of *count* bytes from *jobname job-id* address space failed *abend-code*, *idis-module-name history-file-name* IDIS subsystem request failed

**説明** IDIS サブシステムがデータを返していたときに、*count* バイトのクロス・メモリー GETMAIN が失敗し、異常終了 *abend-code* が出されました。この異常終了は、*jobname* と *job-id* によって識別されたリクエスター・アドレス・スペースのストレージ不足が原因で発生しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** 可能であれば、リクエスター・アドレス・スペースの領域サイズを増やし、今後この問題が起こらないようにしてください。

---

#### IDI0157I

Fault Analyzer が意図的に異常終了 (U0777) して、IDIRFRON DDname のために RFR ダンプを生成しようとしています

**説明** IDIRFRON DD ステートメントが使用されている場合、Fault Analyzer は意図的に異常終了 U0777 を発行して、リカバリー障害記録障害項目が作成されるようにします。IDIRFRON DDname の使用の詳細については、[リカバリー障害記録セットアップの検証 ページ 438](#)を参照してください。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし。

---

#### IDI0158W

IDIS subsystem requires restart with STEPLIB containing SMP/E \*.SIDIAUT2 to load DLL *dll-name*

**説明** Java™ 障害を分析しようとしたのですが、IDI.SIDIAUT2 サブシステム STEPLIB 連結内にデータ・セット IDIS がないために失敗しました。

**システムアクション** 分析を続行しますが、Java™ 情報は欠落しています。

**ユーザーの応答** IDI.SIDIAUT2 を IDIS サブシステム STEPLIB 連結に追加します。詳しくは、[IDIS サブシステムの始動 ページ 321](#)を参照してください。

---

#### IDI0159I

SDUMP requested for *job-id1 fault-id* will not be taken because *reason* usage of SDUMP

#### 説明

スケジュールされた Fault Analyzer リカバリー障害記録 SDUMP (SVC ダンプ) のレートが、これを処理するオペレーティング・システムの能力を超えるため、結果として現行ダンプは取得されませんでした。メッセージ・テキストで、*job-id1* は SDUMP を要求した JES のジョブ ID であり、*reason* は次のいずれかです。

- 「Current *job-id1*」、現行ジョブの SDUMP レートのしきい値が超過している場合。
- 「*job-id2*」、別のジョブ (*job-id2*) の SDUMP しきい値が超過している場合。
- 「High system」、すべてのジョブのしきい値を結合したものが超過している場合。

**システムアクション** リカバリー障害記録 SDUMP は取得されません。

**ユーザーの応答** IBM サポートに連絡して、高レートの Fault Analyzer リカバリー障害記録 SDUMP が要求された理由を特定します。このメッセージが出される直前に取得された SDUMP の例を分析のために提供してください。

---

#### IDI0160I

History file *history-file* I/O error recovery successful

**説明** メッセージ [IDI0033E ページ 703](#) で示される入出力エラーの後、Fault Analyzer は *history-file* で示されるヒストリー・ファイル内に十分なスペースを取り戻すことができ、以降の障害項目の再書き込みができるようになりました。

**システムアクション** 処理は正常に続行します。

**ユーザーの応答** なし。

---

#### IDI0161W

ヒストリー・ファイル *history-file* I/O エラー・リカバリーが失敗しました。 *reason*

##### 説明

*history-file* で示されるヒストリー・ファイルで、入出力エラーからのリカバリーが試行されましたが、*reason* で示される理由により失敗しました。

- **History file is not a PDSE PDS** ヒストリー・ファイルに対する入出力エラー・リカバリーは使用できません。
- **History file contains 25 or less fault entries** 自動スペース管理に関しては、ヒストリー・ファイルの障害項目が 25 個より多い場合のみ、障害項目は暗黙的に削除されます。
- **IGWFAMS error** エラーの詳細情報を含むメッセージ [IDI0095W ページ 716](#) が、このメッセージの直前に出されません。
- **Unable to provide the required space** 障害項目をこれ以上削除できませんでしたが、必要なスペース容量はまだ使用可能になりませんでした。この必要なスペースの欠如は、障害項目がロックされていることが原因の場合があります。
- **Recursive I/O error** 前の入出力エラーのリカバリー中に入出力エラーが発生しました。

**システムアクション** 処理は続行しますが、障害項目はヒストリー・ファイルに書き込まれません。

##### ユーザーの応答

次の 1 つ以上が適切な場合があります。

- 履歴ファイルにスペースを追加して、再割り振りしてください。
  - IDIUTIL SetMinFaultEntries 制御ステートメントを使用して、ヒストリー・ファイルのスペース管理設定を現在の設定より少ない値で変更します。
  - 暗黙的な削除を可能にするために、ロックされている障害項目をアンロックしてください。
- 

#### IDI0162I

MVS™ dump taken to extract Java™ information

**説明** MVS™ 情報の非同期抽出を実行するために、現行の障害に対する Java™ ダンプが取得されました。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0164I**

Fault ID *fault-id* created in history file *history-file* due to *reason*

**説明** *reason*が原因で、MVS™ ダンプ・データ・セットのバッチ分析により、ヒストリー・ファイル *history-file* に新規の障害項目 *fault-id* が作成されました。

*reason* の考えられる値は次のとおりです。

- GenerateSavedReport オプション
- IDIRegisterFaultEntry コマンド

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0165A**

*ddname* updates available--restart IDIS subsystem with //IDIDOC2 DD statement for IDI.SIDIDOC2 data set

**説明** IDIS サブシステムを開始するときに、IDIVSENU、IDIVSJPN、または IDIVSKOR として *ddname* で示された VSAM KSDS メッセージおよび異常終了コード説明リポジトリに対して 1 つ以上の更新が使用可能であることが判明しました。しかし、更新を含む //IDIDOC2 データ・セットを識別するための IDI.SIDIDOC2 DD ステートメントが指定されていません。

**システムアクション** VSAM KSDS を更新せずに処理を続行します。

**ユーザーの応答**

1. VSAM KSDS の更新を含む IDIS データ・セットに対して次のように DD ステートメントを IDI.SIDIDOC2 サブシステム JCL で追加します。

```
//IDIDOC2 DD DISP=SHR,DSN=IDI.SIDIDOC2
```

2. IDIS サブシステムを再始動します。
- 

**IDI0166E**

*ddname* 更新 *member* の処理中にエラーが発生しました: *reason*

**説明**

VSAM KSDS メッセージおよび異常終了コード説明リポジトリを更新するときにエラーが発生しました。メッセージ・テキスト内の各項目の意味は以下のとおりです。

***ddname***

DDname を IDIVSENU、IDIVSJPN、IDIVSKOR のいずれかとし特定します。

***member***

データ・セット IDI.SIDIDOC2 内の更新メンバーの名前。

***reason***

発生したエラーの説明。

**システムアクション** 処理は続行されますが、VSAM KSDS の更新は完了せず、IDIS サブシステムが次回開始されたときに VSAM KSDS の更新が再試行されます。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

**IDI0167I**

CICS® *release* のモデル処理が *count* オフセットを返しました

**説明** CICS® では、Fault Analyzer に特定リリース依存フィールド・オフセット値が必要となります。このオフセットは、Fault Analyzer がインストールされている場合に CICS® の始動時に判別されます。このメッセージは、検出されたオフセットの数を示しています。

**システムアクション** 処理を続行します。

**ユーザーの応答** なし。

---

**IDI0168E**

Models processing for CICS® *release* returned 0 offsets

**説明** CICS® では、Fault Analyzer に特定リリース依存フィールド・オフセット値が必要となります。このメッセージは、オフセット情報を判別する処理が失敗した場合に発行されます。この障害は、CICS® 環境での Fault Analyzer 操作を妨げません。

**システムアクション** CICS® では、オフセット情報がないと Fault Analyzer は動作できません。

**ユーザーの応答** オフセット情報は SIDIMAPS データ・セット・メンバー IDIMCICS から抽出されます。IDIMAPS DataSets オプションが正しく指定されていることを確認してください。問題が解決しない場合は、このメッセージ番号に SLIP トラップを設定して、IBM サポートに連絡してください。

---

**IDI0169E**

Module *module-name* loaded from *library-name* is not APF authorized or the concatenation is not APF authorized

**説明** *module-name* で示された Fault Analyzer ロード・モジュールは APF 許可ロード・ライブラリーから実行する必要があります。今回、ロード・モジュールは *library-name* で示されたロード・ライブラリーから実行されましたが、このライブラリーは APF 許可されていなかったか、あるいは APF 許可されていない1つ以上の他のロード・ライブラリーとともに JOBLIB または STEPLIB 連結に組み込まれていました。

**システムアクション** 処理は終了します。

**ユーザーの応答** *library-name* が APF 許可されていること、および APF 許可されていない他のロード・ライブラリーとの連結に組み込まれていないことを確認してください。

---

**IDI0170W**

Unable to update *data-set-name* due to no UPDATE access

**説明** このメッセージは、*data-set-name* によって示されたメッセージおよび異常終了コード説明リポジトリ・データ・セットに対して更新が使用可能であるが、このデータ・セットに対する UPDATE アクセス権限が Fault Analyzer サブシステムにない場合に、IDIS IDIS サブシステムによって発行されます。この更新によって、既存の説明に含まれるフォーマット問題やタイプミスが修正されたり、新規のメッセージや異常終了コードが追加されたりする可能性があります。

**システムアクション** 更新は実行されずに処理が続行されます。

**ユーザーの応答** 更新を実行するには、最初に IDIS サブシステムによって *data-set-name* に対する UPDATE アクセス権限を付与し、その後で IDIS サブシステムを停止して再始動してください。

---

**IDI0171W**

*count* PDSE クロスシステム SHARING 競合を *history-file-name* 分待機しています

**説明** このメッセージは、ヒストリー・ファイルにアクセスするためにかかった時間が1分を超える場合に Fault Analyzer IDIS システムによって発行され、その後、ヒストリー・ファイルにアクセスできるようになるまで1分ごとに発行されません。

**システムアクション** ヒストリー・ファイルにアクセスできるようになるまでヒストリー・ファイルの処理は中断されません。

**ユーザーの応答** 問題が解決しない場合は、SPFEDIT 大分類名を使用して、誰がヒストリー・ファイルの ENQ を保持しているかを判別してください。

---

**IDI0172I**

Fault Analyzer NameToken anchor built at *storage-addr* by TCB *tcb-addr*

**説明** このメッセージは、CICS® での Fault Analyzer 初期化時に発行されます。その目的は、エラーの発生時に IBM による診断を補助することです。

**システムアクション** 処理は正常に続行します。

**ユーザーの応答** なし。

---

**IDI0173I**

Fault Analyzer system DUMP call already issued for Fault Entry *fault-id*

**説明** Recovery Fault Recording (RFR) 処理の間に、*fault-id* で識別されている障害エントリーが、それ自体に対して既にシステム・ダンプ呼び出しを出していることが分かりました。そのため、これ以上のシステム・ダンプ呼び出しは出されません。

**システムアクション** 処理は正常に続行します。

**ユーザーの応答** なし。

---

**IDI0174I**

Fault Analyzer Java DUMP fault entry written to *history-file1(fault-id)* because IDIS subsystem cannot write to *history-file2*

**説明** Fault Analyzer IDIS サブシステムは、アクセス権限が十分ではないために、Java™ ダンプ障害項目をヒストリー・ファイル *history-file2* に書き込むことができませんでした。代わりに、障害エントリー *fault-id* がデフォルト履歴ファイル *history-file1* に書き込まれました。

**システムアクション** 処理は正常に続行します。

**ユーザーの応答** なし。

---

**IDI0175I**

Fault Analyzer DUMP 分析は、*dump-dsn(history-file)* に *fault-id* を使用します

**説明** このメッセージは、*dump-dsn* で識別されているダンプ・データ・セットが、*history-file(fault-id)* で識別されている障害項目に属していることがわかったときに、IDIS サブシステムによって出されます。この障害エントリは更新され、関連付けられているダンプ・データ・セット名が追加されます。

**システムアクション** 処理は正常に続行します。

**ユーザーの応答** なし。

---

#### IDI0177E

Java DTFJ 処理が *history-file (fault-id)* に対して失敗しました

**説明** 障害エントリ Java™ の再分析の間に、予期されていた *fault-id* 情報が履歴ファイル *history-file* に見つかりませんでした。このメッセージは、Java DTFJ 処理が正常に完了しなかったことを示しています。

**システムアクション** 処理は通常どおりに続行しますが、Java 情報はありません。

**ユーザーの応答** DTFJ 処理が失敗した理由を調べてください。MVS™ ダンプ後出口 IDIXTSEL がインストールされていること (詳しくは、[MVS ポスト・ダンプ出口のインストール IDIXTSEL ページ 418](#)を参照)、また IDIS サブシステムが始動していること (詳しくは、[Fault Analyzer IDIS サブシステムの使用 ページ 319](#)を参照)を確認してください。

---

#### IDI0178E

Fault Analyzer 開始がハングしました。RFR ダンプが異常終了の取り込みを開始しました。

**説明** リアルタイム分析を実行しようとして試みているときに、ハングが検出されました。

**システムアクション** リアルタイム分析は終了し、代わりにリカバリー障害記録障害エントリが書き込まれます。

**ユーザーの応答** 問題が解決しない場合は、IBM サポートに連絡してください。

---

#### IDI0179W

*side-file-type* サイド・ファイルの使用を *program-name* に対して強制しようとして試みているときに処理エラーが発生しました。このサイド・ファイルは使用されません。

**説明** [Listing/Side File Mismatch] プロンプトで、ENTER キー・アクションによってサイド・ファイルの処理が正常に行われませんでした。

**システムアクション** プログラムのコンパイラー・リスト/サイド・ファイル・サポートなしで、処理は続行します。

**ユーザーの応答** 必要に応じて、コンパイル日付とリストされている理由で不一致を調べて、よりプログラムに一致するコンパイラー・リストまたはサイド・ファイルを見つけてください。

---

#### IDI0180I

Fault Analyzer CICS DUMPTABLEEXCLUDE( CheckMaxCurr) のために処理がスキップされました。異常終了コード *abend-code*。

**説明** CICS® トランザクション・ダンプ・テーブル・エントリが *abend-code* に存在していて、現在のダンプ・カウン트가最大設定を超えています。CICS DUMPTABLEEXCLUDE(CheckMaxCurr) オプションが指定されているために、Fault Analyzer 分析はスキップされます。

**システムアクション** 処理は終了します。

**ユーザーの応答** なし



---

**IDI0181W**

IDIS は停止し、ストレージを保護するためのタスクを再始動します。REGION サイズを大きくして再始動してください。

**説明** Fault Analyzer は、サブシステムでストレージ不足状態を検出しました。サブシステムは停止し、必要な時点で IDIS DB2@ サブタスクを再始動して状態の緩和を試みます。

**システムアクション** 処理を続行します。

**ユーザーの応答** このメッセージが IDIS サブシステムによって定期的に行われる場合には、使用可能なストレージの不足によって IDIS サブシステムが終了するのを回避するため、領域サイズを増大することをお勧めします。

---

**IDI0182I**

Fault Analyzer subsystem error from IDISAREQ: *error\_message*

**説明** 予期しない条件が原因で、メッセージが Fault Analyzer IDIS サブシステムから戻されました。*error\_message* に入っている戻されたメッセージには、一般に、別のメッセージ ID が含まれています。

**システムアクション** 処理を続行します。

**ユーザーの応答** Fault Analyzer IDIS サブシステム呼び出しで発生した問題について、*error\_message* にあるメッセージ ID を調べてください。

---

**IDI0183W**

No trace records found in *data\_set\_name*

**説明** *data\_set\_name* にある補助 CICS トレース・データ・セットのマッピングが要求されました。

**システムアクション** トレース・マッピングは終了します。

**ユーザーの応答** 指定されたデータ・セットが正しいことを確認してください。

---

**IDI0184W**

拡張ミニダンプ・データ・セット *data\_set\_name* を割り当てることができません: *reason*

**説明** *data\_set\_name* で示されている拡張ミニダンプ・データ・セットを割り当てようと試みましたが、*reason* で示されている理由で失敗しました。

**システムアクション** 処理は続行しますが、障害エントリーに拡張ミニダンプ・データ・セットが関連付けられることはありません。XDUMP ページを追加した後のミニダンプ・ページの総数が MaxMinidumpPages オプションを実際に超えない限り、Below-the-bar XDUMP ストレージ・ページはミニダンプ・ページに変換されます。その場合、XDUMP データ・セットを宛先としていたすべてのストレージ・ページは破棄されます。

**ユーザーの応答** メッセージに示されている理由に対処してください。

---

**IDI0187I**

関連付けられたダンプ・データ・セット *data\_set\_name* は、不適切なアクセス許可のため、削除されません。ダンプ・データ・セット・タイプ: *dump-type*、ヒストリー・ファイル *hist-file*、障害 ID: *fault-id*、ジョブ名により作成: *jobname*、ユーザー ID: *user-id*、セキュリティー・サーバー・デフォルト・グループ ID *group-id*

**説明** リアルタイム分析時、対話式 MVS ダンプ分析に続く障害項目作成時、または IDIUTIL バッチ・ユーティリティー IMPORT 処理時に、自動ヒストリー・ファイル・スペース管理により、障害項目が削除されました。関連するデータ・セットの削除も試みられました。しかし、アクセス許可が不十分であったため、*data-set-name* に示された関連するダンプ・データ・セットは削除できませんでした。

**システムアクション** 処理を続行します。

**ユーザーの応答** 示されたダンプ・データ・セットを手動で削除します。このメッセージで提供される情報から、このタイプのダンプ・データ・セットへのセキュリティー・サーバー・アクセス許可を変更して、それらデータ・セットとそれに関連する障害項目の自動削除を許すべきかどうかを判別してください。

*dump-type* に応じた追加情報:

- RFR TDUMP および RFR SVC ダンプ。参照: [リカバリー障害記録データ・セット・アクセスの管理 ページ 310](#)
- SVC ダンプ・コピー。参照: [コピーされた SDUMP データ・セット・アクセスの管理 ページ 315](#)
- XDUMP。参照: [XDUMP データ・セット・アクセスの管理 ページ 313](#)

