

Automatic Binary Optimizer for z/OS
バージョン 2 リリース 1

ユーザーズ・ガイド



注記

本書および本書で紹介する製品をご使用になる前に、[85 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® Automatic Binary Optimizer for z/OS® (プログラム番号 5697-AB2) および IBM Automatic Binary Optimizer for z/OS Trial (プログラム番号 5697-TR2) のバージョン 2.1 に適用され、新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。製品のレベルに合った正しい版をご使用ください。

www.ibm.com/shop/publications/order/ においてソフトコピー資料を無償で表示またはダウンロードできます。

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

原典：

SC27-9587-00
Automatic Binary Optimizer for z/OS
Version 2 Release 1
User's Guide
First edition (September 2019)

発行：

日本アイ・ビー・エム株式会社

担当：

トランスレーション・サービス・センター

© Copyright International Business Machines Corporation 2015, 2019.

目次

表.....	vii
前書き.....	ix
本書について.....	ix
略語.....	ix
構文図の読み方.....	ix
変更の要約.....	x
ご意見の送付方法.....	x
Automatic Binary Optimizer for z/OS のアクセシビリティ 機能.....	xi
第 1 章概説.....	1
利点.....	1
ABO と Enterprise COBOL の連携.....	1
第 2 章システム要件.....	3
サポート対象のオペレーティング・システム.....	3
ターゲット・ハードウェア・レベル.....	4
第 3 章 COBOL モジュール要件.....	7
適格なコンパイラ.....	8
COBOL 言語機能およびコンパイラ・オプションのサポート.....	8
不適格 CSECT の処理.....	9
第 4 章インストール、およびインストールの検査.....	11
IBM Automatic Binary Optimizer for z/OS のインストール.....	11
インストール検査プログラム (IVP) を使用してインストールを検査.....	11
第 5 章モジュールの最適化.....	15
必須 DD ステートメント.....	15
オブティマイザー・ディレクティブ.....	16
BOPT.....	16
IEFOPZ.....	18
オブティマイザー・オプション.....	19
ALLOW.....	20
ARCH.....	20
CSECT.....	21
LIST.....	23
LOG.....	24
REPLACE.....	25
SCAN.....	26
コメント.....	26
行の継続.....	27
JCL の例.....	28
BOPT で最適化を指定.....	28
IEFOPZ で最適化を指定.....	31
z/OS JCL REGION パラメーターおよび JCL MEMLIMIT パラメーターの推奨設定.....	32
ABO メッセージに使用される言語を指定.....	33
TSO、REXX、およびアセンブラー・コードからの ABO の起動.....	33
TSO のもとでの最適化.....	33

アセンブラー・プログラムからオブティマイザーを開始する.....	38
第 6 章最適化処理からの出力を理解.....	41
ログ・ファイル.....	41
リスト変換.....	43
リスト変換の内容.....	43
SYSPRINT DD および LIST オプション.....	49
第 7 章最適化の管理、および最適化されたモジュール・デプロイメント・プロセスの管理	53
最適化およびデプロイメントの使用法のシナリオ.....	54
シナリオ 1: 静的デプロイメントによる最適化プロセス.....	54
シナリオ 2: 動的デプロイメントによる最適化プロセス.....	55
シナリオ 3: ハイブリッド・アプローチによる最適化プロセス.....	56
テスト情報.....	57
第 8 章最適化に関する問題と、最適化済みモジュールのデプロイメントに関する問題を解決.....	59
最適化時に発生した問題を解決.....	59
実行時に発生した問題を解決	59
最適化後の COBOL モジュール・サイズの変更.....	60
エラー・メッセージおよび異常終了コードの相違	60
Application Delivery Foundation for z Systems	61
Run Time Instrumentation Profiler.....	61
付録 A JCL サンプル.....	65
付録 B 戻りコード.....	67
付録 C メッセージ	69
BOZ1003U.....	69
BOZ1031S.....	69
BOZ1145U.....	69
BOZ1400S.....	70
BOZ1401S	70
BOZ1402S.....	70
BOZ1403S.....	70
BOZ1404S.....	70
BOZ1405S.....	70
BOZ1406S.....	71
BOZ1407S.....	71
BOZ1408S.....	71
BOZ1409S.....	71
BOZ1410I.....	71
BOZ1411S.....	72
BOZ1412S.....	72
BOZ1413S.....	72
BOZ1414S.....	72
BOZ1415S.....	72
BOZ1416S.....	73
BOZ1417S.....	73
BOZ1418S.....	73
BOZ1419S.....	73
BOZ1420S.....	73
BOZ1421S.....	74
BOZ1422S.....	74