### IDI0188I

SNAPDATA parameter ignored: *text*

**説明** IDISNAP SNAPDATA パラメーターでエラーが検出されました。*text* には、エラーについての追加情報が含まれていません。

**システムアクション** 処理を続行します。

**ユーザーの応答** 考えられる説明について *text* を参照してから、問題を修正します。

### IDI0189W

XDUMP データ・セット *data\_set\_name* は関連する障害項目と切り離して分析できません。その代わりに、ヒストリー・ファイル *history\_file\_name* 障害 ID *fault\_id* の再分析を実行します。

**説明** *data\_set\_name* がダンプ・データ・セットとして分析用に選択されました。例えば、Fault Entry List 画面から「File」メニュー・オプション 5 を使用するなど。データ・セットは XDUMP データ・セットであるため、この方法で分析することはできません。

**システムアクション** ダンプの分析が終了します。

**ユーザーの応答** その代わりに、*history\_file\_name* および *fault\_id* のヒストリー・ファイルおよび障害項目の再分析を実行します。

### IDI0191W

ヒストリー・ファイル *history-file* は、2 次スペースの割り振りなしで作成されました。

**説明** IDIS サブシステムは、*history-file* データ・セットが 2 次スペースなしで割り振られていることを検出しました。

**システムアクション** 処理を続行します。

**ユーザーの応答** Fault Analyzer スペース管理は、2 次スペースを割り振ると効率的に動作するように設計されています。[ヒストリー・ファイルの 2 次スペースの割り振り ページ 346](#) を参照し、[PDSE ヒストリー・ファイルのサイズの変更 ページ 349](#) の説明に従ってヒストリー・ファイルの再割り振りを検討してください。

**IDI0192I**

Dump data set *data\_set\_name* not deleted *timestamp*: *reason*

**説明** Fault Analyzer は、障害項目に関連付けられた *data\_set\_name* ダンプ・データ・セットを削除できませんでした。*timestamp* は、関連する障害項目が削除された時刻を示します。この時点で、*data\_set\_name* ダンプ・データ・セットがマイグレーションされ、非同期プロセスがそのダンプ・データ・セットを削除しようとしていました。*reason* は削除が失敗した理由を示します。

このメッセージは、削除できなかったダンプ・データ・セットごとに 1 回発行されます。メッセージが出される時刻は、*reason* によって異なり、以下のいずれかの時点で発生する可能性があります。

- 関連付けられた障害項目が削除された時点。
- IDIS サブシステムが、削除されていないダンプ・データ・セットを定期的に検査する場合。
- IDIS サブシステムを停止する場合。

**システムアクション** 処理を続行します。

**ユーザーの応答** 手動または自動操作によって、ダンプ・データ・セットを削除します。これを実行しないと、未使用のデータ・セットが孤立状態になります。削除が失敗した理由については、*reason* を参照するか、または SYSLOG を確認して、*timestamp* に示されている時間内に発行された可能性がある他のメッセージがないかどうかを調べてください。

**IDI0193E**

ロック・フラグ制御出口ロード・モジュール IDIXLOCK のエラー: *reason*

**説明** ロック・フラグ制御出口 IDIXLOCK がインストールされている状態で、*reason* が原因で障害項目ロック・フラグの変更が試行されました。*reason* の考えられる値は次のとおりです。

```
Execution abend abend_code
未許可のロード・ライブラリーで検出
LOAD 異常終了 abend_code-reason_code
LOAD error R15=load_reason_code
CSVQUERY error R15=csvquery_reason_code
```

**システムアクション** 処理は続行されますが、ロック・フラグは変更されません。

**ユーザーの応答** システム・プログラマーに連絡してください。

**IDI0194W**

*spilled\_pages* minidump pages spilled to XDUMP data set *xdump\_dsn* due to MaxMinidumpPages(*mmp\_pages*) exceeded

**説明** 有効な MaxMinidumpPages オプション値 (*mmp\_pages*) を超えました。XDUMP データ・セットが使用可能になっているため、MaxMinidumpPages オプション値 (*spilled\_pages*) を超えた数のミニダンプ・ページが XDUMP データ・セット (*xdump\_dsn*) に「スピル」されました。XDUMP データ・セットが存在する間、これは障害項目の再分析中に透過的に処理されます。ただし、XDUMP データ・セットが使用可能でなくなった場合 (削除された、名前が変更された、または再分析が実行されるシステムで使用できないなど)、再分析レポート情報が欠落するか、正しくなくなる可能性があります。この場合、「B」または「I」の行コマンドを使用したとき、または保管レポートがない障害項目に対して「V」または「S」の行コマンドを使用したときに、Fault Analyzer ISPF インターフェースによって警告が表示されます。同様の警告がバッチ再分析レポートおよび対話式再分析レポートに挿入されます。XDUMP が欠落している障害項目のバッチ再分析を実行すると、メッセージ **IDI0195W ページ 740** も発行されます。このタイプの状態を回避するには、以下のようになります。

1. XDUMP など、密結合の関連ダンプ・データ・セットを削除したり名前を変更したりしないでください。Fault Analyzer を使用して障害項目が削除されると、これらは自動的に削除されます。
2. ミニダンプ・ページを XDUMP データ・セットにスプールの必要がないようにするには、MaxMinidumpPages オプション値で十分であることを確認してください。

**システムアクション** 処理を続行します。

**ユーザーの応答** システム・プログラマーに連絡してください。

---

#### IDI0195W

スプールされたミニダンプ・ページを含む障害項目 *fault\_id* XDUMP データ・セット *xdump\_dsn* が存在しません。再分析レポート情報が欠落しているか、正しくない可能性があります

**説明** このメッセージは、障害項目 (*fault\_id*) のバッチ再分析中に発行され、作成時にメッセージ IDI0194W が発行されました (この問題が発生した条件については、「[IDI0194W ページ 739](#)」を参照してください)。関連する XDUMP データ・セット (*xdump\_dsn*) が使用できなくなった (削除された、名前が変更された、または再分析が実行されるシステムで使用できないなど) ので、再分析レポート情報が欠落するか、正しくなくなる可能性があります。追加情報については、「[メッセージ IDI0194W ページ 739](#)」を参照してください。

**システムアクション** 処理は続行されますが、再分析レポート情報が欠落するか、正しくなくなる可能性があります。

**ユーザーの応答** システム・プログラマーに連絡してください。

---

#### IDI0196W

Unable to perform reanalysis of fault entry *fault\_id* in history file *history\_file* due to no READ access.

**説明** このメッセージは、ユーザーが READ アクセス権を持っていない障害項目のバッチ再分析が試行された場合に発行されます。

**システムアクション** 処理は終了します。

**ユーザーの応答** アクセスが必要な場合には、セキュリティー管理者に連絡してください。

## Appendix A. カスタマイズ済み ISPF インターフェース・フロントエンドの例

特定の環境では、Fault Analyzer ISPF インターフェースが呼び出されたときに表示される 初期 Fault Entry List 画面を、動的に調整できることが望ましい場合があります。例えば、使用される障害履歴・ファイルまたはビュー名を事前選択する場合、または動的に作成された MATCH コマンドを提供する場合などです。MATCH コマンドは、例えば当日の日付や特定の PROGRAM 名に一致させるために、使用できます。

この動的調整の実行方法を示す例が、サンプル・データ・セット (IDI.SIDISAM1) に組み込まれています。この例では、ポップアップ・パネルが表示されるので、ユーザーが、オプションのプログラム名およびアプリケーション ID、または「ビュー」名を指定できます (図 340: サンプル画面 1 ページ 742 を参照してください)。「?」を「アプリケーション/ビュー (Application/View)」フィールドに配置すると、使用可能なアプリケーションおよびビューのリストを表示できます (図 341: サンプル画面 2 ページ 742 を参照してください)。サンプルでは、アプリケーション/ビュー ID の長さが 2 である場合、障害履歴・ファイルの名前を書式設定するために、選択された名前が次のように使用されます。

```
<Variable DSNp1>.<System ID>.HIST.<Variable DSNp2>.<Application>
```

長さが 2 でない場合は、ID が Fault Analyzer VIEW の名前であると想定されます。

アプリケーション/ビュー ID が正常に入力され、その存在が確認されたら、ユーザーは Enter を押して、Fault Analyzer を呼び出すことができます。プログラム名も提供された場合は、対応する MATCH コマンドも作成されます。

サンプルは次のファイルで構成されます。各メンバーは、テーブルに示される ddname に連結されているデータ・セットにコピーされます。

表 28. IDI.SIDISAM1 データ・セットのサンプル・メンバー

ファイル	DDname	説明
IDISFEMA	SYSPROC	Main REXX exec
IDISFESK	ISPSLIB	履歴・ファイルの作成に使用する ISPF スケルトン
IDISFECL	SYSPROC	Fault Analyzer 呼び出し時に使用される仲介 CLIST
IDISFEAP	ISPLLIB	アプリケーション選択に使用される ISPF パネル
IDISFEQP	ISPLLIB	照会 ISPF パネル
IDISFEMP	ISPLLIB	ユーザー・パラメータ提供のためのメイン ISPF パネル

図 340. サンプル画面 1

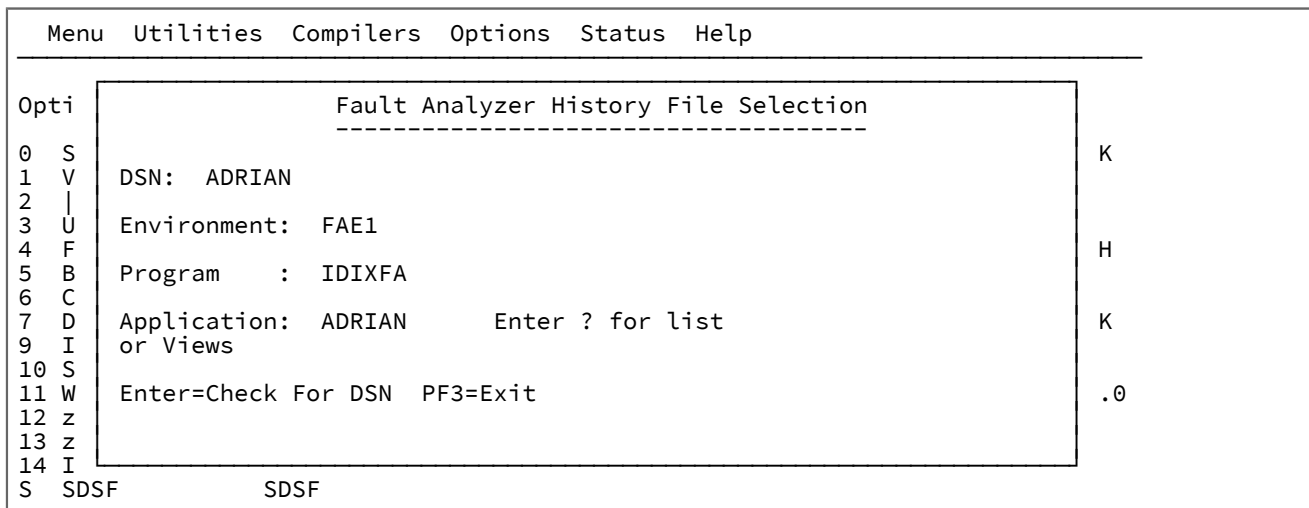
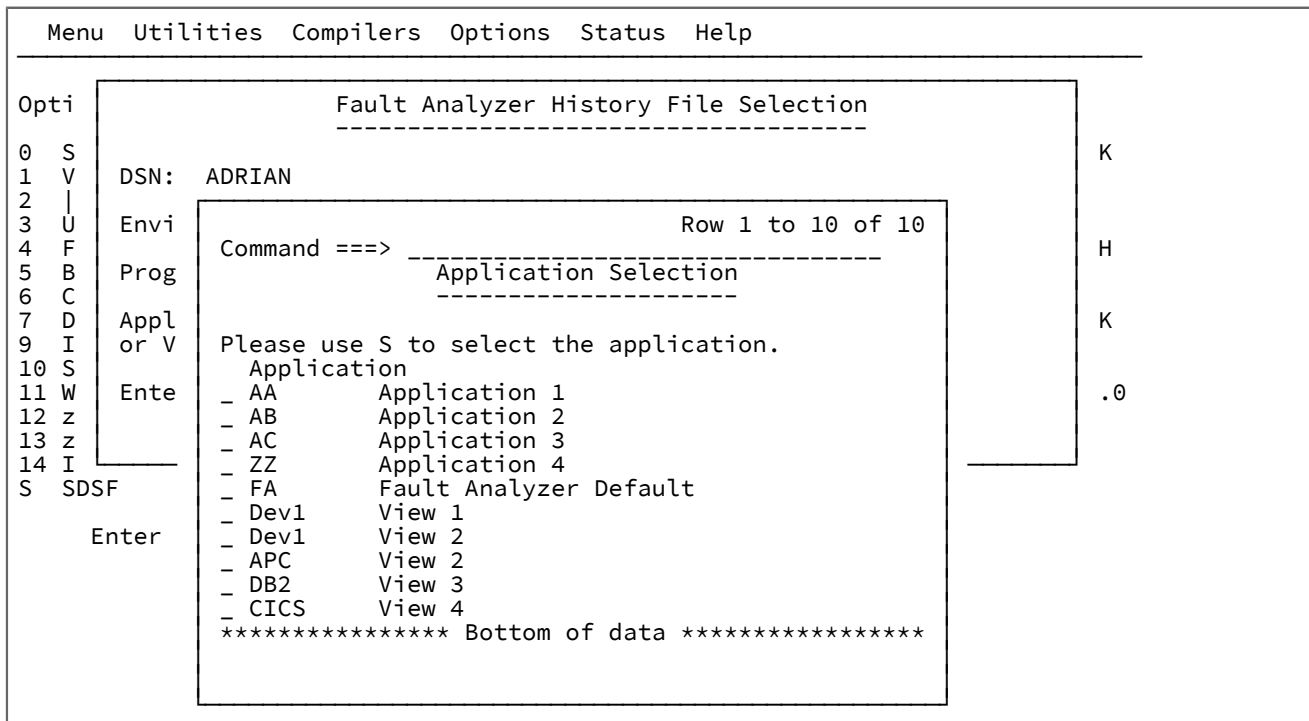


図 341. サンプル画面 2



障害ヒストリー・ファイルまたはビュー名が選択および確認されたら、サンプル・コードによって次の処理が行われます。

## 障害ヒストリー・ファイルまたはビュー名

Fault Analyzer ISPF アプリケーション (IDI) の変数プールには、OLDHIST という名前の変数があります。これは、最後にアクセスした 10 個の障害ヒストリー・ファイルまたはビューのリストです。このリストの最初の項目は、ヒストリー・ファイルまたはビューで、Fault Analyzer ISPF インターフェースで開かれます。サンプル・コードによって、このリストの最初の項目が、ユーザーが入力したヒストリー・ファイルまたはビューになるように、リストが変更されます。この変更は、リストをスキャンし、名前がすでに存在しているかどうかを確認することによって行われます。存在する場合、項目はリ

ストの先頭に移動されます。名前が存在しない場合は、リストの一番上に挿入され、他のすべての項目は1項目ずつ下へ移動されます。そのため、10番目の項目があった場合は、除去されます。

リストではビュー名は括弧で囲まれるので、障害履歴・ファイルと区別できます。

## プログラム名の MATCH

プログラム名が入力されると、次のコマンド・ストリングが作成されます。

```
MATCH PROGRAM <supplied program name>
```

プログラム名が提供されなかった場合は、MATCH ALL コマンドが作成されます。

このコマンド・ストリングは、Fault Analyzer ISPF インターフェースが開始されると選出され、それに応じて実行されません。

## サンプル・アプリケーションのインストール

サンプル・アプリケーション (IDISFEMA) は、メイン Fault Analyzer ISPF アプリケーションと同じ ISPF アプリケーション ID を使用して実行されることが重要です。このように実行されない場合、サンプルによって行われる、ISPF 変数に対する更新内容は、正しく選出されません。サンプルの使用方法として、Fault Analyzer の通常の呼び出しに使用されるとの同じ ISPF NEWAPPL アプリケーションを使用して、メイン・サンプル・プログラムを呼び出す新規 ISPF コマンドを追加することができます。例えば、次のコマンドを [USER] コマンド・テーブルに追加します。これは、Fault Analyzer ISPF ユーザーが使用します。

```
Verb      T  Action
FASEL     0  SELECT CMD(%IDISFEMA &ZPARM) NEWAPPL (IDI)
```

このコマンドでは、Fault Analyzer が通常は IDI の NEWAPPL を使用して呼び出されることを前提としています。例えば、アプリケーション選択パネルから次のようなコマンドを使用します。

```
9,PGM(IDIPDDIR) NEWAPPL (IDI) SCRNAME (FAULTA)
```

上記のように、CLIST IDISFECL は、ユーザーの SYSPROC 連結に割り振られるいずれかのデータ・セットに含まれている必要があります。

## サンプルの機能方法

MATCH コマンド・ストリングが Fault Analyzer ISPF インターフェースによって受け渡されて実行できるように、仲介 CLIST が使用されます (IDISFECL)。この CLIST は、ISPF DISPLAY PANEL コマンドの COMMAND オプションで呼び出し コマンドを引き渡す方法で、呼び出されます。DISPLAY PANEL コマンドで COMMAND オプションを指定すると、参照される実際の PANEL は表示されず、COMMAND オプション内のコマンドが実行されます。例えば、プログラム名 MYPROG1 が入力されると、次のコマンド・ストリングが作成されます。

```
TSO IDISFECL; ;MATCH PROGRAM MYPROG1
```

このコマンド・ストリングは、サンプル・コードの変数 CMDSTACK に割り当てられ、次のように ISPF DISPLAY コマンドで参照されます。

```
Address ISPEXEC 'DISPLAY PANEL (IDISFEAP) COMMAND (CMDSTACK)'
```

## Appendix B. Fault Analyzer レポートをダウンロードするための Java API

この Java™ API は、Fault Analyzer レポートをホストからダウンロードし、Java™ で開発された Eclipse ベースの製品で表示します。

プロセスのダウンロード部分は、Fault Analyzer レポートがまだダウンロードされていない場合にのみ実行されます。この API は FAAPI パッケージに含まれており、このパッケージは Eclipse 製品の Fault Analyzer プラグインの一部として入手できます。指定されたホスト ID とポート番号が「システム情報」ビューで定義済みであることを前提としています。

### 指定

この機能を利用するために 2 つのインターフェースが設計されています。

```
FAAPI.openReport(String UniqueAddressOfFaultEntry);
//UniqueAddressOfFaultEntry : "hostId/portNo/historyFile/faultId"

FAAPI.openReport(String hostId, int portNo, String historyFile, String faultId);
```

どちらのインターフェースも同じ機能を実行しますが、引数の指定のスタイルのみが異なります。引数はすべて、大/小文字が区別されます。

次に示す例では、両方のインターフェースを使用して、ポート 7799 で listen するシステム pthfae1 上のヒストリー・ファイル DA.DCAT 内の障害項目 F03004 を開きます。

```
FAAPI.openReport("pthfae1/7799/DA.DCAT/F03004");

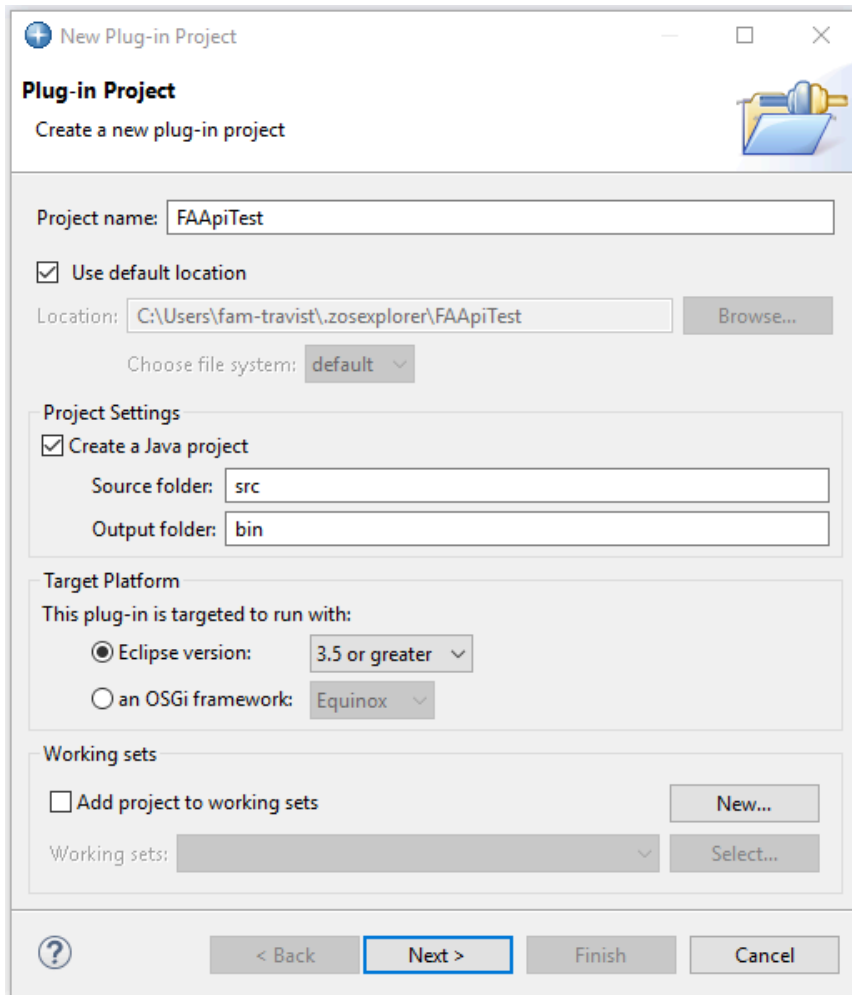
FAAPI.openReport("pthfae1", 7799, "DA.DCAT", "F03004");
```

### 例

この例は、Fault Analyzer レポートを開くために「Hello, World Command」サンプル・プラグインを Eclipse で拡張する方法を示しています。以下の手順に従って、「Hello, World Command」サンプル・プラグインを作成し、Fault Analyzer レポートをダウンロードして開くようにこのプラグインを変更します。

1. IBM® Explorer for z/OS® の場合、the Java™ パースペクティブをアクティブ化します。
2. ADFz ファミリー製品用のプラグインをアクティブ化します。
3. 「New」->] **Plug-in Project** をクリックします。





4. プロジェクト名(このサンプルでは FAApiTest)を入力して「Next」をクリックします。

**New Plug-in Project**

**Content**  
Enter the data required to generate the plug-in.

**Properties**

ID:

Version:

Name:

Vendor:

Execution Environment:

**Options**

Generate an activator, a Java class that controls the plug-in's life cycle  
Activator:

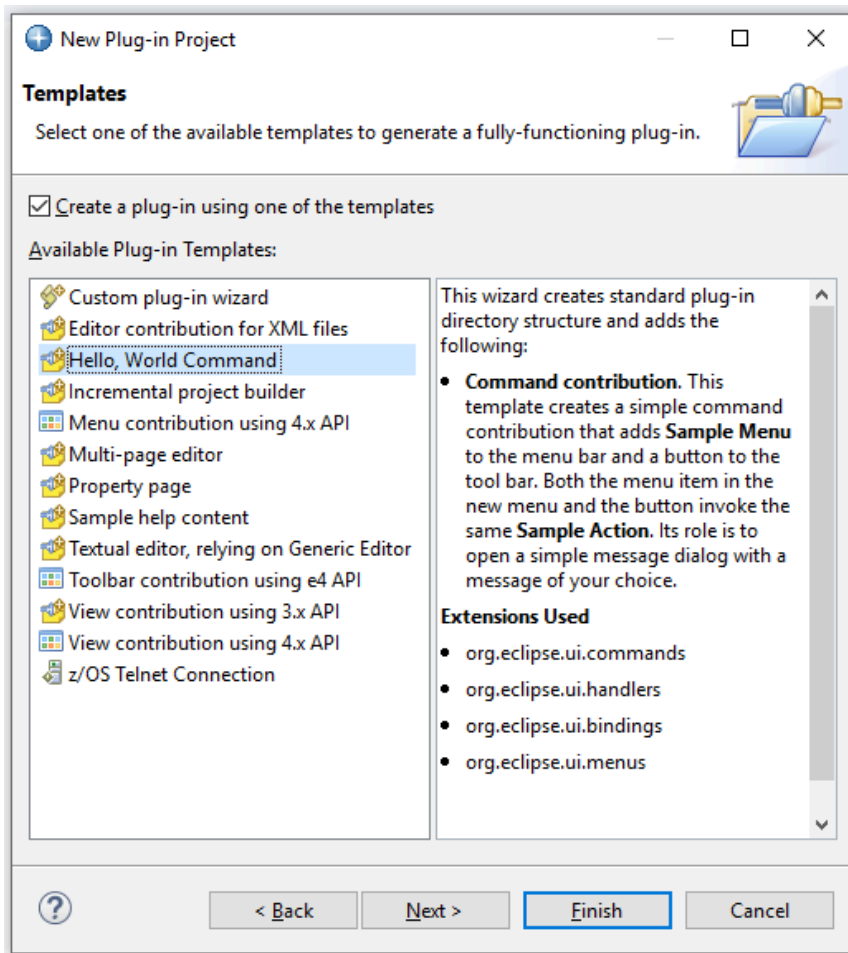
This plug-in will make contributions to the UI

Enable API analysis

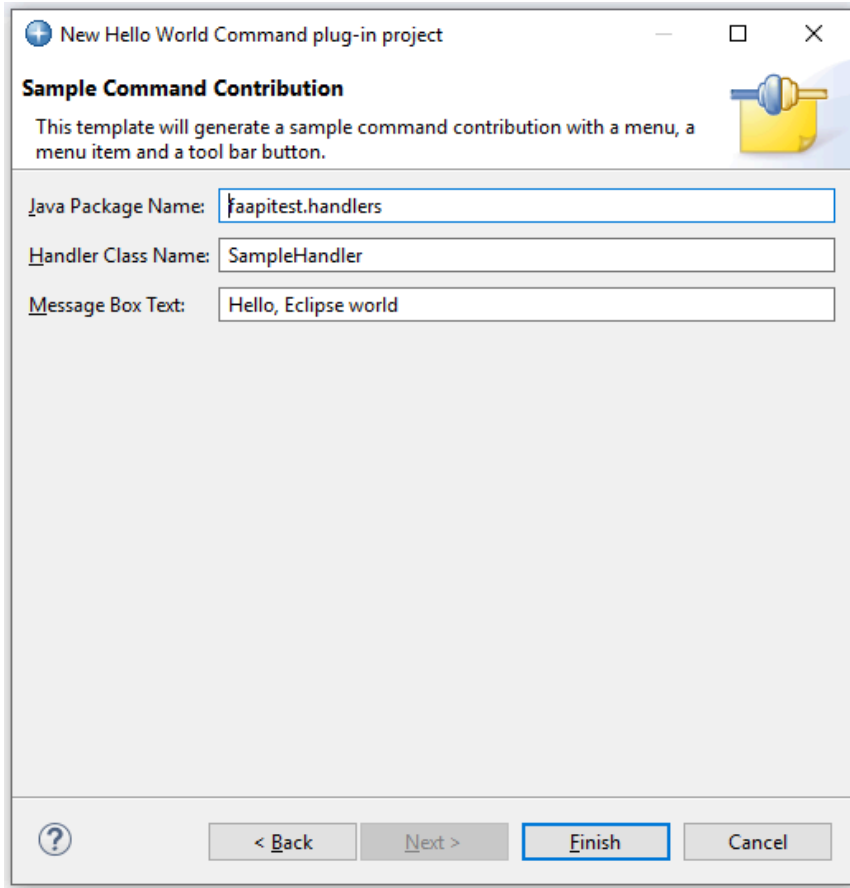
**Rich Client Application**

Would you like to create a rich client application?  Yes  No

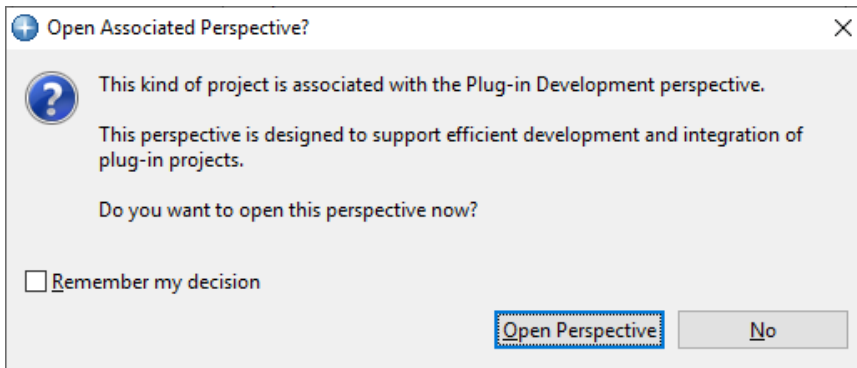
5. **[Next]** をクリックします。



6. 「Hello, World Command」テンプレートを選択し、「Next」をクリックします。

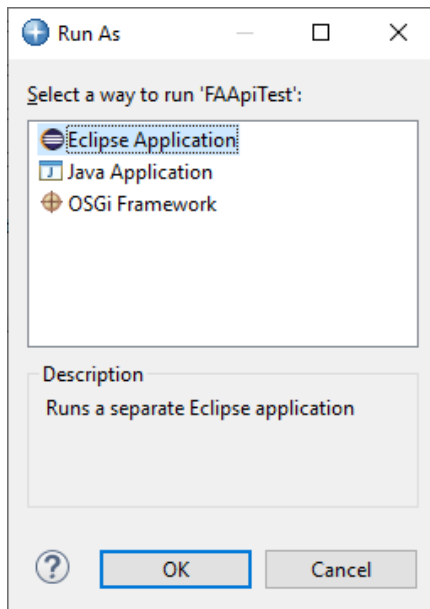


7. **Finish** をクリックします。次のメッセージが表示されたら、**Yes** をクリックします。

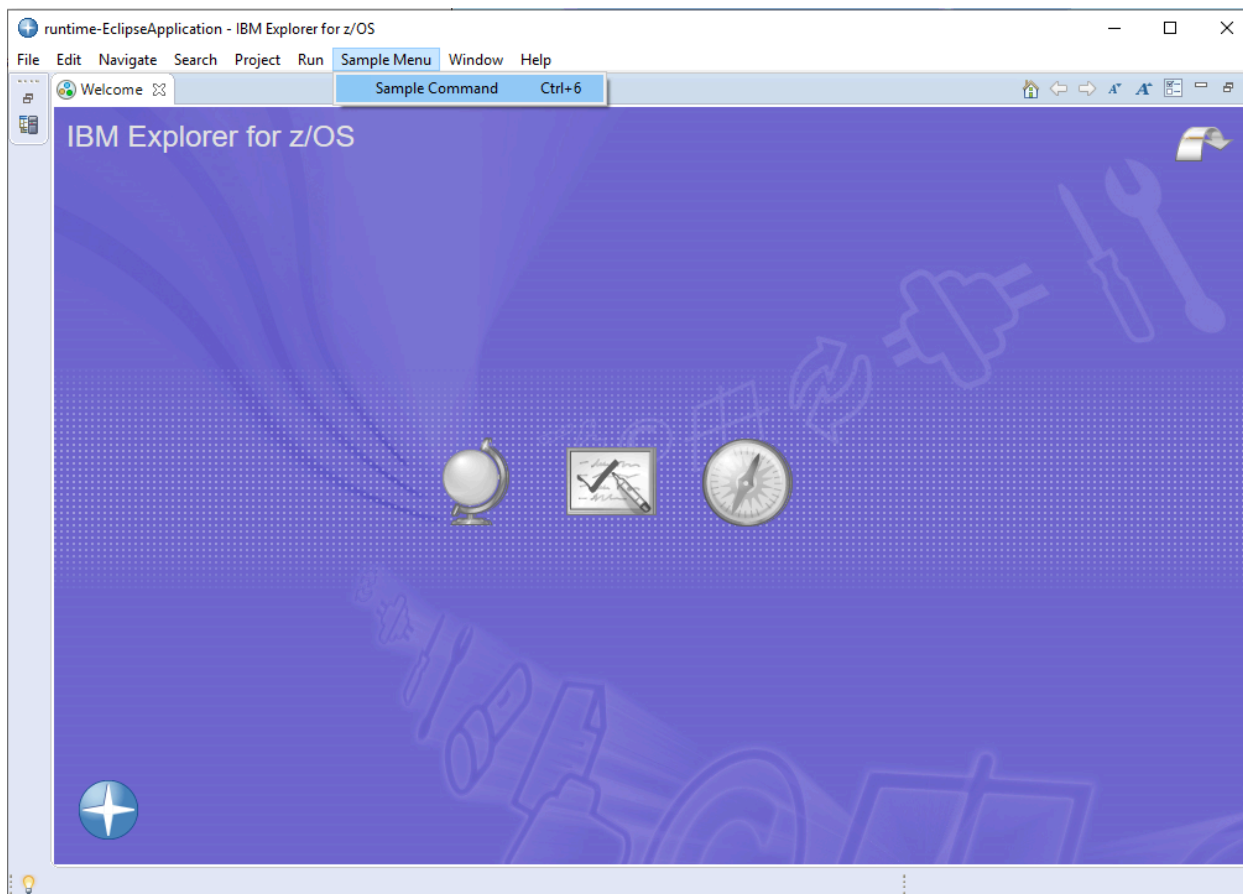


この段階で、FAAPITest が作成されます。生成されたプラグインをこの段階で実行するには、以下の手順に従ってください。

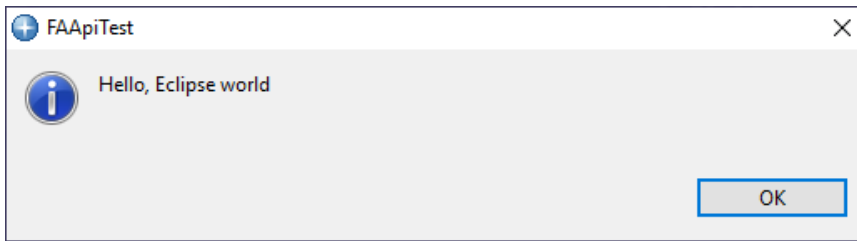
1. Package Explorer で「FAAPITest」をクリックします。
2. メインメニューで「実行」をクリックして、「Run」を選択します。
3. 「Eclipse Application」を選択して、「OK」をクリックします。



4. 実行中の環境で、次のウィンドウが表示されます。「Sample Menu > Sample Command」をクリックします。



5. 次のメッセージが画面に表示されます。



6. 「OK」をクリックして、実行中の Eclipse 環境を閉じます。

ここで、「Sample Menu」->「Sample Command」をクリックすると、障害項目が表示されるように FAPIITest プラグインが変更されます。

1. MANIFEST.MF を開きます。このファイルは、FAAPITest プラグインの META-INF フォルダにあります。
2. 「Dependencies」タブをクリックします。次に、必要なプラグインの「Add...」をクリックして「com.ibm.etools.fa.pdtclient.ui」を選択します。「OK」をクリックして MANIFEST ファイルを保管します。
3. `faapitest.handlers.SampleHandler.java` を開いて、次の例に示すように run メソッドを変更します。

```
package faapitest.handlers;

import org.eclipse.core.commands.AbstractHandler;
import org.eclipse.core.commands.ExecutionEvent;
import org.eclipse.core.commands.ExecutionException;
import org.eclipse.ui.IWorkbenchWindow;
import org.eclipse.ui.handlers.HandlerUtil;
import org.eclipse.jface.dialogs.MessageDialog;
import com.ibm.etools.fa.pdtclient.ui.util.FAAPI;

public class SampleHandler extends AbstractHandler {

    @Override
    public Object execute(ExecutionEvent event) throws ExecutionException {
        IWorkbenchWindow window = HandlerUtil.getActiveWorkbenchWindowChecked(event);
        MessageDialog.openInformation(
            window.getShell(),
            "FAApiTest",
            "Hello, Eclipse world");
        return null;
    }
}
```