BOZ1423S.....	74
BOZ1424S.....	74
BOZ1428U	75
BOZ1429U.....	75
BOZ1430U.....	75
BOZ1431S.....	76
BOZ1432S	76
BOZ1436S.....	76
BOZ1437S.....	76
BOZ1438U.....	77
BOZ1439U.....	77
BOZ1446U.....	77
BOZ1447U.....	77
BOZ1449U.....	77
BOZ1450U.....	78
BOZ1451S.....	78
BOZ1452S.....	78
BOZ1453U.....	78
BOZ1455W.....	78
BOZ1456S.....	79
BOZ1457S.....	79
BOZ1490W.....	79
BOZ1491W.....	79
BOZ1492W	80
BOZ1493S.....	80
BOZ1494S.....	80
BOZ4089I.....	81
BOZ4091I.....	81
BOZ4092I.....	81
BOZ4097I.....	81
BOZ4101W.....	81
BOZ4107I.....	81
BOZ4109I.....	82
BOZ4110I.....	82
BOZ4111I.....	82
BOZ4113I.....	83
BOZ4114I.....	83
BOZ4116I.....	83
BOZ4117I.....	83
BOZ4119S.....	83
BOZ4120S.....	84
BOZ4121S.....	84
BOZ4124I.....	84

特記事項.....	85
商標.....	85
資料名リスト.....	87
IBM Automatic Binary Optimizer for z/OS 資料.....	87
関連資料.....	87

表

1. オプティマイザーとコンパイラーにおけるユース・ケースの比較.....	2
2. サポート対象のハードウェア・レベル.....	4
3. ABO がサポートしない COBOL モジュール.....	7
4. 不適格な CSECT および発行されるメッセージ.....	9
5. 戻りコードおよび対応する欠落 LE PTF.....	13
6. バイナリー最適化に使用される DD 名.....	15
7. オプティマイザー・オプション.....	19
8. 入力モジュールと、それらに含まれている CSECT.....	50
9. 出力 1: 最適化されたモジュールと、それらの CSECT.....	50
10. 出力 2: リスト変換.....	50
11. 推奨される割り振りパラメーター.....	63
12. 推奨される割り振りパラメーター.....	65
13. IBM Automatic Binary Optimizer for z/OS 戻りコード.....	67

前書き

本書について

本書は、コンパイル済みの COBOL プログラムのパフォーマンスを IBM Automatic Binary Optimizer for z/OS によって向上させたい、IBM COBOL コンパイラーを使用しているお客様を対象にしています。

略語

本書では、短縮形で使用される用語があります。最も頻繁に使用される用語の省略形を下の表にアルファベット順にリストします。

使用される用語	長形式
ABO	IBM Automatic Binary Optimizer for z/OS
CSECT	Control section
EBCDIC	Extended binary coded decimal interchange code
HFS	Hierarchical file system
JCL	Job control language
PDS	Partitioned data set
PDSE	Partitioned data set extended

その他の用語で、一般に理解されていないものは、初出時に略さずに記載されます。

構文図の読み方

本書中の構文図を読むには、以下の説明を参照してください。

- 構文図は、左から右、上から下へと線をたどってください。
 - >>--- 記号は、構文図の開始を示します。
 - > 記号は、構文図が次の行に続くことを示します。
 - >--- 記号は、構文図が前の行からの続きであることを示します。
 - >< 記号は、構文図の終わりを示します。

完全なステートメントではない構文単位の図は、>--- 記号で始まり、---> 記号で終わります。

- 必須項目は、横線 (幹線) 上に示されています。

▶▶ required_item ◀◀

- 任意指定項目は、幹線の下に示されています。

▶▶ required_item — optional_item —▶▶

- 複数の項目から選択できる場合には、それらの項目は縦方向に重ねて示されます。それらの項目のうち 1 つを選択しなければならない場合には、それらの項目のうちの 1 つが幹線上に示されます。



項目のうちの1つがデフォルトであれば、それがメインパスより上に示され、残りの選択項目はメインパスより下に示されます。



- 幹線の上を左に戻る矢印は、その項目を繰り返して指定できることを示しています。



- キーワードは大文字で示されています (例えば、FROM)。これは表記どおり正確につづらなければなりません。変数は日本語または小文字で示されます (例えば *column-name*)。それらの変数は、ユーザーが指定する名前や値を表します。
- 句読記号、括弧、算術演算子などの記号が示されている場合には、これらも構文の一部として入力しなければなりません。

変更の要約

このセクションでは、IBM Automatic Binary Optimizer for z/OS バージョン 2 リリース 1 に関して本書で行われた主な変更をリストします。最新の技術的な変更は、HTML 版では強調表示され、PDF 版では左側の余白にある縦棒 (|) で示されています。

バージョン 2 リリース 1

- z/OS バージョン 2.4 がサポート対象オペレーティング・システムのリストに追加され、z/OS バージョン 2.1 が削除されました。(「サポート対象のオペレーティング・システム」を参照してください。)
- ARCH(13) が追加されました。これは、新規 IBM z15 メインフレームが活用されるコードを生成するためのものです。(「ARCH」を参照してください。)
- メッセージ BOZ4124I が追加されました。(「メッセージ」を参照してください。)
- オプティマイザー・オプション HANDLERS が削除されました。

ご意見の送付方法

- www.ibm.com/software/awdtools/rcf/ にある Online Readers' Comments Form をご使用ください。
- 電子メールを利用される場合、アドレスは compinfo@cn.ibm.com です。

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

Automatic Binary Optimizer for z/OS のアクセシビリティ機能

アクセシビリティ機能は、運動や視覚などに障害を持つユーザーが情報技術製品を快適に使用できるように支援します。z/OS のアクセシビリティ機能には、Automatic Binary Optimizer for z/OS (ABO) に対するアクセシビリティが備わっています。