4. 変更したプラグインを前述のように実行します。
5. Eclipse を初期化したら、システムが「システム情報」ビューで定義されていることを確認します。定義されていない場合は、システムを追加します。
6. 「**Sample Menu**」->「**Sample Command**」をクリックします。レポートがダウンロードされ、Eclipse エディター領域の現行パースペクティブに表示されます。



## Appendix C. 画面サイズ調整の技術的詳細

幅広い端末特性をサポートするには、IBM® 提供 MODETAB の ISTINCLM で、DLOGMOD 指定の D4C32XX3 を使用します。この詳細について、および Telnet サーバー要件については、z/OS® Communications Server の資料に記載があります。

IBM® パーソナル・コミュニケーションズのユーザーは [IC71220: PCOM: HOW CAN YOU USE A 62X160 SCREEN SIZE?](#) を IBM® サポート・ポータルで参照する必要があります。

# Appendix D. サポート・リソース

これらのリソースを使用して、製品の詳細、フィックス、サポートを検索できます。

## 知識ベースの検索

- プログラム・ディレクトリーをダウンロードします。
  - [Application Delivery Foundation for z/OS Common Components](#) のプログラム・ディレクトリー
  - [IBM File Manager for z/OS](#) のプログラム・ディレクトリー
  - [IBM Fault Analyzer for z/OS](#) のプログラム・ディレクトリー
- 以下の製品のインストール、カスタマイズ、使用に関する最新の詳細を入手できます。
  - [Application Delivery Foundationの z/OS Common Components カスタマイズ・ガイドおよびユーザーズ・ガイド](#)
  - [Fault Analyzer ユーザーズ・ガイドおよびリファレンス](#)
  - [File Managerカスタマイズ・ガイド](#)
  - [File Managerユーザーズ・ガイドおよびリファレンス](#)
  - [DB2のFile Manager ユーザーズ・ガイドおよびリファレンス](#)
  - [File Manager ユーザーズ・ガイドおよびリファレンス \(CICS 用\)](#)
  - [IMSのFile Managerユーザーズガイドおよびリファレンス](#)

## 最新の PTF の取得

- [ADFz Common Components](#)
- [Fault Analyzer for z/OS](#)
- [File Manager の z/OS](#)
- [z/OS Debugger](#)
- [IBM Developer for z/OS Enterprise Edition](#)
- [Application Performance Analyzer for z/OS](#)

## 診断データの収集

サポートに連絡する前に、これらの質問に答えられるようにしましょう:

- どのソフトウェア・バージョンを実行しているのか?
- 当該の問題に関連したログ、トレース、メッセージがあるか?
- 問題を再現できるかどうか。再現できる場合、どのように当該の問題を再現するのか?
- ハードウェア、オペレーティング・システム、ネットワークング・ソフトウェアを変更したか?
- 当該の問題の回避策はあるか?

## サポートへ連絡してください

<https://www.ibm.com/mysupport> でケースをオープンしたり、サポートとチャットしたり、リソースとコミュニティに接続したりすることができます。

## Appendix E. Fault Analyzer のアクセシビリティ機能

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つお客様が、IT 製品を快適に使用できるように支援します。

### 概要

Fault Analyzer には、以下の主要なアクセシビリティ機能が含まれています。

- キーボードだけを使用した操作
- スクリーン・リーダーを使用する操作

Fault Analyzer は IBM® z/OS® SMP/E を使用して製品をインストールします。IBM® z/OS® SMP/E のアクセシビリティ機能については、[https://www.ibm.com/support/knowledgecenter/en/SSLTBW\\_2.4.0/com.ibm.zaddinfor.doc/access.html](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zaddinfor.doc/access.html) で参照できます。

Fault Analyzer では、IBM® Explorer for z/OS® を使用して、オプションの Fault Analyzer Eclipse プラグインをインストールします。IBM® Explorer for z/OS® のアクセシビリティ機能については、[https://www.ibm.com/support/knowledgecenter/SSBDYH\\_3.2/com.ibm.zexpl.doc/accessibility.html](https://www.ibm.com/support/knowledgecenter/SSBDYH_3.2/com.ibm.zexpl.doc/accessibility.html) を参照してください。

Fault Analyzer は、最新の W3C 標準である WAI-ARIA 1.0 ([www.w3.org/TR/wai-aria/](http://www.w3.org/TR/wai-aria/)) を使用することにより、US Section 508 ([www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards](http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards)) への準拠を徹底しています。アクセシビリティ機能を利用するには、Fault Analyzer でサポートされるスクリーン・リーダーの最新リリースを使用します。

PDF ファイルのアクセシビリティ・サポートは制限付きです。PDF 文書では、オプションでフォントの拡大、ハイコントラスト表示設定を使用でき、キーボードだけでナビゲーションできます。

### キーボードによるナビゲーション

TSO/E または ISPF を使用した ISPF パネルのナビゲートの詳細は、[https://www.ibm.com/support/knowledgecenter/en/SSLTBW\\_2.4.0/com.ibm.zaddinfor.doc/access.html](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zaddinfor.doc/access.html) を参照してください。このトピックでは、キーボード・ショートカットおよびファンクション・キーの使用法を含む、インターフェースを介したナビゲーション方法について説明します。

### インターフェース情報

Fault Analyzer のユーザー・インターフェースには、毎秒 2 回から 55 回フラッシュするコンテンツは含まれません。

Fault Analyzer の Web ユーザー・インターフェースは、カスケーディング・スタイル・シートを使用することで、コンテンツを適切にレンダリングし、使いやすさを実現しています。このアプリケーションでは、低視力のユーザーがユーザーのシステム表示設定 (ハイコントラスト・モードを含む) を使用するために、これに相当する方法が提供されています。デバイスまたは Web ブラウザーの設定を使用して、フォント・サイズを制御できます。

Fault Analyzer Web ユーザー・インターフェースには WAI-ARIA ナビゲーション・ランドマークが含まれ、アプリケーションの機能領域にすばやく移動するために使用できます。

## 小数点付き 10 進数の構文図

スクリーン・リーダーを使用してオンライン製品資料にアクセスするユーザー用に、構文図は小数点付き 10 進数形式で提供されます。小数点付き 10 進数の形式で、構文のエレメントはそれぞれ別の行に書き込まれます。複数の構文エレメントが、必ず両方とも存在するか、あるいは必ず両方とも存在しないかのどちらかであれば、それらのエレメントは単一の複合構文エレメントと見なすことができるため、同じ行に表示されることがあります。

各行は、3、3.1、3.1.1 など、小数点付き 10 進数で始まります。これらの番号が正しく聞き取られるようにするために、句読点を読み上げるようにスクリーン・リーダーを設定してください。同じ小数点付き 10 進数を持つすべての構文エレメント (例えば、3.1 という番号を持つすべての構成エレメント) は、同時に複数選択することはできません。3.1 USERID と 3.1 SYSTEMID という行が読み上げられた場合、その構文には USERID か SYSTEMID のどちらか一方を含めることはできますが、両方を含めることはできません。

小数点付き 10 進数の番号付けのレベルは、ネスティングのレベルを表します。例えば、3 という小数点付き 10 進数を持つ構文エレメントの後ろに 3.1 という小数点付き 10 進数を持つ一連の構文エレメントが続いた場合、3.1 という数字の付いたすべての構文エレメントは、3 という数字の構文エレメントに従属しています。

構文エレメントに関する情報を付け加えるために、小数点付き 10 進数に続けて特定の語や記号が使用されます。それらの語および記号が、エレメント自体の先頭に現れることもあります。それらの語または記号が構文エレメントに含まれているものである場合には、識別しやすくするために、直前に円記号 (\) が付加されます。小数点付き 10 進数の後に \* 記号を付けて、その構文エレメントが繰り返されることを示すことができます。例えば、小数点付き 10 進数 3 を持つ構文エレメント \*FILE は 3 \\* FILE という形式で表されます。形式 3\* FILE は、構文エレメント FILE が繰り返されることを示します。形式 3\* \\* FILE は、構文エレメント \* FILE が繰り返されることを示します。

構文エレメントのストリングの分離に使用されるコンマなどの文字は、構文内でそれらの文字が分離する項目の直前に表示されます。これらの文字は、個々の項目と同一の行、または関連項目と同じ小数点付き 10 進数を持つ別の行に表示される場合があります。その行には、その構文エレメントに関する情報を示す別の記号も付けることができます。例えば、5.1\*、5.1 LASTRUN、および 5.1 DELETE という行は、LASTRUN 構文エレメントおよび DELETE 構文エレメントを複数使用する場合には、それらのエレメントをコンマで分離する必要があることを意味しています。分離文字が指定されていない場合には、各構文エレメントを分離するために空白が使用されているものと見なしてください。

構文エレメントの前に % 記号がある場合、この記号は、別の場所で定義されている参照を示します。% 記号の後に続くストリングは、リテラルではなく、構文の断片の名前です。例えば、行 2.1 %OP1 は、別の構文の断片 OP1 を参照する必要があることを意味します。

次の記号が、小数点付き 10 進数に続けて使用されます。

### ? はオプションの構文エレメントを示します

疑問符 (?) 記号は、オプションの構文エレメントを示します。小数点付き 10 進数に続く疑問符記号 (?)

は、対応する小数点付き 10

進数を持つすべての構文エレメント、およびすべての従属構文エレメントがオプションであることを示します。その小数点付き 10 進数を持つ構文エレメントが 1 つのみの場合、?

記号はその構文エレメントと同じ行に表示されます (例えば、5? NOTIFY

のようになります)。ある小数点付き 10 進数を持つ構文エレメントが複数ある場合は、?

記号は単独で 1 行に表示され、その後にオプションの構文エレメントが続きます。例えば、5 ?、5

NOTIFY という行と、5 UPDATE という行が読み上げられた場合は、構文エレメント NOTIFY および UPDATE はオプションであることがわかります。すなわち、いずれか 1 つを選択するか、いずれも選択しないことが可能です。? 記号は、構文図における迂回線に相当します。

#### ! はデフォルトの構文エレメントを示します

感嘆符 (!) 記号は、デフォルトの構文エレメントを示します。小数点付き 10 進数の後ろに ! 記号と 1 つの構文エレメントが続く場合、その構文エレメントが、同一の小数点付き 10 進数を共用するすべての構文エレメントのデフォルト・オプションであることを示します。! 記号を指定できるのは、その小数点付き 10 進数を共用する構文エレメントのうち、1 つのみです。例えば、行 2? FILE、2.1!(KEEP)、および 2.1 (DELETE) という行が読み上げられた場合には、(KEEP) が FILE キーワードのデフォルト・オプションであることがわかります。この例では、オプションを指定せずに FILE キーワードを含めると、デフォルト・オプションの KEEP が適用されます。デフォルト・オプションは、直上の小数点付き 10 進数にも適用されます。この例の場合、FILE キーワードを省略すると、デフォルトの FILE(KEEP) が使用されます。ただし、行 2? FILE、2.1、2.1.1!(KEEP) および 2.1.1 (DELETE) という行が読み上げられた場合には、デフォルト・オプション KEEP は、直上の小数点付き 10 進数である (関連するキーワードのない) 2.1 にのみ適用され、2? FILE には適用されません。キーワード FILE が省略された場合には何も使用されません。

#### \* は反復可能なオプションの構文エレメントを示します

アスタリスクまたは絵文字 (\*) 記号は、ゼロ回以上の繰り返し可能な構文エレメントを示します。小数点付き 10 進数の後ろに \* 記号が続く場合、この構文エレメントは 0 回以上繰り返して使用できます。つまり、その構文エレメントはオプションであり、かつ繰り返し可能です。例えば、5.1\* データ域という行が読み上げられた場合、データ域を 1 つ含める、複数のデータ域を含める、もしくは、データ域を含めない、という選択が可能であることがわかります。3\*、3 HOST、3 STATE という行が読み上げられた場合は、HOST、STATE、その両方、またはなし、という選択が可能です。

#### 注:

1. 小数点付き 10 進数に続けてアスタリスク (\*) があるが、その小数点付き 10 進数を持つ項目が 1 つしかない場合は、その項目を複数回繰り返すことができます。
2. 小数点付き 10 進数に続けてアスタリスクがあり、その小数点付き 10 進数を持つ項目が複数ある場合は、そのリストの中から複数の項目を使用できますが、各項目はそれぞれ 1 回しか使用できません。前出の例では、HOST STATE と記述することはできません、HOST HOST と記述することはできません。
3. \* 記号は構文図におけるループバック線に相当します。

#### + は含める必要がある構文エレメントを示します

プラス (+) 記号は、少なくとも 1 回含める必要がある構文エレメントを示します。小数点付き 10 進数の後ろに + 記号が続く場合、この構文エレメントを 1

回以上含めなければならないことを示しています。つまり、この構文エレメントを少なくとも 1 回含める必要があり、繰り返すことが可能です。例えば、6.1+ データ域という行が読み上げられた場合、少なくとも 1 つのデータ域を含める必要があります。2+、2 HOST、および 2 STATE という行が読み上げられた場合、HOST、STATE、または両方を含める必要があるということになります。\* 記号と同様に、+ 記号は特定の項目がその小数点付き 10 進数を持つ唯一の項目である場合に、その項目を繰り返すことができます。+ 記号は、\* 記号と同じく、構文図におけるループバック線に相当します。

## 関連するアクセシビリティ情報

標準の IBM® ヘルプデスクとサポート Web サイトに加え、IBM® は、聴覚が不自由なお客様が営業やサポートサービスにアクセスするために使用できる TTY 電話サービスを提供します。

TTY service 800-IBM®-3383 (800-426-3383) (within North America)

アクセシビリティへの IBM® の取り組みの詳細については、[IBM® Accessibility \(www.ibm.com/able\)](http://www.ibm.com/able) を参照してください。

# 通知

本書は米国 HCL 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。

ライセンス対象マテリアル - HCL Technologies Ltd. の所有物です。

© Copyright IBM Corporation 2000, 2016. © Copyright HCL Technologies Ltd. 2017, 2024. 米国政府の限定権利。IBM Corp との GSA ADP Schedule Contract によって使用、複製、または公開が制限されています。

IBM® では、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およびサービスについては、日本の IBM® 担当者にお問い合わせください。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。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、プログラム、またはサービスの操作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。ライセンスについてのお問い合わせは、書面にて下記宛に送ることができます。

*IBM Director of Licensing*  
*IBM Corporation*  
*North Castle Drive, MD-NC119*  
*Armonk, NY 10504-1785*  
*US*

2 バイト文字セット (DBCS) 情報についてのライセンスに関するお問い合わせは、お住まいの国の IBM® Intellectual Property Department に連絡するか、書面にて下記宛先にお送りください。

*Intellectual Property Licensing*  
*Legal and Intellectual Property Law*  
*IBM Japan Ltd.*  
*19-21, Nihonbashi-Hakozakicho, Chuo-ku*  
*Tokyo 103-8510, Japan*

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、お客様自身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、自ら適切と信じる方法で、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。

本書に記載されているパフォーマンス・データは、特定の動作および環境条件下で得られたものです。実際の結果は異なる場合があります。

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、IBM 以外の製品に関するパフォーマンス、互換性、またはその他の要求の精度については確認できません。IBM 以外の製品の機能に関する質問は、それらの製品の供給者にお問い合わせください。

Fault Analyzer for z/OS® の将来の方向性、または目的に関する記述については、予告なしに変更または撤回される場合があります。単に目標や目的を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれています。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

#### 著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれらを複製し、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暗示したり、保証したりできません。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わないものとします

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的創作物にも、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。

© (your company name) (year).

Portions of this code are derived from IBM Corp. and/or

HCL Technologies Ltd. sample programs.

© Copyright IBM Corp. 2000, 2016. © Copyright HCL Technologies Ltd. 2017, 2024.

## プログラミング・インターフェース情報

この「ユーザーズ・ガイドおよびリファレンス」には、プログラムを作成するユーザーが Fault Analyzer のサービスを使用するためのプログラミング・インターフェースが記述されています。

## 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 の商標または登録商標です。他の製品名およびサービス名等は、それぞれ IBM® または各社の商標である場合があります。現時点での IBM® の商標リストについては、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) の「著作権と商標情報」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できます。ただし、IBM® の明示的な承諾を得ずに、これらの資料またはその一部について、二次的著作物を配布（頒布、送信を含む）、表示（上映を含む）、または作成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、および表示できます。ただし、IBM® の明示的な承諾を得ることなく、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、または表示したり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場合を除き、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して、明示的にも黙示的にも、いかなる許可、ライセンス、または権利も付与されません。

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完全に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行いません。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。

## IBM オンラインでのプライバシー・ステートメント

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品「ソフトウェア・オファリング」では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が個人情報の収集を可能にする場合、オファリングによる Cookie に関する具合的な情報を以下に記載します。

この「ソフトウェア・オファリング」では、個人情報を収集するために Cookie またはその他のテクノロジーを使用することはありません。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人情報を収集する機能を提供する場合、お客様は、エンド・ユーザーへの通知や同意の要求など情報を収集する際に適用される法律を遵守する必要があります。

このような目的での Cookie などの各種テクノロジーの使用について詳しくは、「IBM オンラインでのプライバシー・ステートメントのハイライト」(<http://www.ibm.com/privacy/jp/ja/>)、「IBM オンラインでのプライバシー・ステートメント」(<http://www.ibm.com/privacy/details/jp/ja/>)の「Cookie、ウェブ・ビーコン、その他のテクノロジー」セクション、および「IBM Software Products and Software-as-a-Service Privacy Statement」(<http://www.ibm.com/software/info/product-privacy>)を参照してください。

# 用語集

この用語集は、本書で使用される用語および省略語を定義しています。探している用語が見つからない場合は、索引 または「*IBM® Dictionary of Computing*」を参照してください。

## A

### 分析 (analysis)

問題の体系的な調査のことで、問題に関連性のある小単位に分けて詳細に調査すること。

### 異常終了 (abend)

タスクの異常終了。タスクの実行中にリカバリー機能により解決できないエラー条件が発生し、タスクが完了せずに終了すること。

### ADFz

IBM® Application Delivery Foundation for z/OS® 製品ファミリー。

### ADFzCC

ADFz 製品に共通機能を提供する IBM Application Delivery Foundation for z/OS Common Components。

## C

### キャッシュ (cache)

頻繁にアクセスされる命令やデータを含むバッファ・ストレージ。アクセス・タイムを短縮するために使用される。

### Compiler Listing

プログラムのコンパイルまたはファイルの作成により出された印刷出力で、行ごとのソース・リスト、相互参照リスト、診断情報、および (プログラムの場合) 外部記述ファイルの説明などをオプションとして含む。

## D

### ダンプ (dump)

エラー情報を収集するために、仮想ストレージの内容の全部または一部をコピーすること。

## E

### エキスパート・システム (expert system)

専門家が取得した知識ベースから推論を行い、特定のアプリケーション・エリアにおける問題解決法を提供するシステム。

## F

### 障害項目 (fault entry)

リアルタイム分析の終了時にメンバーとしてヒストリー・ファイルに保管された、障害に関する情報。障害項目には、ミニダンプも含まれる場合がある。

## H

### ヒストリー・ファイル

個別のメンバーとして障害項目を含む、PDSE データ・セット。

## L

### LANGX サイド・ファイル (LANGX side file)

ADFz Common Components の IPVLANGX ユーティリティー、またはその別名の CAZLANGX、EQALANGX、または IDILANGX の 1 つを使用して 2 進形式に変換される、コンパイラー・リストまたは SYSADATA ファイル。

### リスト (listing)

コンパイラー・リスト (*compiler listing*) を参照。

### 論理ヒストリー・ファイル・サイズ

AUTO 管理になる時点で PDSE ヒストリー・ファイルに割り振られるか、または障害ヒストリー・ファイル設定に明示的に指定される、4K ページの数。

### 疎結合ダンプ・データ・セット

特定の障害項目と一意にリンクされていないダンプ・データ・セットです。障害項目は既存のダンプ・データ・セットの分析の結果として作成されました。

## M

### ミニダンプ

Fault Analyzer によりリアルタイム分析時に参照され、障害用のヒストリー・ファイル項目に保管された、ストレージ・ページ。割り振られた MVS ダンプ・データ・セットが書き込まれていない場合であっても、ミニダンプを使用して、障害を後で再分析できる。

## O

### オプション (option)

Fault Analyzer のオペレーションを制御するパラメーター。

## P

### 区分データ・セット (PDS) (partitioned data set (PDS))

区分に分割された直接アクセス・ストレージ上のデータ・セット。これはメンバーと呼ばれる。それぞれのデータ・セットに、プログラム、プログラムの一部、またはデータを入れることができる。

### 拡張区分データ・セット (PDSE) (partitioned data set extended (PDSE))

区分データ・セットのディレクトリーおよびメンバーに類似した索引付きディレクトリーおよびメンバーを含むシステム管理データ・セット。区分データ・セットの代わりに PDSE を使用できる。

**PDS または PDSE**

区分データ・セット (PDS) または拡張区分データ・セット (PDSE) のいずれかのデータ・セット。

**R****リアルタイム分析**

プログラムの異常終了直後に Fault Analyzer により行われる分析のこと。

**再分析**

2 番目またはそれ以降の分析のこと。

**Recovery fault recording (RFR)**

Fault Analyzer リアルタイム分析の異常終了が発生しても、元のアプリケーション異常終了の障害項目の作成を可能にする Fault Analyzer の機能。

**RFR ダンプ (RFR dump)**

リカバリー障害記録処理中に書き込まれ、RFR 障害項目に関連付けられる SDUMP データ・セットまたは IEATDUMP データ・セット。

**RFR 障害項目 (RFR fault entry)**

リカバリー障害記録処理中に書き込まれ、RFR ダンプ・データ・セットに関連付けられるヒストリー・ファイル障害項目。

**S****保管レポート**

障害項目内に含まれ、再分析を実行する必要なしに、ISPF インターフェースから表示できるレポート。

**サイド・ファイル (side file)**

コンピューター可読フォーマットのリストの圧縮バージョン (人間には読み取れない)。

**スピルされたミニダンプ・ページ**

通常はミニダンプに含まれるが、MaxMinidumpPages を超えたために XDUMP に書き込まれたストレージページ。

**システム・ダンプ (system dump)**

ダンプ (dump) を参照。

**T****TDUMP**

IEATDUMP と同義。Fault Analyzer リカバリー障害記録処理で使用されるダンプ・データ・セット・タイプ。

**密結合ダンプ・データ・セット**

特定の障害項目と一意にリンクされているダンプ・データ・セットです。これは Fault Analyzer によってリアルタイム処理中に作成されました。

## V

### ビュー (view)

Fault Analyzer ISPF

インターフェースを使用して同時に表示される、一連の障害ヒストリー・ファイルのデータ・セット名を含む *PDS* または *PDSE* のメンバー。

## X

### XDUMP

拡張ミニダンプ・データ・セット。

# 索引

## 記号

\_IDL\_OFF  
環境変数  
463  
\_IDLOPTS  
環境変数のオプション  
572  
\_IDLOPTSFILE  
環境変数のオプション  
571  
-DROPCNF-  
データ・セット名 580  
ユーザー出口名 598  
-HistCols キーワード 351  
-Match 351  
:NONE.  
Fault Analyzer  
の検証  
433  
IDIS  
サブシステムの要件  
324, 326  
LE の考慮事項 422  
pre-V10 のパフォーマンスの向上 422  
サポートされるバージョン 27  
バインディング 422  
「Change Fault History File Settings (障害  
ヒストリー・ファイル設定の変更)」画面の  
例 95  
「Compiler Listing Not Found」画面の例 239  
「Confirm Exit」画面の例 173  
「Excluded Program Names」画面の例 166  
「Include」オプション 590  
&SYSCON MVS システム・シンボル  
IDICNFxx  
parmlib メンバー接尾部  
372  
++IDIOPT2  
343  
\$\$BACKUP ヒストリー・ファイル・データ・  
セット・メンバー 27  
\$\$INDEX データ  
キャッシュ 320  
\$\$INDEX ヒストリー・ファイル・データ・  
セット・メンバー 27  
\$\$UFMTX の例 238  
宛先コード  
WTO メッセージ 31  
暗黙最新表示 98  
以下の呼び出しに必要となる Language  
Environment オプション  
Fault Analyzer  
304  
異常終了 S913 340  
異常終了コード  
拡張 212  
異常終了コードの説明  
ユーザー定義 456  
リポジトリー 309  
表示する 128  
異常終了データ、保存 517  
異常終了トラップ 60  
異常終了のトラップ 60  
一時データ・セット  
コンパイラー・リスト制限 391  
サイド・ファイル制限 391  
一致

一致する値をカーソルで選択 84  
既存値の上書き入力 85  
稼働環境  
カスタマイズ 307  
画面  
16 進値列が縮小表示された Associated  
Storage Areas 209  
Abend Job Information 206  
Analyze MVS Dump Data Set 251  
Associated File Control Blocks 181  
Associated Storage Areas 208  
Batch Reanalysis Options 154, 157  
CFA IVP テスト 432  
CFA トランザクション 405, 406  
CICS System Abend Interactive Reanalysis  
Report 254  
CICS System Information 256  
CICS Trace Selection Parameters 186  
CICS Transaction Storage Summary 183  
CICS システム異常終了 Synopsis 255  
CICS トレース 188  
CICS レベル、コマンド域、およびチャネ  
ル 192  
CICS 異常終了ジョブ情報 256  
CICS 情報 182  
CICS 領域の SDSF ジョブ・データ・セッ  
ト 407  
Compiler Listing 213, 214  
Compiler Listing Not Found 239  
Confirm Fault Entry Deletion 131  
Confirm Java Fault Entry Reanalysis 263  
Confirm SYSDUMP Open 164  
Create History File Entry 259  
Create Java Fault Entry 262  
DB2 Information 194  
Dump Storage 215  
Event Details 176  
Event Summary 175  
Fault Analyzer  
Options  
207  
Fault Analyzer  
プリファレンス  
117  
Fault Entry List 63, 64  
Fault Entry List Column Configuration 71  
File Browse 71  
File Information 180  
format CICS auxiliary trace data set 260  
Formatting User Exit Selection List 237  
Hex-Dumped Storage 204  
History File Properties 89  
History File Updates Pending 99  
IMS 情報 197  
IMS 要約情報 201  
Interactive Reanalysis Options 162  
Interactive Reanalysis Report 171  
interactive reanalysis status 170  
Java event details 266  
Java event summary 265  
Java Information 267  
Java の対話式再分析レポート 264  
Last CICS 3270 Screen Buffer 184  
Last CICS 3270 Screen Buffer Hex 185  
LE ヒープ分析情報 204, 205  
Lookup Search and Browse 129  
Message Explanation 212  
Message ID Look-Up 129

MTRACE Records 205  
Preferred formatting Width 128  
Real-Time Report 125  
Specify Compiler Listing or Side File 240  
Specify Move/Copy Options 143  
STCK Conversion 236  
Storage DSECT Mapping Entry 228  
Storage DSECT Mapping Map 228  
Storage RUNCHAIN Command 入力 231  
summarized CICS trace 185  
Synopsis 174  
System-Wide Messages 194  
System-Wide Open Files 179  
System-Wide Storage Areas 203  
User Note List 221  
User Notes Update プロンプト 222  
XMIT オプション 145  
コンパイラー・リスト 212  
サイド・ファイル検索からプログラムを除  
外 243  
ストレージの逆アセンブル 234  
リスト/サイド・ファイルの不一致 241  
リストの表示 70  
レベル 88 項目 211  
レベル 88 項目を縮小表示した associated  
storage areas 210  
終了の確認 173  
除外対象プログラム名 166  
障害ヒストリー・ファイル設定の変更 95  
障害項目情報 134  
新しいヒストリー・ファイルの割り振り 93  
前回アクセスされたヒストリー・ファ  
イル/ビュー 68  
前回アクセスされたヒストリー・ファ  
イル項目 69  
保留状態の更新情報がある Fault Entry  
List 98  
有効になっている CICS 対話式再分析オプ  
ション 258  
抑止ダンプストレージ 217  
画面サイズ 753  
各国語  
設定 426  
拡張ミニダンプ (XDUMP) データ・セット 59  
環境変数  
\_IDL\_OFF  
463  
管理されているヒストリー・ファイル 148  
関連したストレージ域の画面  
16 進値列が縮小表示された例 209  
レベル 88 の項目が縮小されている場合の  
例 210  
例 208  
関連付けられたダンプ・データ・セット 26  
記述子コード  
WTO メッセージ 31  
記録域  
NoDup(CICSFAST(...)) の消去 407  
言語環境  
see LE  
言語環境プログラム CEEWUCHA 異常終了の  
特殊な処理 31  
呼び出し深さ  
maximum 282  
構文図、見方 xii  
行カウント 65  
行コマンド、ISPF 3.4 データ・セット・リス  
トの 329

- 高速 Exclude オプション処理 375
- 国/地域別環境
  - 指定 376
- 再分析
  - interactive 162
  - バッチ 87, 154, 159
  - 対話式
    - see 対話式再分析
- 最小ストレージ要件 297
- 最大呼び出し深さ 282
- 削除確認画面 131
- 使用許諾 dcclix
- 使用出口の概要 301
- 始動 PLT
  - Fault Analyzer
    - のインストール 402
    - プログラムの追加 402
- 主なレポート・セクション 278
- 終了処理ユーザー出口 494
- 終了処理ユーザー出口 (障害項目リフレッシュ) 498
- 重複する障害の判別
  - ユーザー出口での制御 494
  - 条件 619
- 重複ヒストリー 88, 139
- 重複障害カウント 138
- 重複障害検出
  - 概要 611
- 重複障害処理
  - 概要 56
- 出口
  - CICS 303
  - Fault Analyzer
    - の呼び出しの場合 299
  - MVS 299, 303
  - ユーザー
    - see ユーザー出口
  - 言語環境プログラム 300, 303
- 除外されたヒストリー・ファイル 149
- 紹介 20
- 詳細 PCB 画面 202
- 障害
  - 再分析 62
- 障害 ID
  - フォーマット 606
- 障害ヒストリー・ファイル 25
  - リアルタイム実行時の選択 34
- 障害ヒストリー項目
  - カーソル一致 86
  - コピーする 143
  - ワイルドカード 86
  - 移動 145
  - 移動範囲 145
- 障害項目
  - ISPF 行コマンド 87
  - システム間での送信 361
  - ソート 83
  - パッケージ化 146
  - ロック
    - flag 117
    - メソッド 132
    - 強調表示 117, 132
    - 削除の防止 100
    - 出口 132
  - 一致 83
  - 検索 83
  - 最小/最大 94
  - 削除する 131
  - 障害項目接頭部 94
  - 選択 83
  - 送信 145
  - 分析向けの収集 517
  - 有効期限の制御 100
  - 障害項目のリフレッシュ処理 244
  - 障害項目の最小数 94
  - 障害項目の最大数 94
  - 障害項目の重複ヒストリー 139
  - 障害項目リフレッシュ
    - 終了処理ユーザー出口 498
  - 障害項目情報
    - 最新表示 98
    - 表示 134
  - 障害項目選択
    - 初期基準の指定 351
  - 障害再分析 23
    - interactive 162
    - バッチ 154
    - 独自のジョブの作成 160
  - 障害点 668
  - 障害分析レポート
    - 表示 89
  - 条件
    - コンパイラー・リスト選択 31
    - サイド・ファイル選択 31
    - 正常な分析 668
  - 状況ポップアップ画面 170
  - 新しいヒストリー・ファイルの割り振り画面の例 93
  - 推奨されるストレージ 297
  - 据え置きブ레이크ポイント機能 215, 250
  - 正常な分析
    - 条件 668
  - 選択されたヒストリー・ファイル 150
  - 前回アクセスされたヒストリー・ファイル/ビューの画面の例 68
  - 前回アクセスされたヒストリー・ファイル項目の画面の例 69
  - 多文化サポート 426, 607, 609
  - 対話式オプション行 163
  - 対話式レポート
    - Abend Job Information 206
    - COBOL Explorer 246
    - Event Summary 175
    - Fault Analyzer
      - Options 207
    - Java 分析の表示 264
    - Synopsis 174
    - ソース・コードの表示 212
    - ユーザー・ノート 207
    - 一般的な解説 171
    - 拡張メッセージ 212
    - 拡張異常終了コード 212
    - 関連したストレージ域の表示 208
    - 終了 173
    - 遅延ブ레이크ポイント機能 250
    - 保管場所の表示 215
  - 対話式画面
    - ファイルへのコピー 130
  - 対話式再分析 25
    - CICS 下での実行 286
    - オプション 162
    - データ・セット 244
    - ユーザー・ノート 152, 218
    - 開始 88, 170
    - 実行 162
  - 通知ユーザー出口 499
  - 通知ユーザー出口 (MVS SVC ダンプ登録) 507
  - 特記事項
    - Fault Analyzer
  - dcclix
    - 特殊なヒストリー・ファイル・データ・セット・メンバー 26
    - \$\$BACKUP 27
    - \$\$INDEX 27
    - 日本語フィーチャー
      - ISPF データ・セットの割り振り 425
      - カスタマイズ 425
    - 入力パラメーター・リスト
      - IDISNAP 40
    - 反復可能項目、構文図 xii
    - 汎用レポート情報 275
    - 非 CICS トランザクション異常終了
      - 分析の呼び出し 299
    - 非 ISPF インターフェース
      - インストール 563, 563
      - ヒストリー・ファイルへのアクセス 286
    - 非 LE ランタイム
      - アプリケーション 335
    - 非 REXX ユーザー出口のバッファ・データ・フォーマット 636
    - 非 REXX ロギング・ルーチン
      - REXX からの呼び出し 503
    - 必要なプログラム
      - CICS に対する定義 401
    - 表示の合計行 65
    - 表示域
      - 拡大 332
    - 表示情報
      - 優先フォーマット幅の設定 128
    - 幅
      - 優先フォーマットの設定 128
    - 分析
      - リアルタイム
        - see リアルタイム分析
      - 分析エンジン 20
      - 分析プロセス 20
      - 分析レポート、リアルタイム 37
      - 分析制御ユーザー出口 475
      - 分析制御ユーザー出口 (MVS SVC ダンプ登録) 479
      - 変数、構文図 xii
      - 保管レポート
        - 表示 125
      - 保管場所
        - 表示する 215
      - 保留状態の更新情報 98
      - 保留状態の更新情報がある Fault Entry List 画面の例 98
      - 本書の編成 xii
      - 戻りコード
        - IDIDA 693
        - IDILANGX
        - ・ユーティリティ 693
        - IDIUTIL
        - バッチ・ユーティリティ 693
        - バッチ再分析 693
      - 戻り指定
        - IDISNAP 43
      - 抑止されるコピーブック 277
      - 抑止ダンプ出口
        - IDIXDCAP
        - インストール 334
      - 領域サイズ
        - 追加に必要な 30
        - 領域サイズの要件 297



例、XFACILIT インプリメンテーション  
の 367, 367, 368  
例、レポート 284  
列  
-HistCols キーワード 351  
16 進値  
see 16 進値列  
Fault Entry List 画面 73  
デフォルトのレイアウトの指定 351  
構成のソース 72  
列構成ソースの識別 72  
論理ヒストリー・ファイル・サイズ 347

## 数字

16 進値列  
非表示にする 209  
3270 screen buffer  
ブレーン・テキスト表示 184  
3270 画面バッファ  
フルカラー・ディスプレイ 105  
64 ビット・アドレス  
指定 147

## A

Abend Job Information 画面の例 206  
ADATA オプション  
SYSADATA ファイルの作成 377  
AdditionalIDIOFFDD  
オプション  
573  
ADDR タグ 544  
address TSO REXX コマンド 468  
ADFzCC  
IPVCNF00 parmlib メンバー 567  
IPVOPTLM 構成オプション・モジュール 567  
ADFzCC  
サーバー  
プラグイン用のカスタマイズ 563  
Analyze MVS Dump Data Set 画面の例 251  
AREA タグ 544  
Associated File Control Blocks 画面の例 181  
AUTO で管理される PDSE ヒストリー・ファイル 94, 347  
Available ダンプ状況 139