アクセシビリティ機能

z/OS には、以下の主要アクセシビリティ機能が含まれています。

- スクリーン・リーダーおよび画面拡大機能ソフトウェアで一般的に使用されるインターフェース
- キーボードのみによるナビゲーション
- 色、コントラスト、フォント・サイズなど表示属性のカスタマイズ機能

z/OS では、[US Section 508](http://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>) および [Web Content Accessibility Guidelines \(WCAG\) 2.0](http://www.w3.org/TR/WCAG20/) (<http://www.w3.org/TR/WCAG20/>) に確実に準拠するために、最新の W3C 標準である [WAI-ARIA 1.0](http://www.w3.org/TR/wai-aria/) (<http://www.w3.org/TR/wai-aria/>) を使用します。アクセシビリティ機能を利用するには、最新リリースのスクリーン・リーダーを、この製品でサポートされる最新の Web ブラウザーと併用してください。

[IBM Knowledge Center](http://www.ibm.com/support/knowledgecenter/en/about/releasenotes.html) にある ABO オンライン製品資料はアクセシビリティに対応しています。IBM Knowledge Center のアクセシビリティ機能については、<http://www.ibm.com/support/knowledgecenter/en/about/releasenotes.html> に説明があります。

キーボード・ナビゲーション

ユーザーは、TSO/E または ISPF を使用して z/OS ユーザー・インターフェースにアクセスできます。

また、ユーザーは IBM Developer for z Systems® Enterprise Edition を使用することで z/OS サービスにもアクセスできます。

上記インターフェースへのアクセスについては、以下の資料を参照してください。

- [z/OS TSO/E Primer](http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.ikjp100/toc.htm) (http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.ikjp100/toc.htm)
- [z/OS TSO/E User's Guide](http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.ikjc200/toc.htm) (http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.ikjc200/toc.htm)
- [z/OS ISPF User's Guide Volume I](http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.f54ug00/toc.htm) (http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.f54ug00/toc.htm)
- [IBM Developer for z Systems Knowledge Center](http://www.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html?lang=en) (http://www.ibm.com/support/knowledgecenter/SSQ2R2/rdz_welcome.html?lang=en)

上記の資料には、キーボード・ショートカットまたはファンクション・キー (PF キー) の使用方法を含む TSO/E および ISPF の使用方法が記載されています。それぞれの資料では、PF キーのデフォルトの設定値とそれらの機能の変更方法についても説明しています。

インターフェース情報

ABO オンライン製品資料は [IBM Knowledge Center](https://www.ibm.com/support/knowledgecenter/SSERQD) (<https://www.ibm.com/support/knowledgecenter/SSERQD>) で入手できます。この資料は標準の Web ブラウザーで表示できます。

PDF ファイルでのアクセシビリティ・サポートは限定的です。PDF 資料では、オプションのフォント拡大機能およびハイコントラスト表示設定を使用でき、キーボードのみでナビゲートできます。

スクリーン・リーダーで、ピリオドやコンマなどの PICTURE 記号を含む構文図、ソース・コード例、およびテキストを正確に読み上げるには、すべての句読点を読み上げるようにスクリーン・リーダーを設定する必要があります。

支援技術製品は、z/OS に備わっているユーザー・インターフェースと連動します。特定のガイダンス情報については、z/OS インターフェースへのアクセスに使用する支援技術製品の資料を参照してください。

関連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 電話サービスを立ち上げました。

TTY サービス
800-IBM-3383 (800-426-3383)
(北米内)

IBM およびアクセシビリティ

IBM のアクセシビリティへの取り組みについて詳しくは、[IBM Accessibility \(www.ibm.com/able\)](http://www.ibm.com/able) を参照してください。

第 1 章 概説

IBM Automatic Binary Optimizer for z/OS は、コンパイル済みの IBM COBOL プログラムのパフォーマンスを向上させます。ABO は、ソース・コード、ソース・コード・マイグレーション、およびパフォーマンス・オプション調整のいずれも必要としません。これは、新しい最適化テクノロジーを使用して最新の IBM Z メインフレーム (IBM z15 など) をターゲットにし、COBOL アプリケーションのパフォーマンスを向上させます。

ABO への入力となるのは、COBOL プログラム・モジュールです。ABO はプログラム・モジュールをスキャンして最適化対象の適格な COBOL プログラムを検索します。以下の状態がすべて当てはまる場合、プログラムは ABO による最適化に適格です。

- プログラムは適格な IBM COBOL コンパイラによって生成された。
- すべての言語機能がサポートされている。
- すべての最適化検査にパスした。

ABO は、入力プログラムの検証後にプログラム・モジュールを処理し、最適化済みプログラム・モジュールを生成して、最新の IBM Z メインフレームをターゲットにします。

ABO や最適化済みプログラムを実行するには、z/OS バージョン 2.2 以降が必要です。最適化済みプログラムを、アプリケーション JCL を変更することなく複数レベルの z/Architecture® で実行できます。

利点

ABO によって作成された最適化済みプログラムは、最新 IBM Z メインフレームの機能を活用してパフォーマンスを向上させ、処理コストを減らし、プログラムの実行時間を短縮します。

ABO は最新の拡張 COBOL 最適化テクノロジーを利用して、パフォーマンスを向上させるための処理能力を提供する、最新のデプロイメント・システムをターゲットにしたコードを生成します。

以前の COBOL コンパイラはコードを ARCH(0) レベルでのみ生成していました。Automatic Binary Optimizer for z/OS を使用すれば、最新の ARCH(13) z15 メインフレーム、ARCH(12) z14 メインフレーム、ARCH(11) z13s®/ z13® メインフレーム、および ARCH(10) zEC12/zBC12 メインフレームを活用するように、ARCH(0) レベル COBOL アプリケーションがアップグレードされます。

Automatic Binary Optimizer を以前の COBOL プログラムで使用すると、ハードウェア・テクノロジーの進化を 25 年分飛び越えて、z15 メインフレーム、z14 メインフレーム、z13s メインフレーム、z13 メインフレーム、zEC12 メインフレームおよび zBC12 メインフレームでは既に実装されている 600 個超の新規ハードウェア命令にアクセスできるようになります。

ABO と Enterprise COBOL の連携

IBM Automatic Binary Optimizer for z/OS と最新の IBM Enterprise COBOL コンパイラは、さまざまな、ただし補足的な目的に役立ちます。このセクションでは、ABO を使用するタイミング、および最新の IBM Enterprise COBOL コンパイラでソース・コードを再コンパイルすることでパフォーマンスを向上させるタイミングについて説明します。

ご使用のプログラムのパフォーマンスを向上させるには、以下のいずれかの方法を選択してください。

- プログラム・ソース・コードを再コンパイルする場合は、IBM Enterprise COBOL for z/OS V5 または V6 を使用する
- 再コンパイル計画にないコンパイル済みプログラムを最適化する場合、またはプログラム・ソース・コードがない場合は、IBM Automatic Binary Optimizer for z/OS を使用する

2 ページの表 1 に従って、どちらを使用するかを選択してください。

表 1. オプティマイザーとコンパイラーにおけるユース・ケースの比較		
ユース・ケース	IBM Automatic Binary Optimizer for z/OS	IBM Enterprise COBOL コンパイラー
ソース、マイグレーション、オプションのいずれの調整も必要としない、大きなパフォーマンス向上	√	
複数のハードウェア・レベルをターゲットにするために z/OS V2.2 以降で組み込まれたサポート	√	
相互運用性およびレガシー互換性 (例: PDS 入出力、OS/VS COBOL および VS COBOL II NORES との相互運用)	√	
災害復旧マシンに合わせるために ARCH 設定をダウングレードする必要がない	√	
新規 COBOL アプリケーション開発、または新規 COBOL 機能の使用		√
既存の COBOL プログラムでの保守		√
最大のパフォーマンス向上 (ソース、マイグレーション、およびオプションの調整が必要)		√

第 2 章 システム要件

サポート対象のオペレーティング・システム

IBM Automatic Binary Optimizer for z/OS は、以下のオペレーティング・システムで実行できます。

- z/OS バージョン 2.4
- z/OS バージョン 2.3
- z/OS バージョン 2.2

z/OS 2.4 より前のオペレーティング・システム・レベルの場合、ABO に関しては、Automatic Binary Optimizer for z/OS がインストールされていて実行されているシステムにおいて、いくつかの PTF が適用されていなければなりません。ABO によって生成された最適化済みモジュールが実行されるシステムでは、そのシステムに ABO がインストールされなくても、他の PTF が必要になります。ABO、または ABO によって生成されたモジュールが実行されるシステムでは、APAR/PTF (OA47689/UA90982) が必要です。この APAR/PTF は、z/OS 2.2 に関してのみ使用可能です。

ABO が実行されるシステムでは、以下の PTF が必要です。

- z/OS V2.3
 - OA55985/UA97356 (バインダー)
- z/OS V2.2
 - OA47829/UA78084 (バインダー)
 - OA50640/UA82866 (バインダー)
 - OA47689/UA90982 (IEFOPZxx SYS1.PARMLIB サポート)
 - OA55985/UA97372 (バインダー)

ABO 最適化済みモジュールが実行されるシステムでは、以下の PTF が必要です。

- z/OS V2.3
 - PH14705/UI64417 (Language Environment Automatic Binary Optimizer ランタイム・エンジン)
- z/OS V2.2
 - PI51546/UI33445 (Language Environment)
 - PI51802/UI32944 (Language Environment CICS® システム 定義サンプル 更新)
 - OA47689/UA90982 (IEFOPZxx SYS1.PARMLIB サポート)
- PH14705/UI64419 (Language Environment Automatic Binary Optimizer ランタイム・エンジン)

ABO と ABO 最適化済みモジュールの実行に同じシステムが使用される場合は、z/OS バージョンごとに上記リストの PTF をすべてそのシステムにインストールする必要があります。

ABO と一緒に使用できる任意指定のプログラム

- [Application Delivery Foundation for z Systems V3.1](#)
 - Developer for z Systems Enterprise Edition V14.1
 - Debug for z Systems V14.1
 - Fault Analyzer for z/OS V14.1
 - Application Performance Analyzer for z/OS V14.1