## B

Batch Reanalysis Options 画面の例 154, 157  
BatchOpts オプション 120, 599  
BPX014I メッセージ 308

## C

C プログラム  
CSECT 命名 383  
Fault Analyzer  
の検証  
430  
Java サンプル 273  
CA-Panexec  
出口 336  
CallEqauDat オプション 573  
CE コマンド 104  
CEEWUCHA  
特殊な処理 31  
CEEWUCHA LE ユーザー条件ハンドラー 275  
CFA 405  
現在分析中のタスクでの FORCEPURGE の  
実行 403  
CFA CICS トランザクション 401  
CFA IVP テスト画面の例 432  
CFA トランザクション 403  
CFA トランザクション画面の例 405

CFA 終了オプション画面の例 406  
CICS  
CICS プログラム定義ジョブの例 401  
IDIOPTS  
DDname  
417  
IVP テスト 432  
XPCABND グローバル・ユーザー出口 405,  
414  
カスタマイズの検証 432  
サポートされるバージョン 27  
タスク・ストレージ域 105  
トランザクション異常終了分析 405, 414  
始動 PLT  
Fault Analyzer  
のインストール  
402  
プログラムの追加 402  
重複障害基準 617  
対話式再分析の実行 286  
必要なプログラム 401  
保護キーのサポート 27  
CICS 3270 screen buffer  
ブレーン・テキスト表示 184  
CICS 3270 画面バッファ  
フルカラー・ディスプレイ 105  
CICS Information 画面の例 182  
CICS NoDup(CICSFAST) オーバーライド・ア  
センブラー出口 ( IDINFUE )  
414  
CICS Storage Accounting Area (SAA) オーバー  
レイ 183, 419  
CICS System Abend Interactive Reanalysis  
Report  
abend job information 画面 256  
CICS System Information 画面 256  
Options in Effect 画面 258  
Synopsis 画面 255  
オプションの設定 162  
ファースト・パス・コマンド 255  
画面の例 254  
CICS Trace Selection Parameters 画面の  
例 186  
CICS Transaction Storage Summary 画面の  
例 183  
CICS XDUREQ グローバル・ユーザー出口  
Fault Analyzer  
分析の停止  
341, 342  
CICS オープン (L9) TCB  
異常終了の取り込み 418  
CICS グローバル・ユーザー出口 34  
CICS コマンド域 192  
CICS コンテナー 192  
CICS サンプル・プログラム定義ジョブ 404  
CICS システム・ダンプ分析のためのファース  
ト・パス・コマンド 255  
CICS システム異常終了ダンプ分析 251  
CICS システム異常終了分析  
オプションの設定 251  
ダンプ・データ・セットの選択 251  
ヒストリー・ファイル項目の作成 259  
ユーザー出口使用法 251  
CICS システム初期化パラメーター  
SIT コマンドで表示 117  
対話式再分析中の表示 194  
CICS シャットダウン PLT  
see シャットダウン PLT  
CICS タスク・トレース・テーブル 192

CICS ダンプ・データ・セット  
CICS トランザクション LE ダンプ出力 417  
CICS チャンネル 192  
CICS トランザクション LE ダンプ出力  
CICS ダンプ・データ・セットへの誘導 417  
CICS トランザクション異常終了  
特殊処理 341, 342  
CICS トランザクション異常終了の場合の呼び  
出し 303  
CICS トランザクション異常終了分析  
CFA 403  
CICS に対する定義 401  
パフォーマンスの最大化 419  
動的制御の有効化 403  
CICS トレース  
LE からの循環を回避 417  
システム全体情報 186  
ストレージ域のサイズ 416  
画面の例 188  
CICS トレース、要約済み  
System-Wide Information 185  
CICS レベル 192  
CICS 環境  
カスタマイズ 400  
CICS 環境での対話式再分析 564  
CICS 高速重複障害抑止 496, 618  
CICS 補助トレース・データ・セット 260  
CICS 用の言語環境プログラム異常終了出口  
使用可能にする 401  
CICS 領域の SDSF ジョブ・データ・セット画  
面の例 407  
CICSD コマンド 105  
CICSDumpTableExclude オプション 574  
CICSSTG コマンド 105  
CICSTranAnalysisUser オプション 575  
CMD5 コマンド 105  
COBOL  
Java サンプル 272  
サポートされるバージョン 27  
基本ロケーション 211  
COBOL Explorer 246  
例 246  
COBOL SYSDEBUG ファイルの使用 381, 384  
COBOL で抑止されるコピーブック 277  
COBOL プログラム  
Fault Analyzer  
の検証  
428  
COBOL 呼び出し  
IDISNAP  
43  
COBOL 報告書作成プログラム・プリコンパイ  
ラー 378  
COLS コマンド 106  
com.ibm.faultanalyzer.Snap.dump  
method  
リアルタイム分析レポート 37  
Compiler Listing 画面の例 212, 213, 214  
Confirm Fault Entry Deletion オプション 131  
Confirm Fault Entry Deletion 設定 117  
Confirm SYSMDUMP Open 画面の例 164  
COPY コマンド 106  
copyright  
表示する 131  
Create History File Entry 画面の例 259  
CSECTS  
命名基準 383  
CSV042I メッセージ 308  
CTL データ域 636  
CUROPTS コマンド 106

## D

DATA タグ 545  
DataSets オプション 576  
DB2  
    System-Wide Information 画面の例 194  
DB2 IVP  
    C 434  
    COBOL 435  
DD タグ 545  
DDnames  
    IDIRLOAD  
        392  
DEBUG オプションの考慮事項 381  
DeferredReport オプション 584  
DELETE 制御ステートメント 445  
Detail オプション 586  
DFHRPL 連結 404  
DISASM コマンド 107  
DL タグ 545  
DROPCNF  
    データ・セット名 580  
    ユーザー出口名 598  
DSECT コマンド 107  
DSECT データ・セット (\$DINDEX メンバー)  
    索引付け 230  
DSECT 索引作成のユーティリティ (IDIPDSCU)  
    ) 230  
DSECT 情報  
    ストレージ域のマッピング 227  
DSNACLI デフォルト DB2 ブラン 422  
DT タグ 546  
DTJF Processing (MQ 処理) 269  
DUMMY データ・セット  
    DataSets オプションを介した指定 579  
Dump Storage  
    IDIXDLOC  
        関数を使用した検索  
            553  
    IDIXXLOC  
        関数を使用した検索  
            561  
Dump Storage 画面の例 215  
DUMP タグ 547  
DUMP 異常終了トランザクション・オプション 414  
DUMPA タグ 548  
DumpDSN オプション 587  
DumpRegistrationExits オプション 588  
DUPS コマンド 107

## E

Eclipse プラグイン  
    ADFzCC  
        サーバー構成  
            563  
        概要 286  
EDIT コマンド 107  
Enterprise COBOL  
    Java サンプル 272  
Enterprise PL/I  
    64 ビット Java サンプル 272  
    Java サンプル 271  
    SYSDEBUG ファイルの使用 381, 384  
ENV データ域 650  
EPC データ域 667  
EQUATEDAT 出口  
    SYSDEBUG ファイルの位置決め 385  
ErrorHandler オプション 589  
EVALUATE REXX コマンド 519

Event Summary  
    画面の例 175  
    詳細 176  
Exclude オプション 590  
Exclude 処理  
    使用したジョブの分析 374  
EXEC CICS ABEND コマンドの ABCODE キーワード 405  
EXEC CICS ABEND コマンドの NODUMP キーワード 405  
EXEC コマンド 108  
Exits オプション 595  
EXITs 制御ステートメント 450  
EXPORT 制御ステートメント 449

## F

FA の標準日付形式 645  
Fault Analyzer  
    422  
        CPU 使用時間 514, 518  
        IFAPRDxx parmlib メンバーへの登録 317  
        PL/I V2R3 アプリケーションからの暗黙呼び出しの有効化 335  
        オフにする (IFAPRDxx parmlib メンバー) 462  
        オフにする (JCL スイッチ IDIOFF)  
        ) 463  
        カスタマイズ  
            (CICS 用) 400  
            for ISPF 328  
            IDIOPTLM  
            構成オプション・モジュールを使用  
                338  
            USERMOD を使用 334  
            ユーザー出口を使用 464  
            準備 292  
        カスタマイズの検証 427  
        マイグレーション (migrating)  
            V10.1 から V11.1 への 290  
            V11.1 から V12.1 への 290  
            V12.1 から V13.1 への 289  
            V13.1 から V14.1 への 289  
            V14.1 から V15.1 への 288  
            V9.1 から V10.1 への 291  
            旧バージョンからの 288  
        メンテナンス 459  
        ライセンス情報の表示 131  
        レポート 275  
        一時的にアンインストール 461  
        環境変数をオフにする 463  
        起動  
            CICS から 401  
            Java try-catch ブロックから 47  
            Java ダンプ・イベント 50  
            Language Environment for CICS を使用 401  
            PL/I PLIDUMP 335  
            SDSF 331  
        許可 307  
        呼び出しの有効化 334  
        使用不可 461  
        著作権情報の表示 131  
        特記事項 dcclix  
        分析の停止 341, 342  
        類似のサード・パーティー製品を使用した  
        実行 305  
Fault Analyzer  
    DB2 パフォーマンス  
        向上 422

Fault Analyzer ISPF インターフェース 62  
Fault Analyzer  
    Options  
        207  
Fault Analyzer  
    plug-in for Eclipse  
        インストール 286, 563  
Fault Analyzer  
    システム要件  
        27  
Fault Analyzer  
    のサブシステム  
        ユーザー出口の実行元 588  
Fault Analyzer  
    の終了  
        402  
Fault Analyzer の登録 317  
Fault Analyzer  
    プリファレンス  
        117  
Fault Analyzer  
    モジュール  
        使用可能にする 307  
Fault Entry Information 画面の例 134  
Fault Entry List Column Configuration 画面の例 71  
Fault Entry List 画面  
    カスタマイズ 376  
Fault Entry List 画面の例 63  
FaultID オプション 605  
File Browse 画面の例 71  
File Information 画面の例 180  
File Manager for  
z/OS  
    必要な ISPF データ・セット割り振り 328  
FILES 制御ステートメント 442  
FIND コマンド 108  
    画面のタイプによる差異 110  
FND 域  
    クリアする 407  
Format CICS Auxiliary Trace Data Set 画面の例 260  
Formatting User Exit Selection List の例 237

## G

GenerateSavedReport オプション 606  
GlobalExclude オプション 603  
GRS 365

## H

Hex-Dumped Storage 画面の例 204  
HistCols オプション 604  
History File Properties 画面の例 89  
History File Updates Pending 画面の例 99  
HLASM 46  
HP タグ 549

## I

ICH408I メッセージ 340  
ICH420I メッセージ 308  
ICH422I メッセージ 308  
IDLSDUMP\_ACCESS  
    XFACILIT リソース・クラス 47, 311  
IDI.SIDISAM1  
    サンプル・メンバー  
        422  
IDI\* メッセージ 695  
IDIADATA  
    DataSets オプションを使用した指定 576  
    DD statement (SQL ステートメント) 36  
    データ・セット 640, 640, 640  
    属性 391

IDIADATA  
 DD ステートメント  
 377  
 IDIALLOC  
 REXX コマンド  
 523  
 IDIBOPT  
 DDname  
 571  
 IDICHI  
 ユーティリティー  
 292  
 IDICHI  
 ・ユーティリティー  
 460  
 IDICNFUM  
 ユーザー・オプション・モジュール  
 569, 570  
 IDICNFxx  
   &SYSCLONE parmlib メンバー接尾部 372  
   parmlib メンバー 372, 569  
   オプション・ファイル 569  
   除去  
     データ・セット 580  
     ユーザー出口 598  
   代替 parmlib データ・セット 340  
 IDIDATST  
 393  
 IDIDTEST  
 REXX コマンド  
 527  
 IDIDOC  
   DataSets オプションを使用した指定 576  
   READ アクセス要件 310  
 IDIDOxxx  
   DataSets オプションを使用した指定 576  
 IDIDSECT  
   DataSets オプションを使用した指定 576  
   連結 230  
 IDIDSECTdsn  
 REXX コマンド  
 528  
 IDIDNSTEST  
 REXX コマンド  
 529  
 IDIEventInfo  
 REXX コマンド  
 530  
 IDIEXEC  
   DataSets オプションを使用した指定 576  
 IDIEXEC  
 DDname  
 467, 588, 596, 625  
 IDIFREE  
 REXX コマンド  
 531  
 IDIGET  
   REXX コマンド 532  
 IDIGSVRJ  
 サンプル・メンバー  
 563  
 IDIHIST  
   DataSets オプションを使用した指定 576  
   データ・セット 659  
 IDIHIST  
 DD ステートメント  
 349  
 IDIHIST\_GROUP\_DSN  
   XFACILIT リソース・クラス 366  
 IDIHIST\_USERID\_DSN  
   XFACILIT リソース・クラス 366  
  
 IDIHUSRM  
 455  
 IDIJAVA  
   DataSets オプションを使用した指定 576  
 IDIJAVA  
 オプション  
 395  
   例 396  
 IDIJVM  
 DD ステートメント  
 326  
 IDIJVM6  
 DD ステートメント  
   JVM  
     デフォルトの指定 326  
 IDILANGX  
   DataSets オプションを使用した指定 576  
   DD statement (SQL ステートメント) 36  
   データ・セット 643, 643, 643  
   属性 391  
   戻りコード 693  
 IDILANGX  
 に必要なコンパイラ・オプション  
 378  
 IDILC  
   DataSets オプションを使用した指定 576  
   DD statement (SQL ステートメント) 36  
   データ・セット 640, 641, 641  
   属性 391  
 IDILCOB  
   DataSets オプションを使用した指定 576  
   DD statement (SQL ステートメント) 36  
   データ・セット 641, 642, 642  
   属性 391  
 IDILCOBO  
   DataSets オプションを使用した指定 576  
   DD statement (SQL ステートメント) 36  
   データ・セット 642, 642, 643  
   属性 391  
 IDILPLI  
   DataSets オプションを使用した指定 576  
   DD statement (SQL ステートメント) 36  
   データ・セット 644, 644, 644  
   属性 391  
 IDILPLIE  
   DataSets オプションを使用した指定 576  
   DD statement (SQL ステートメント) 36  
   データ・セット 644, 645, 645  
   属性 391  
 IDIMAPS  
   DataSets オプションを使用した指定 576  
 IDIModQry  
 REXX コマンド  
 532  
 IDINDFUE  
 CICS NoDup(CICSFAST) オーバーライド・ア  
 センブラー出口  
 414  
 IDIOFF  
 DD ステートメント  
 463  
 IDIOPTLM  
   サンプル・ジョブ 338  
   構成オプション・モジュール 338  
 IDIOPTLM 構成オプション・モジュール 340  
 IDIOPTS  
   DD ステートメント 161  
   オプション・ファイル 417  
 IDIOPTS  
 DDname  
 568, 569  
  
 IDIOPTS  
 オプション・ファイル  
 571  
 IDIPANEX  
 サンプル・メンバー  
 336  
 IDIPUT  
   REXX コマンド 533  
 IDIRegisterFaultEntry  
 REXX コマンド  
 534  
 IDIREPRT  
 DD ステートメント  
 37  
   DUMMY 割り振り 38  
   SYSOUT クラスの変更 37  
   ユーザー出口の割り振り 37  
   レポートの結合 37  
   レポートの抑止 38  
   動的割り振り 37  
   動的割り振り解除の防止 629  
 IDIRFR\_TDUMP\_HLQ  
   XFACILIT リソース・クラス 312  
 IDIRLOAD  
 DDname  
   CSECT マッピング 392  
 IDIROBOT  
 サンプル EXEC  
 353  
 IDIS  
 サブシステム  
 319  
   DB2 の要件 324  
   Java 用の要件 325  
   ストレージ要件 324  
   削除の要件 326  
   始動 321  
   状況の表示 147  
   停止 326  
 IDISCICS  
 サンプル・メンバー  
 404  
 IDISCLST  
 EXEC  
 383  
 IDISCMDS  
 コマンド・テーブル  
 328  
 IDISCNFU  
 サンプル・メンバー  
 570  
 IDISCPRO  
 EXEC  
 383  
 IDISDB2B  
 サンプル・メンバー  
 435  
 IDISF\* メッセージ 695  
 IDISFA  
 サンプル・メンバー  
 329  
 IDISFE\*  
 サンプル・メンバー  
 741  
 IDISHIST  
 サンプル・メンバー  
 345  
 IDISINFO  
 コマンド  
 111, 147  
 IDISISPF

サンプル・メンバー  
 328  
 IDISJAV1  
 サンプル・メンバー  
 271  
 IDISJAV2  
 サンプル・メンバー  
 272  
 IDISJAV3  
 サンプル・メンバー  
 272  
 IDISJAV4  
 サンプル・メンバー  
 272  
 IDISJAV5  
 サンプル・メンバー  
 273  
 IDISJAV6  
 サンプル・メンバー  
 273  
 IDISJCTL  
 スケルトン・メンバー  
 332  
 IDISNAP  
 39, 657  
     COBOL 呼び出しの例 43  
     IDISNAP  
     サンプル・メンバー  
     40  
     PL/I 31 ビット DLL 呼び出しの例 45  
     PL/I 64 ビット DLL 呼び出しの例 45  
     PL/I 非 DLL 呼び出しの例 44  
     Snapdata オプション 40, 628  
     アセンブラー呼び出しの例 46  
     エントリーの指定 39  
     呼び出し 39  
     入力パラメーター・リスト 40  
     戻り指定 43  
 IDISPDМ サンプル・メンバー 335  
 IDISPLI  
 USERMOD  
 335  
 IDISPLI  
 サンプル・メンバー  
 335  
 IDISPLIA  
 USERMOD  
 335  
 IDISPLIA  
 サンプル・メンバー  
 335  
 IDISRC1  
 サンプル・メンバー  
 430  
 IDISROBT  
 サンプル・メンバー  
 356  
 IDISTSOB  
 サンプル・メンバー  
 360  
 IDISUFM3  
 サンプル・メンバー  
 542  
 IDISUFMn  
 サンプル・メンバー  
 494  
 IDISUFMX  
 サンプル・メンバー  
 238  
 IDISUSI  
 サンプル・メンバー