最新の IBM Automatic Binary Optimizer または IBM Automatic Binary Optimizer Trial PTF をインストールすることを強くお勧めします。 [フィックス・リストおよび新機能のページ](#)を参照してください。

ターゲット・ハードウェア・レベル

IBM Automatic Binary Optimizer for z/OS は、最新の IBM Z サーバー用のプログラム・モジュールを生成できます。

Automatic Binary Optimizer for z/OS は、COBOL コンパイラと同じハードウェア番号付けスキームを使用します。IBM Automatic Binary Optimizer for z/OS バージョン 2.1 でサポートされるハードウェア・レベルが、4 ページの表 2 にリストされています。[ARCH オプション](#)を使用すれば、ABO 生成モジュールでターゲットにしたいハードウェア・レベルを指定できます。

表 2. サポート対象のハードウェア・レベル	
ハードウェア・レベル	説明
10	z/Architecture モードの 2827-xxxx (IBM zEnterprise® EC12) モデルおよび 2828-xxxx (IBM zEnterprise BC12) モデルで有効な命令を使用するコードを生成します。 具体的には、これらのレベル 10 マシンおよびその後継マシンには、以下の機能のサポートを使用する命令が追加されています。 • 実行ヒント機能 • ロード・アンド・トラップ機能 • 各種命令拡張機能 • トランザクション実行機能 • ゾーン 10 進数データ項目と 10 進浮動小数点データ項目間のより効率的な変換を可能にする、拡張 10 進浮動小数点機能。算術演算を実行するためにゾーン 10 進数データ項目をパック 10 進数データ項目に変換する代わりに、コンパイラはゾーン 10 進数データ項目を 10 進浮動小数点データ項目に直接変換し、計算の完了後に再度ゾーン 10 進数データ項目に戻します。
11	z/Architecture モードの 2964-xxxx (IBM z13®) および 2965-xxx (IBM z13s) モデルで有効な命令を使用するコードを生成します。 具体的には、これらのレベル 11 マシンおよびその後継マシンには、次の機能のサポートを使用する命令が追加されています。 • パック 10 進数データ項目と 10 進浮動小数点中間結果データ項目間のより効率的な変換を可能にする、拡張 10 進浮動小数点機能
12	z/Architecture モードの 3906-xxx (IBM z14) および 3907-xxx (IBM z14 ZR1) モデルで利用できる命令が使用されるコードが生成されます。 具体的には、レベル 12 マシンおよびその後継マシンでは、ベクトル・パック 10 進数機能をサポートする命令が追加されます。ベクトル・パック 10 進数機能では、中間結果がメモリーではなくベクトル・レジスターに格納されるためパック 10 進数の計算が高速になっています。

表 2. サポート対象のハードウェア・レベル (続き)	
ハードウェア・レベル	説明
13	z/Architecture モードの 8561-xxx (IBM z15) モデルで利用できる命令を使用するコードが生成されます。 具体的には、これらのレベル 13 マシンおよびその後継マシンには、以下の機能でサポートされる命令が追加されています。 • ベクトル・パック 10 進数拡張機能 • ベクトル拡張機能 2 • その他の命令拡張機能 3 • 調整済みベクトル・ロード/保管のヒント
注: ABO は、z/OS レベルでサポートされている任意のシステムで実行できます。z/OS V2.2 以降をサポートする IBM Z サーバーの完全なリストについては、「z/OS Server Support」を参照してください。	

第 3 章 COBOL モジュール要件

IBM Automatic Binary Optimizer for z/OS は、バインダーからのプログラム・モジュール出力と、リンケージ・エディターからのロード・モジュール出力を最適化します。バインダーからのプログラム・モジュール出力はプログラム・オブジェクトかロード・モジュールです。リンケージ・エディターによって生成されたロード・モジュールは、ABO がそのロード・モジュールを最適化できるようにバインダーに対して受け入れ可能な入力でなければなりません。

ABO は、完全にバインドされたプログラム・モジュールも部分的にバインドされたプログラム・モジュールも最適化できます。部分的にバインドされたモジュールとは、CALL=NO または NCAL オプションでバインドされたモジュールのことです。多くの場合、このモジュールはリンク・ライブラリーに含まれています。ALLOW=NOUNRESEXE オプションが指定されている場合、ABO は、部分的にバインドされたプログラム・モジュールを最適化しません。詳しくは、[ALLOW オプション](#)を参照してください。

ABO は以下の COBOL モジュールをサポートしません。

表 3. ABO がサポートしない COBOL モジュール	
ABO がサポートしない COBOL モジュール	発行されるメッセージ
バインダーが処理しないモジュール	BOZ4116I と、その後に発行される BOZ1429U 例えば、バインダーは、無効な文字が含まれている ESD 名を持つロード・モジュールを処理しません。ABO は、無効な ESD 名を持つモジュールを検出すると、BOZ4116I メッセージを生成します。BOZ4116I メッセージにはバインダー・メッセージ IEW2512E のテキストが含まれています。IEW2512E のテキストには、無効な ESD 名が含まれています。
EDIT=NO でバインドされたモジュール	BOZ1423S
「SIGN 対象」モジュール	BOZ1424S
実行不可としてマークされたモジュール	BOZ1422S
プリリンク・ステップからオブジェクト・ファイルを含んでいるがプリリンク・ステップが正しく実行されなかったモジュール	BOZ4116I と、その後に発行される BOZ1419S 例えば、一部のオブジェクト・ファイルに対して実行されなかったプリリンクは不適切である可能性があります。このようにプリリンクが不適切だと、バインダーや ABO はロード・モジュールを処理しません。バインダーがモジュールを処理しない場合、ABO は BOZ4116I メッセージを発行し、続いて BOZ1419S メッセージを発行します。モジュールに複数のプリリンク・ステップの出力が含まれている場合も、プリリンクは不適切です。この場合、通常、オリジナルのモジュールは正しく実行されず、ABO は、正しく実行されないモジュールを生成します。

ABO はプログラム・モジュール内の CSECT をスキャンして、最適化に適格な CSECT を検索します。CSECT が ABO による最適化に適格となるのは、その CSECT が適格な COBOL コンパイラーによって生成されたものであり、オリジナルの COBOL プログラムで使用されているすべての COBOL 機能が ABO によってサポートされている場合です。

適格なコンパイラー

IBM Automatic Binary Optimizer for z/OS V2.1 は、以下の COBOL コンパイラーで生成されたプログラム・モジュール内で CSECT を最適化できます。

- Enterprise COBOL for z/OS V4
- Enterprise COBOL for z/OS V3
- COBOL for OS/390® & VM V2
- COBOL for MVS™ & VM V1.2
- COBOL/370 V1.1
- VS COBOL II V1.4.0 (LE 対応モジュールのみ)
- VS COBOL II V1.3.x (LE 対応モジュールのみ)

注：CA オプティマイザーによって処理された COBOL モジュールは、ABO では最適化できません。このようなタイプのモジュールに関しては、COBOL コンパイラーで作成されたオリジナルのモジュールを、CA オプティマイザーで処理する前に ABO で最適化することをお勧めします。

COBOL 言語機能およびコンパイラー・オプションのサポート

サポートされている COBOL 言語機能およびコンパイラー・オプション

IBM Automatic Binary Optimizer for z/OS の使用によって、ほとんどの COBOL 言語機能がサポートされます。

IBM Automatic Binary Optimizer for z/OS バージョン 2.1 でサポートされる、主要な COBOL 機能のリスト：

- ARITH(EXTEND | COMPAT)
- CICS
- CICS HANDLE ABEND
- CICS HANDLE AID
- CICS 言語変換プログラムで生成された SERVICE LABEL ステートメント
- CMPR2
- DB2®
- DLL
- ENTRY
- IMS
- 入出力宣言とデバッグ宣言
- NOOPTIMIZE、OPTIMIZE(STD | FULL)
- NUMPROC(NOPFD | PFD | MIG)
- プログラム分割¹
- RECURSIVE
- RENT と NORENT
- SORT と MERGE
- SQL

¹ 多くのプログラム分割事例がサポートされます。サポートされない残りの事例は、ABO によって処理される CSECT のソースに、独立したセグメント、変更された GO TO ステートメント、および GO TO DEPENDING ON ステートメントが含まれる場合です。この場合は、メッセージ「B0Z1455W: unsupported feature "Program Complexity 176" found」が発行され、CSECT がスキップされます。

- SSRANGE
- TEST²
- THREAD
- TRUNC(STD | BIN | OPT)
- ユーザーが作成した SERVICE LABEL ステートメント
- XML

サポートされていない COBOL 言語機能およびコンパイラー・オプション

IBM Automatic Binary Optimizer for z/OS バージョン 2.1 は、以下の COBOL 機能を使用するプログラム・モジュールを最適化しません。

- LABEL 宣言で使用する ACCEPT FROM SYSIPT
- CLASS
- LABEL 宣言で使用する DISPLAY UPON SYSLST
- LABEL 宣言で使用する DISPLAY UPON SYSPCH
- ENTER
- INVOKE
- Java ベースのオブジェクト指向 (OO) 構文
- RERUN

不適格 CSECT の処理

IBM Automatic Binary Optimizer for z/OS (ABO) は、『[適格なコンパイラー](#)』セクションにリストされているコンパイラーによって生成された CSECT のみを最適化しますが、他の COBOL コンパイラーや言語からの CSECT を含むモジュールも許容します。

ABO は最適化の開始前に各 CSECT を調べます。以下の表のいずれかの状態が当てはまる場合、対象の CSECT は最適化には不適格であり、メッセージが発行されて、その CSECT はスキップされます。

表 4. 不適格な CSECT および発行されるメッセージ	
不適格な CSECT	発行されるメッセージ
CSECT 名が指定の CSECT フィルター 演算式に一致しない。『 CSECT 』を参照してください。	BOZ4113I
CSECT が COBOL 以外の言語から生成された。例えば、CSECT が HLASM、C/C++、または PL/I からのものである。	発行される具体的なメッセージなし ³
CSECT がいずれの適格な COBOL コンパイラーによってもコンパイルされていない。	BOZ1455W
CSECT はいずれかの適格な COBOL コンパイラーによってコンパイルされているが、『 サポート対象外の COBOL 機能 』にリストされている 1 つ以上のサポート対象外の COBOL ステートメントが含まれている。	BOZ1455W