299  
 IDISUSI  
 出口サンプル  
 299  
 IDISUTL1  
 サンプル・メンバー  
 516  
 IDISVENU  
 サンプル・メンバー  
 309  
 IDISVJPN  
 サンプル・メンバー  
 310  
 IDISXNFB  
 サンプル・メンバー  
 354  
 IDISXPLA  
 サンプル・メンバー  
 468  
 IDISXPLB  
 サンプル・メンバー  
 468  
 IDISXPLC  
 サンプル・メンバー  
 468  
 IDISXPLP  
 サンプル・メンバー  
 468  
 IDISYSDB  
     DataSets オプションを使用した指定 576  
     DD statement (SQL ステートメント) 36  
     データ・セット 646, 646, 646  
     属性 391  
 IDISYSLB 連結 227  
 IDITRACE  
     コンパイラー・リスト/サイド・ファイル  
     の検索トレースの有効化 387  
     書き込み 559  
     動的に制御 646  
 IDITRACE  
 CICS において  
 407  
 IDITRACE  
 DD ステートメント  
 469, 570, 594, 620, 713  
 IDIUTIL  
 ListHF ユーザー出口  
 512  
 IDIUTIL  
 ListHFDUP ユーザー出口  
 517  
 IDIUTIL  
 インポート・ユーザー出口  
 508  
 IDIUTIL  
 バッチ・ユーティリティ  
 441  
     戻りコード 693  
 IDIUTIL  
 バッチ・ユーティリティ・ユーザー出口の  
 例  
 454  
 IDIUTIL  
 削除ユーザー出口  
 511  
 IDIVIEWS  
 DDname  
 66  
 IDIVPASM  
 サンプル・メンバー  
 427, 427

IDIVPBLE  
 サンプル・メンバー  
 431  
 IDIVPC  
 サンプル・メンバー  
 430  
 IDIVPCOB  
 サンプル・メンバー  
 428, 431  
 IDIVPDB2  
 サンプル・メンバー  
 434  
 IDIVPDBB  
 サンプル・メンバー  
 435  
 IDIVPPLE  
 サンプル・メンバー  
 429  
 IDIVPLI  
 サンプル・メンバー  
 429, 431  
 IDIVSENU  
     READ アクセス要件 310  
 IDIVSJPN  
     READ アクセス要件 310  
 IDIVSxxx  
     DataSets オプションを使用した指定 576  
 IDIWCIDI サンプル・メンバー 565  
 IDIWRITE  
 REXX コマンド  
 535  
 IDIWTO  
 REXX コマンド  
 537  
 IDIWSEL  
 サンプル・メンバー  
 418  
 IDIXCCEE  
 657  
 IDIXCEE  
 657  
 IDIXCEE  
 LE 出口の有効化  
     検証 431  
 IDIXCX53  
 出口  
 405, 656  
 IDIXDCAP  
 657  
     インストール 334  
 IDIXDCAP  
 リアルタイム分析  
     抑止 343  
 IDIXDLOC  
 関数  
 553  
 IDIXEINF  
 関数  
 554  
 IDIXFA  
 405  
 IDIXFXIT  
     エントリーの指定 369  
     ユーザー出口目的 369  
     入力パラメーター・リスト 369  
     戻り指定 370  
     例 371  
 IDIXGETN  
 関数  
 555  
 IDIXJAVA

47  
 IDIXLIST  
 関数  
 557  
 IDIXMIT  
 サンプル・メンバー  
 361  
 IDIXSFOR  
 アセンブラーの例 391  
 入力パラメーター・リスト 388  
 IDIXSFOR  
 出口  
 Compiler Listing 388  
 サイド・ファイル (side file) 388  
 IDIXTSEL  
 657  
 IDIXUFMT  
 ロード・モジュール  
 552  
 IDIXDLOC  
 関数  
 553  
 IDIXEINF  
 関数  
 554  
 IDIXGETN  
 関数  
 555  
 IDIXGETS  
 関数  
 556  
 IDIXGETX  
 関数  
 556  
 IDIXLIST  
 関数  
 557  
 IDIXNOTE  
 関数  
 558  
 IDIXTRCE  
 関数  
 559  
 IDIXWRIT  
 関数  
 559  
 IDIXWTO  
 関数  
 560  
 IDIXXLOC  
 関数  
 561  
 関数 553  
 IDIXWRIT  
 関数  
 559  
 IDIXWTO  
 関数  
 560  
 IDIXXLOC  
 関数  
 561  
 IEATDUMP  
 ダンプ・タイトル 59  
 リカバリー障害記録のデフォルト名の変更 341  
 IEAVTABX MVS 出口 36  
 IFAPRDxx parmlib メンバー 462  
 Fault Analyzer の登録 317  
 IGZUIXB 出口  
 SYSDEBUG ファイルの位置決め 384

I  
 IDI  
 S\$NDX サンプル・メンバー  
 27  
 IMPORT 制御ステートメント 448  
 IMS  
 Java サンプル 272  
 LE の考慮事項 424  
 サポートされるバージョン 27  
 IMS Environment  
 カスタマイズ 424  
 IMS Information 画面の例 197  
 IMS 高速重複障害抑制 613  
 IMS 情報  
 System-Wide Information 197  
 IMS 情報要約画面の例 201  
 INFO コマンド 111  
 Interactive Reanalysis Report 画面の例 171  
 interactive reanalysis status 画面の例 170  
 InteractiveExitPromptsSeconds オプション 605  
 IPL 要件 307  
 IPVCNF00 parmlib メンバー 340, 343, 567  
 IPLLANGX  
 メッセージ 695  
 IPVOPTLM 構成オプション・モジュール 343, 567  
 ISPF  
 Fault Analyzer  
 の検証  
 438  
 ISPF インターフェース 62  
 オンライン・ヘルプ 63  
 カスタマイズ 376  
 コマンド 104  
 セキュリティの考慮事項 146  
 メニュー・バーのプルダウン・メニュー 100  
 起動 63  
 行コマンド 87  
 障害項目コマンド 87  
 新規ユーザーにデフォルトを提供する 331  
 分割画面のサポート 63  
 ISPF データ・セット  
 日本語フィーチャー用の割り振り 425  
 ISPF データ・セットの割り振り 328  
 ISPF バック・データ・フォーマット 394  
 ISPF 環境  
 変更 328  
 ISPF 選択パネル  
 更新 329  
 ISPLIBD TSO/ISPF コマンド 328  
 ISRDDN TSO/ISPF コマンド 328  
 ISRFIND TSO/ISPF コマンド 328  
 IVP テスト  
 CICS 432

## J

Java  
 Fault Analyzer  
 の呼び出し  
 47  
 IDIS  
 サブシステムの要件  
 325  
 JVTMI エージェント 52  
 サポートされるバージョン 27  
 レポートの制限 271  
 最適なデバッグのためのコンパイル 397, 398  
 Java API  
 ダウンロード・レポート 744

Java event details 画面の例 266  
 Java Event Summary 画面の例 265  
 Java Information  
 System-Wide Information 204  
 Java Information 画面の例 267  
 Java try-catch ブロック  
 サンプル 273  
 Java アプリケーション異常終了  
 Java ソース情報の提供 395  
 必須の LE オプション 304  
 例 396  
 Java イベント 269  
 Java サンプル  
 64 ビット・エンタープライズ PL/I 272  
 C++ プログラム 273  
 Enterprise COBOL 272  
 Enterprise PL/I 271  
 IMS バッチ処理 272  
 データ・セット 271  
 ラッパー・ユーティリティ 273  
 一時ファイル 271  
 要件 271  
 Java ダンプ・イベント  
 Fault Analyzer  
 の呼び出し  
 50  
 Java ダンプ・データ・セット  
 選択 262  
 Java ダンプ分析  
 ヒストリー・ファイル障害項目の作成 262  
 Java の対話式再分析レポート画面の例 264  
 Java ラッパー・ユーティリティ 50  
 サンプル 273  
 Java 取り込み SDUMP データ・セット  
 XFACILIT リソース・クラスによる管理 47  
 Java 障害項目の再分析 263  
 Java 分析 262  
 オプションの設定 262  
 対話式レポートでの表示 264  
 JCL コマンド 111  
 JclCapture オプション 607  
 JVTMI エージェント 52

## L

L タグ 549  
 Language Environment for CICS  
 Fault Analyzer  
 を起動するための構成  
 401  
 Language オプション 607  
 LangxCapture オプション 608  
 Last CICS 3270 Screen Buffer Hex 画面の例 185  
 LE  
 CICS トレースの循環を回避 417  
 DB2 についての考慮事項 422  
 IMS の考慮事項 424  
 LE heap analysis  
 System-Wide Information 204  
 LE オプション 304, 417  
 CICS 異常終了の場合 304  
 Java アプリケーションの異常終了の取り込み 304  
 非 CICS 異常終了の場合 304  
 LE ヒープ分析  
 情報画面の例 204, 205  
 LE 異常終了出口  
 MVS 変更オプション/抑止ダンプ出口 300  
 使用可能にする 334  
 LI タグ 549  
 Liberty 分析または Java ダンプ分析 262

LINKLIST  
モジュール 307

LIST REXX コマンド 538

LISTHF 制御ステートメント 442

LISTHFDUP 制御ステートメント 443

LOADER の制限 33

Locale オプション 609

LOOKC ISPF コマンド 331

LOOKC コマンド 113

Lookup Search and Browse 画面の例 129

LOOKUP コマンド 112

カーソル選択を使用した起動 331

LoopProtection オプション 609

LPA  
モジュール 307  
モジュールを入れる 459

LST データ域 668

## M

MATCH ALL 一致条件 85

MATCH CSR 一致条件 86

MATCH コマンド 83, 85, 113

MaxMinidumpPages オプション 610

Message ID Look-Up 画面の例 129

MQSeries  
サポート要件 27

MTRACE Records  
System-Wide Information 205

MTRACE Records 画面の例 205

MVS IEAVTABX 出口 36

MVS SVC ダンプ登録  
通知ユーザー出口 507  
分析制御ユーザー出口 479

MVS オープン・プロンプト 164

MVS コンソール  
IDIXWTO  
関数を使用して書き込み  
560

MVS ダンプ・データ・セットのサイズ 305

MVS ポスト・ダンプ出口

IDIXTSEL  
インストール 418

MVS 変更オプション  
インストール 334

MVS 変更オプション/抑止ダンプ出口  
LE 異常終了出口 300

## N

NEXT コマンド 113

NFY データ域 672

NoDup オプション 611

NoDup(CICSFAST) オプション  
変更 618

NoDup(ImageFast) オプション  
変更 613

NoErrorHandler オプション 589

NOIPVOPT の設定 343

NoPrintInactiveCOBOL オプション 624

NoQuiet オプション 624

NoSource オプション 629

NoSpinDIREPRT  
オプション  
629

Not found ダンプ状況 139

NOTE REXX コマンド 540

NOTE コマンド 113

NOTEL タグ 550

NOTELIST コマンド 114

NOTEST(DWARF) オプションの制限 382

## O

OMVS ホーム・パス 320

## P

p タグ 550

PARM フィールド・オプション 572

parmlib データ・セット  
IDICNFxx  
に代替を指定  
340

PDS または PDSE  
ヒストリー・ファイルとしての割り振り 345  
ヒストリー・ファイルの変換 347

PDTCCopts オプション 620

PermitLangx オプション 621

PF キー  
表示 64

PL/I  
64 ビット Java サンプル 272  
Fault Analyzer  
の検証  
429  
Java サンプル 271  
サポートされるバージョン 27

PL/I ON ERROR 使用 305

PL/I TEST (NOSEPERATE NOSOURCE) オプションの制限 382

PL/I V2R3 アプリケーション  
Fault Analyzer  
暗黙呼び出しの有効化  
335

PL/I バージョン 2 リリース 3  
Fault Analyzer  
の起動を容易にするための USERMOD  
335

PL/I 呼び出し  
31 ビット DLL の例 45  
64 ビット DLL の例 45  
非 DLL の例 44

PLIDUMP  
Fault Analyzer  
の呼び出し  
335

PLT  
CICS 始動  
Fault Analyzer  
のインストール  
402  
必要なプログラム 402

Plug-in for Eclipse  
インストール 563  
概要 286

Preferred Formatting Width 画面の例 128

PreferredFormattingWidth オプション 623

PREV コマンド 114

PrintInactiveCOBOL オプション 624  
programs  
CICS に対する定義 404  
障害分析用に設定 30

## Q

Quiet オプション 624

QUIT コマンド 114

## R

READ アクセス要件  
IDIDOC  
310  
IDIVSENU  
310  
IDIVSJPN  
310

REFRESH コマンド 98, 115

RefreshExits オプション 625

RESET コマンド 115

RetainCICSDump オプション 626

RetainDump オプション 627

REXX  
非 REXX ロギング・ルーチンの呼び出し 503

REXX exec ライブラリー  
ポインティング 376

REXX SAY 命令 474

REXX TRACE 命令 474

REXX コマンド 519  
Evaluate 519  
IDIALLOC  
523  
IDIDDTEST  
527  
IDIDSECTdsn  
528  
IDIDSNTEST  
529  
IDIEventInfo  
530  
IDIFREE  
531  
IDIGET  
532  
IDIModQry  
532  
IDIPUT  
533  
IDIRegisterFaultEntry  
534  
IDIWRITE  
535  
IDIWTO  
537  
List 538  
注 540

RFR SDUMP データ・セット  
XFACILIT リソース・クラス 311

RFR TDUMP XFACILIT の例 313

RFR ダンプ・タイトル 59

RPTFIND コマンド 115

RUNCHAIN コマンド 115

## S

S913 異常終了 340

SAA オーバーレイ 183, 419

Saved Report 画面の例 125

SAY  
REXX 命令 474

SDUMP  
データ・セット名前パターン 342

SDUMP data set (DBD データ・セット)  
アクセスの管理 315

SDUMP SVC  
スクリーニング 403

SDUMP リカバリー障害記録データ・セット 311

SDUMPDSN の設定 342

SETFAULTPREFIX 制御ステートメント 446

SETMAXFAULTENTRIES 制御ステートメント 447

SETMINFAULTENTRIES 制御ステートメント 447

SETPROG コマンド  
モジュールの管理 459

SFA コマンド 331

SHOW コマンド 116

SIT コマンド 117

SLIP トラップ 306  
 SLIP,COMP=0C4 トラップ 59  
 SMF タイプ 89 レコード 56, 461  
 SMP/E APPLY 459  
 SMP/E RESTORE 459  
 snap dump メソッドからの  
 Fault Analyzer  
 の呼び出し  
 47  
 Snapdata オプション 628  
 Source オプション 629  
 Specify Compiler Listing or Side File 画面の例 240  
 SpinIDIREPRT  
 オプション  
 629  
 SQLCA 197, 633  
 STCK Conversion 画面の例 236  
 STCK コマンド 117  
 Storage Accounting Area (SAA) オーバーレイ 183, 419  
 storage areas  
 DSECT 情報を使用したマッピング 227  
 IDXLIST  
 関数を使用したレポート  
 557  
 System-Wide Information 203  
 関連しているものの表示 208  
 Storage Disassemble 画面の例 234  
 Storage DSECT Mapping Entry 画面の例 228  
 Storage DSECT Mapping Map Display 画面の例 228  
 Storage RUNCHAIN Command 入力画面の例 231  
 StoragePrintLimit オプション 630  
 StorageRange オプション 631  
 STORE CLOCK 値 (STCK コマンド)  
 変換 236  
 summarized CICS trace  
 System-Wide Information 185  
 Summarized CICS Trace 画面の例 185  
 SVC 51  
 スクリーニング 403  
 SVC インストール  
 IDICZSVC  
 による動的な実行  
 307  
 SVC ダンプ・スクリーニング 403  
 SVC ダンプ・データ・セット画面 251  
 SVC ダンプ出口 418  
 SVC ダンプ登録出口 303  
 SVC を動的に更新する 307  
 SVCDUMP ダンプ・タイトル 59  
 Synopsis 174  
 Synopsis 画面の例 174  
 SYSADATA DD ステートメント 377  
 SYSCONE MVS システム・シンボル  
 IDICNFxx  
 parmlib メンバー接尾部  
 372  
 SYSDEBUG ファイル  
 EQUAEDAT 出口を使用した検索 385  
 IGSZUXXB 出口を使用した検索 384  
 SYSLOG メッセージ  
 マイナーなものの抑止 376  
 SYSLOG 要約 38  
 SYSMDUMP  
 ASA プリンター制御文字 587  
 ダンプ・データ・セット画面 251  
 バッチ再分析の入力として指定 587

論理レコード長 587  
 System-Wide Information  
 :NONE. 194  
 CICS 182  
 summarized CICS trace 185  
 CICS 情報  
 CICS Trace Formatting 186  
 IMS 情報 197  
 Java Information 204  
 LE heap analysis 204  
 MTRACE Records 205  
 storage areas 203  
 ファイルを開く 179  
 メッセージ 194  
 System-Wide Messages 画面の例 194  
 System-Wide Open Files 画面の例 179  
 System-Wide Storage Areas 画面の例 203  
 SystemWidePreferred オプション 632, 634  
 SYSUDUMP SYSOUT クラス  
 IDIREPRT  
 の割り振り  
 37

## T

TACB  
 CICS での重複の判別 617  
 TCB 保護キーのサポート 27  
 TDUMP リカバリー障害記録データ・セット 311  
 TEST オプションの考慮事項 381  
 TEST(NONE,SYM,SEPARATE) COBOL コンパイラ・オプション 381, 384  
 疑似アセンブラー命令の表示に関する制約事項 214  
 TEST(STMT,SYM,NOHOOK,SEPARATE)  
 Enterprise PL/I コンパイラ・オプション 381, 384  
 TH タグ 551  
 title 116  
 TRACE  
 REXX 命令 474  
 try-catch ブロック  
 サンプル 273  
 TSO  
 REXX 環境制限事項 468

## U

U タグ 551  
 UFM データ域 673  
 UL タグ 552  
 User Note List 画面の例 221  
 User Notes Update プロンプトの例 222  
 USERMOD  
 IDISPLI  
 335  
 IDISPLIA  
 335  
 IDIWTSSEL  
 418  
 再適用 459  
 再表示 459  
 USRHDLR(CEEWUCHA)  
 特殊な処理 31  
 UTL データ域 687

## V

View List 画面の例 70  
 VIEWS コマンド 117

## W

WebSphere Liberty 262  
 WTO

宛先コード 31  
 記述子コード 31

## X

XCF 320, 365  
 XDUMP  
 Fault Analyzer  
 の検証  
 427  
 XDUMP data set (DBD データ・セット) 59  
 XFACILIT リソース・クラス 314  
 アクセスの管理 313  
 名前パターン 342  
 XDUREQ CICS グローバル・ユーザー出口 341  
 XEIIIN グローバル・ユーザー出口 419  
 XFACILIT インプリメンテーション例 367,  
 367, 368  
 XFACILIT リソース・クラス 314  
 IDLSDUMP\_ACCESS  
 47, 311  
 IDIHIST\_GROUP\_DSN  
 366  
 IDIHIST\_USERID\_DSN  
 366  
 IDIRFR\_TDUMP\_HLQ 312  
 Java 取り込み SDUMP データ・セットの管理 47  
 TDUMP RFR データ・セットの管理 312  
 ヒストリー・ファイル障害項目の管理 366  
 リカバリー障害記録 SDUMP データ・セットの管理 311  
 XMIT オプション画面の例 145  
 XPCABND CICS グローバル・ユーザー出口 405, 414  
 XPL データ域 689

## Z

z/OS  
 V2.2 より前のリアルタイム・ダンプ処理 36  
 サポートされるバージョン 27

## あ

アクセシビリティ機能 755  
 アセンブラー ADATA オプション 377  
 アセンブラー・プログラム  
 Fault Analyzer  
 の検証  
 427  
 アセンブラー呼び出し  
 IDISNAP  
 46  
 アセンブラー出口  
 CICS NoDup(CICSFAST) によるオーバーライド 414  
 アドレス・スペース  
 分析用の選択 253  
 アプリケーション・エラー処理  
 Fault Analyzer  
 呼び出しへの影響  
 305  
 アプリケーション固有の説明、次を提供 455

## い

イベント情報  
 IDIXEINF  
 関数を使用した取得  
 554  
 インストール・システム全般のデフォルト・オプション 372, 569  
 インストール・チェックリスト 292  
 インストール後のライブラリー名 296

インストール状況、検証 292

## え

エラー画面の例 64  
エラー処理による  
Fault Analyzer  
呼び出しへの影響  
305

## お

オブジェクト・コード  
DISASM コマンドによる逆アセンブル 234  
オプション  
\_IDL\_OPTS  
環境変数  
572  
\_IDL\_OPTSFILE  
環境変数  
571  
AdditionalIDIOffDD  
573  
BatchOpts 120, 599  
CallEqauDat 573  
CICS システム異常終了分析用の設定 251  
CICSDumpTableExclude 574  
CICSTraceMax 575  
CICSTranAnalysisUser 575  
DataSets 376, 576  
DeferredReport 584  
Detail 376, 586  
DumpDSN 587  
DumpRegistrationExits 588  
ErrorHandler 589  
Exclude 374, 590  
Exits 595  
FaultID 605  
GenerateSavedReport 606  
GlobalExclude 603  
HistCols 604  
IDICNFxx  
parmlib メンバー  
569  
IDIOPTS  
571  
Include 374, 590  
InteractiveExitPromptSeconds 605  
Java 分析用の設定 262  
JclCapture 607  
Language 607  
LangxCapture 608  
Locale 609  
LoopProtection 609  
MaxMinidumpPages 376, 610  
NoDup 611  
NoErrorHandler 589  
NoPrintInactiveCOBOL 624  
NoQuiet 624  
NoSource 629  
NoSpinIDIREPRT  
629  
PARM フィールド 572  
PDTCOptions 620  
PermitLangx 621  
PreferredFormattingWidth 623  
PrintInactiveCOBOL 624  
Quiet 624  
RefreshExits 625  
RetainCICSDump 626  
RetainDump 627  
Snapdata 628  
Source 629  
SpinIDIREPRT

629  
StoragePrintLimit 630  
StorageRange 631  
SystemWidePreferred 632, 634  
インストール・システム全般 569  
デフォルト  
see デフォルト・オプション  
バッチ再分析 154  
ユーザー・オプション・ファイル 571  
ユーザー・オプション・モジュール  
IDICNFUM  
570  
現在の画面 106  
現在の表示 102, 152  
言語 426  
構文規則 568  
指定 569  
処理順序 567  
静止 376  
設定 426, 567  
対話式再分析 162  
目的 567  
累積 568  
オプション変更 572  
オペレーター宛メッセージ  
see WTO

## か

カーソル一致 86  
カスタマイズ  
検証 427  
カスタマイズ・チェックリスト 292  
カスタマイズ済み ISPF インターフェース・フ  
ロントエンドの例 741

## き

キーワード、構文図 xii

## く

グローバル・リソースの逐次化 365  
グローバル環境データ域 (ENV) 467

## こ

コマンド  
CE 104  
CICSD 105  
CICSSTG 105  
CMDS 105  
COLS 106  
COPY 106  
CUROPTS 106  
DISASM 107  
DSECT 107  
DUPS 107  
EDIT 107  
EXEC 108  
FIND 108  
IDISINFO  
111  
INFO 111  
ISPF インターフェース 104  
JCL 111  
LOOKUP 112  
MATCH 113  
NEXT 113  
NOTE 113  
NOTELIST 114  
PREV 114  
QUIT 114  
REFRESH 115  
RESET 115  
RPTFIND 115

RUNCHAIN 115  
SHOW 116  
SIT 117  
STCK 117  
VIEWS 117  
構文図の読み取り xii  
コンパイラー・リスト  
プロンプト画面 239  
位置決め 384  
一時データ・セット禁止 391  
検索トレースの有効化 387  
障害再分析指定の例 161  
選択基準 31  
属性 391  
提供 377  
命名基準 382  
コンパイラー・リスト読み取りユーザー出  
口 480

## さ

サード・パーティー製品 305  
サイド・ファイル  
プロンプト画面 239  
リストに対する利点 30  
位置決め 384  
一時データ・セット禁止 391  
検索トレースの有効化 387  
選択基準 31  
属性 391  
定義 239  
提供 377  
命名基準 382  
サイド・ファイルの使用可能性テスト・ユー  
ティリティ 393  
サイド・ファイル検索からプログラムを除外  
する画面の例 243  
サブシステム  
IDIS  
319  
サンプル・データ・セット・メンバー  
IDISIDISAM1  
422  
IDICNFxx  
372  
IDIGSVRJ  
563  
IDIOPTLM  
338  
IDIPANEX  
336  
IDI  
S\$NDX  
27  
IDISCICS  
404  
IDISCNFU  
570  
IDISDB2B  
435  
IDISFE\*  
741  
IDISHIST  
345  
IDISISPF  
328  
IDISJAV1  
271  
IDISJAV2  
272  
IDISJAV3  
272



IDISJAV4  
272  
IDISJAV5  
273  
IDISJAV6  
273  
IDISPDM 335  
IDISPLI  
335  
IDISPLIA  
335  
IDISRC1  
430  
IDISROBT  
356  
IDISTSOB  
360  
IDISUFM3  
542  
IDISUFMn  
494  
IDISUFMX  
238  
IDISUSI  
299  
IDISUTL1  
516  
IDISVENU  
309  
IDISVJPN  
310  
IDISXNFY  
354  
IDISXPLA  
468  
IDISXPLB  
468  
IDISXPLC  
468  
IDISXPLP  
468  
IDIVPASM  
427, 427  
IDIVPBLE  
431  
IDIVPC  
430  
IDIVPCOB  
428, 431  
IDIVPDB2  
434  
IDIVPDBB  
435  
IDIVPPLE  
429  
IDIVPPLI  
429, 431  
IDIWCIDI 565  
IDIWTSEL  
418  
IDIXMIT  
361

## し

システム・ダンプの抑止  
ユーザー出口での制御 494  
システム間カップリング・ファシリ  
ティ 365  
システム間カップリング・ファシリティー  
(XCF) 320  
システム要件 27

シスプレックス  
ヒストリー・ファイルの共用 364  
シスプレックス全体にわたるサブシステムの  
相互通信 320  
シャットダウン PLT 402  
シンボル・サブストリングの指定 583

## す

ストレージ要件 297

## せ

セキュリティの考慮事項 146

## そ

ソース・コード (source code)  
表示する 212

## た

タイトル 743  
タグ  
see フォーマット・タグ  
ダンプ・ストレージ抑止画面の例 217  
ダンプ・データ・セット  
Fault Analyzer  
のサイズへの影響  
305  
関連付けられた 26  
名前、表示 89  
ダンプの登録  
ENV データ域内の呼び出し出口の指示 657  
Fault Analyzer  
IDIS  
サブシステムの使用  
319  
IDIXTSEL  
呼び出し出口  
303  
のユーザー出口の指定 588  
検証 439  
処理の説明 53  
障害項目の識別 75  
通知ユーザー出口 507  
分析制御ユーザー出口 479  
ダンプの抑止 33  
ユーザー出口での制御 494  
ダンプ状況  
available 139  
not found 139

## ち

チェックリスト  
Fault Analyzer  
のインストール  
292  
Fault Analyzer  
のカスタマイズ  
292

## て

データ・セット  
Compiler Listing  
一時データ・セット制限 391  
DDname からの廃棄 580  
ISPF  
see ISPF データ・セット  
サイド・ファイル (side file)  
一時データ・セット制限 391  
バッチ再分析 159  
再分析 579  
対話式再分析 244  
データ・セット・セキュリティ・サー  
バー・アクセス要件  
IDIDOC

310  
IDIVSENU  
310  
IDIVSJPN  
310  
データ・セット・メンバー  
see サンプル・データ・セット・メン  
バー  
データ・セットのステージング  
IDIROBOT  
サンプル EXEC  
353  
IMPORT 制御ステートメント 448  
データ・セット名の置換シンボル 580  
データ・セット名内のシンボル 476, 580  
データ・セット名内の置換シンボル 476, 580  
データ域  
10 進数からの変換 555  
16 進数からの変換 556  
チェーンの表示 231  
表示する 217  
テーブル表示  
ソート 257  
一致 257  
デフォルト・オプション 372

## と

トランザクション異常終了制御ブロック  
see TACB  
トランザクション異常終了分析  
CICS  
see CICS

## は

バージョンとメンテナンス・レベル, 表示 131  
バインド関連の従属関係 30  
バック・データ・フォーマット  
ISPF  
see ISPF バック・データ・フォーマッ  
ト  
バッチ TSO 受信、サンプル EXEC 353  
バッチ・オプション行 155  
バッチ・ユーティリティ  
IDIUTIL  
441  
バッチ再分析 24  
JCL 制御ステートメント 332  
オプション 154  
ジョブの実行依頼 87  
データ・セット 159  
開始 159  
保管レポートの表示 125  
目的 154  
戻りコード 693

## ひ

ヒストリー・ファイル  
IDIUTIL  
による管理  
441  
PDS と PDSE の間での変換 347  
PDSE 管理の 345  
アクセスの管理 365  
アクセス情報のリセット 98  
コンテンツ 25  
サイズ 349  
サイズの決定 345  
シスプレックス全体での共用 364  
ダンプ・データ・セット名の表示 89  
デフォルト名 349  
バッチ・ダンプ再分析の実行依頼 87  
画面の選択 66

- 割り振り 345
- 管理されている 148
- 共用 DASD のない MVS システム全体での管理 353
- 項目のコピー 88
- 項目の移動 88
- 項目の削除 88
- 項目の送信 89
- 最新表示 98
- 自動管理 347
- 重複ヒストリー情報の表示 88
- 除外された 149
- 障害
  - see 障害ヒストリー・ファイル
  - 障害の突き合わせ 83
  - 障害項目の削除 131
  - 設定 345
  - 選択された 150
  - 対話式再分析の開始 88
  - 非 ISPF インターフェースによるアクセス 286
  - 保管された障害分析レポートの表示 89
  - 名前の設定 349, 579
  - 名前変更 345
  - 論理サイズ 347
- ヒストリー・ファイル更新
  - ユーザー出口での制御 494, 498
- ヒストリー・ファイル項目
  - 移動 145
  - 移動範囲 145
  - 作成 259
- ヒストリー・ファイル障害項目
  - XFACILIT リソース・クラス 366
  - 作成 262
- ヒストリー・ファイル設定
  - 変更の制限 308
- ヒストリー・ファイル選択
  - ユーザー出口での制御 494
- ヒストリー項目
  - 障害
    - see 障害ヒストリー項目
- ビュー 66
  - 画面の選択 66
  - 設定 350
- ビューに関する考慮事項
  - XFACILIT リソース・クラスを使用しない 352

## ふ

- ファイルを開く
  - System-Wide Information 179
- フォーマット・タグ
  - ADDR (アドレス) 544
  - AREA (領域) 544
  - DATA (データ) 545
  - DD (定義説明) 545
  - DL (定義リスト) 545
  - DT (定義語) 546
  - DUMP (EBCDIC ダンプ) 547
  - DUMPA (ASCII ダンプ) 548
  - HP (強調表示されたフレーズ) 549
  - L (行) 549
  - LI (リスト項目) 549
  - NOTEL (注釈リスト) 550
  - P (パラグラフ) 550
  - TH (表の見出し) 551
  - U (下線) 551
  - UL (順序なしリスト) 552
- フォーマット・ユーザー出口 491
  - サンプル 494
- フォーマット済みテキスト

- IDIXWRIT
  - 関数を使用したレポートへの書き込み 559
- フラグメント、構文図 xii
- ブランク行
  - 除去 127
  - 追加する 127
- ブリファレンス、Fault Analyzer 117
- プログラム SNAP インターフェース 39
- プログラムの説明
  - ユーザー定義 457
- プログラム制御アクセス
  - 定義 308
- プロンプト
  - 制御 244

## へ

- ヘルプ・テキスト
  - 除去 127
  - 追加する 127

## ほ

- ポイント・アンド・シュート方式のフィールド 172

## み

- ミニダンプ
  - concept 20
  - サイズの制限 376

## め

- メッセージ
  - Fault Analyzer 695
  - IDISF\* 695
  - IPVLANGX 695
  - System-Wide Information 194
  - 拡張 212
- メッセージおよび異常終了コード説明ユーザー出口 485
- メッセージの説明
  - ユーザー定義 455
  - リボジトリ 309
  - 画面の例 212

## ゆ

- ユーザー・オプション・ファイル 568, 571
- ユーザー・オプション・モジュール
- IDICNFUM 570
- ユーザー・ノート 207
  - リカバリー 152
  - 管理 218
  - 作成 218
  - 分割画面の制限 63
- ユーザー固有レポートをフォーマットする (EXEC コマンド) 237
- ユーザー出口 464
  - CICS システム異常終了分析 251
  - IDIUTIL ListHF 512
  - IDIUTIL ListHFDUP 517
  - IDIUTIL インポート 508
  - IDIUTIL 削除

- 511
- コンパイラー・リスト読み取り 480
- サポートされるプログラム言語 467
- データ域のバージョンの検査 469
- トレース 469
- フォーマット 491
- メッセージおよび異常終了コードの説明 485
- 起動パラメーター 467
- 終了処理 494
- 終了処理 (障害項目リフレッシュ) 498
- 除去 598
- 通知 499
- 通知 (MVS SVC ダンプ登録) 507
- 複数の出口タイプの場合 468
- 分析制御 475
- 分析制御 (MVS SVC ダンプ登録) 479
- 返されたデータ・セット名内の置換シンボル 476
- ユーザー出口タイプに固有のデータ域 467
- ユーザー定義
  - プログラムの説明 457
  - メッセージの説明 455
  - 異常終了コードの説明 456
- ユーザー定義メッセージの説明
  - 表示する 128
- ユーザー変更
  - parmlib データ・セット名 372

## ら

- ライセンス情報
  - 表示する 131
- ラッパー・ユーティリティ
  - サンプル 273

## り

- リアルタイム SNAP 分析 22
- リアルタイム・レポートの SYSOUT クラス
  - 制御 37
- リアルタイム異常終了分析 20
- リアルタイム分析 33
  - オプションによる制御 35
  - キャプチャーされた JCL の表示 88
  - デバッグ情報の表示 87
- リアルタイム分析レポート 37
  - SYSOUT クラスの変更 37
  - 結合 37
  - 動的
    - IDIREPRT
    - 割り振り解除の防止 629
    - 動的割り振り 37
    - 保管の表示 125
    - 抑止 38
- リカバリー障害記録 311
  - Fault Analyzer
  - IDIS
  - サブシステムの使用 319
  - デフォルトの IEATDUMP データ・セット名の変更 341
  - 概要 57
  - 設定の確認 438
- リカバリー障害記録データ・セット・アクセス
  - 管理 310
- リスト
  - ポインティング 376
- リスト/サイド・ファイルの不一致画面の例 241
- リフレッシュ処理 244

## れ

- レベル 88 項目
  - 縮小表示 210
- レベル 88 項目画面の例 211
- レポート
  - Java API によるダウンロード 744
  - リアルタイム
    - see リアルタイム・レポート
  - 内容 275
  - 表示域の拡大 332
- レポート・セクション
  - Abend Job Information 284
  - Epilog 284
  - Event Details 282
  - main 278
  - Prolog 278
  - summary 278
  - Synopsis 278
  - System-Wide Information 284
  - オプション 284
- レポート・ユーザー出口
  - see 分析制御ユーザー出口
- レポートの詳細レベル
  - 制御 376
- レポート例 284

## ろ

- ロード・モジュール
  - CA-Panexec から取得 336
- ロケール名 645
- ロック・フラグ 100, 136, 661
- ロック・フラグ値のデフォルト設定 117

## わ

- ワイルドカード
  - 障害ヒストリー項目の突き合わせ 86