² TEST および何らかのサブオプションでコンパイルされたプログラムは ABO によって最適化できますが、LE は ABO が生成したモジュールに対して定様式可変ダンプを生成しません。LE は代わりにメッセージ「<prog> was not compiled with the SYM suboption of the TEST. A formatted variable dump cannot be produced」を出力します。

³ オプティマイザー・オプション SCAN=Y を使用すれば、モジュールにある CSECT のタイプを判別できる。

表 4. 不適格な CSECT および発行されるメッセージ (続き)	
不適格な CSECT	発行されるメッセージ
CSECT が複雑すぎて ABO が、正しい機能コードを確実に最適化/生成できない。	BOZ1455W
予期しないデータまたはコードが CSECT に含まれている。これには、ABO が CSECT の内容を正しく理解して正しい最適化を実行するために依存するストリングがなかったり、壊れていたり、他のエラーを含んでいたりすることなどが考えられるが、これらに限定されない。	BOZ1455W
CSECT が以前に ABO によって最適化された。	BOZ1455W

最適化に適格な CSECT がモジュール内に 1 つでも見つかり、不適格な CSECT はすべて未変更のまま、最適化された CSECT とともにターゲット・モジュールにコピーされます。

第4章 インストール、およびインストールの検査

IBM Automatic Binary Optimizer for z/OS のインストール

IBM Automatic Binary Optimizer for z/OS のインストールに関する情報はすべて、製品に付属のプログラム・ディレクトリーに入っています。

最新の IBM Automatic Binary Optimizer for z/OS PTF もインストールすることを強くお勧めします。IBM Automatic Binary Optimizer for z/OS の PTF および APAR のリストについては、[フィックス・リストおよび新機能のページ](#)を参照してください。

関連資料

87 ページの『IBM Automatic Binary Optimizer for z/OS 資料』

インストール検査プログラム (IVP) を使用してインストールを検査

ABO の SMP/E インストールが完了したら、ABO インストール検査プログラム (IVP) BOZJIVP を使用して、ABO が正しくインストールされて機能していることを確認します。

BOZJIVP の概要

ABO インストール検査プログラム (IVP) BOZJIVP は ABO サンプル・ライブラリー *HLQ.SBOZJCL* にあります。*HLQ* は、ABO SMP/E インストール済み環境においてターゲット・ライブラリーに使用される接頭部です。

この IVP を、ABO を使用する予定の任意のシステム、および ABO によって生成されて最適化されたモジュールが実行されることになる任意のシステムで実行します。

注：ABO は、最小 z/OS レベルでサポートされるいずれのハードウェア・レベルでも実行できますが、ABO によって生成されて最適化されたモジュールは zEC12 システム、zBC12 システム、z13 システム、z13s システム、z14 システム、z14 ZR1 システム、および z15 システムでのみ実行できます。詳しくは、4 ページの『ターゲット・ハードウェア・レベル』を参照してください。IVP 結果を調べるときは、最小ハードウェア要件に留意してください。

BOZJIVP の使用

選択したシステムに対して IVP 処理を進めるには、組み込まれた JCL 記述に従って BOZJIVP を編集して実行依頼します。

このジョブには以下のステップが含まれています。

1. LKED - 同じサンプル・ライブラリーにあるオブジェクト BOZOBJ1 を入力として使用することでオリジナルの COBOL プログラムをリンク・エディットします。

注：この BOZOBJ1 プログラムは、OPT(STD) オプションを有効にした状態で Enterprise COBOL for z/OS V4R2 を使用してコンパイルされました。便宜のためプログラム・ソース・サンプル BOZSRC1 も同じライブラリーにあります。

2. GOBEFORE - オリジナルのプログラムを実行します。
3. VERIFY1 - z/OS のバージョンが ABO の実行に適格かどうかを検査します。
4. OPTIMIZE - ABO を使用してオリジナルのプログラムを最適化します。
5. VERIFY2 - ABO 最適化済みモジュールを実行するために IBM z サーバー・タイプが適格かどうかを検査します。
6. GOAFTER - 最適化版のオリジナル COBOL プログラムを実行します。
7. REPORT - IVP 結果を報告します。

結果

IVP が正常に実行されたときに、先行する各ステップの戻りコード 0 または 4 を受け取ります。REPORT ステップが完了すると、SYSTSRPT 出力ファイルと JESMSG LG JOBLOG にレポートが生成されます。

SYSTSRPT 出力ファイルにおけるサンプル IVP レポートを以下に例示します。

```
*** The original program start time is: 10:42:22.72
*** The original program end time is: 10:44:10.71

***
***          Optimization successful!
***
***
*** The optimized program start time is: 10:44:11.41
*** The optimized program end time is: 10:44:15.63

***
*** The elapsed time is reduced by 103.77 sec
***
***
***
***          Installation verification successful!
***
***
```

サンプル JESMSG LG JOBLOG を以下に例示します。Installation verification successful! メッセージが JOBLOG とコンソールの両方に出力されます。

10.42.22	JOB07227	HTRT01I						CPU (Total)
Elapsed								
10.42.22	JOB07227	HTRT02I	Jobname	Stepname	ProcStep	RC	I/O	hh:mm:ss.th
hh:mm:ss.th								
10.42.22	JOB07227	HTRT03I	BOZIVP	LKED		00	176	00.01
00.10								
10.44.10	JOB07227	HTRT03I	BOZIVP	GOBEFORE		00	192	01:47.49
01:48.14								
10.44.11	JOB07227	HTRT03I	BOZIVP	OPTIMIZE		00	13457	00.06
00.54								
10.44.15	JOB07227	HTRT03I	BOZIVP	GOAFTER		00	205	04.21
04.37								
10.44.15	JOB07227	****	Installation verification successful!					

10.44.15	JOB07227	HTRT03I	BOZIVP	REPORT		00	64	00.01
00.03								

VERIFY1 ステップが失敗した場合は、JOBLOG とコンソールの両方に以下のメッセージが出力されます。

```
z/OS version: xx.xx is not a supported z/OS version to run AB0.
```

VERIFY2 ステップが失敗した場合は、JOBLOG とコンソールの両方に以下のメッセージが出力されます。

```
IBM z server: type (name)
is not a supported hardware level to run AB0 optimized modules.
```

以下に例を示します。

```
IBM z server: 2817 (zEnterprise 196)
is not a supported hardware level to run AB0 optimized modules.
```

OPTIMIZE ステップが失敗した場合は、このステップ・ログ・ファイルにあるメッセージを調べて、足りないと考えられるシステムや Language Environment コンポーネントを確認してください。問題を修正してから BOZJIVP ジョブを再度実行します。

GOAFTER ステップが失敗した場合は、足りないと考えられる Language Environment PTF を調べます。3 ページの『サポート対象のオペレーティング・システム』にリストされる 1 つ以上の「Language Environment Automatic Binary Optimizer Runtime Engine」PTF がインストールされていない場合は、OC1 異常終了が発

生する可能性があります。「Language Environment Automatic Binary Optimizer Runtime Engine」PTF がインストールされているにもかかわらず、その PTF が、プログラム・ディレクトリーにリストされている最新の PTF ではない場合は、U4038 異常終了が発生し、次のいずれかのメッセージが表示されます。

```
IGZ0153S Program BOZSRC1 was compiled with a level of the compiler that requires service to be
installed on Language Environment.
IGZ0355S Program BOZSRC1 was optimized with a level of the Automatic Binary Optimizer that
requires service to be installed on Language Environment.
```

最初のメッセージは z/OS 2.2 および 2.3 の「Language Environment Automatic Binary Optimizer Runtime Engine」PTF によって発行されるものです。2 番目のメッセージは z/OS 2.4 の PTF によって発行されるものです。

異常終了が発生せずに GOAFTER ステップがゼロ以外の戻りコードで失敗した場合、その戻りコードは以下のような不足 Language Environment PTF に該当します。

表 5. 戻りコードおよび対応する欠落 LE PTF		
戻りコード	z/OS 2.2	z/OS 2.3
17	PI84563	PI84561

必要な PTF をインストールして BOZJIVP ジョブを再度実行します。

ABO で生成されたモジュールを、ABO でサポートされていないシステムで実行しようとした場合も、OC1 異常終了が発生します。サポート対象のシステムについては、[4 ページの『ターゲット・ハードウェア・レベル』](#)を参照してください。

第 5 章 モジュールの最適化

Automatic Binary Optimizer for z/OS を使用するには、最適化プロセスのための JCL を作成してください。

ABO を EXEC ステートメントで起動

JCL で EXEC ジョブ制御ステートメントを使用して、ABO を起動します。EXEC ステートメントは次のとおりです。

```
//OPT      EXEC PGM=B0ZOPT
```

必須 DD ステートメント

最適化プロセスでは、最適化プロセスにおける特定の使用目的のデータ・セットを指定する必要があります。それらのデータ・セットは、必須 DD 名を付けて DD ステートメントに定義できます。ABO によって使用される DD 名とそれらの特性は、[15 ページの表 6](#) に示されています。

オブティマイザー・ディレクティブ BOPT または IEFOPZ の指定

BOPT または IEFOPZ を使用して、ABO に指示を出します。1 つ以上の BOPT または IEFOPZ ディレクティブを SYSIN DD に組み込むことができます。詳しくは、[16 ページの『BOPT』](#)および [18 ページの『IEFOPZ』](#)を参照してください。

必須 DD ステートメント

Automatic Binary Optimizer for z/OS によって使用される DD 名を次の表に示します。

表 6. バイナリー最適化に使用される DD 名			
DD 名	タイプ	必須	説明
STEPLIB	入力	はい	ABO が入っているデータ・セットの名前と、従属 Language Environment® ランタイム・データ・セットを指定します。
SYSIN	入力	はい	オブティマイザー・ディレクティブ (BOPT および IEFOPZ) とオブティマイザー・オプションを含むファイルのロケーションを指定します。便宜上、DD * を使用してインストリーム・ファイルを JCL に指定できます。
OPTLOG	出力	はい	この DD に最適化の要約情報 (最適化の対象や最適化されたバイナリーのロケーションなど) が書き込まれることを指定します。SCAN 出力もここに書き込まれます。

表 6. バイナリー最適化に使用される DD 名 (続き)

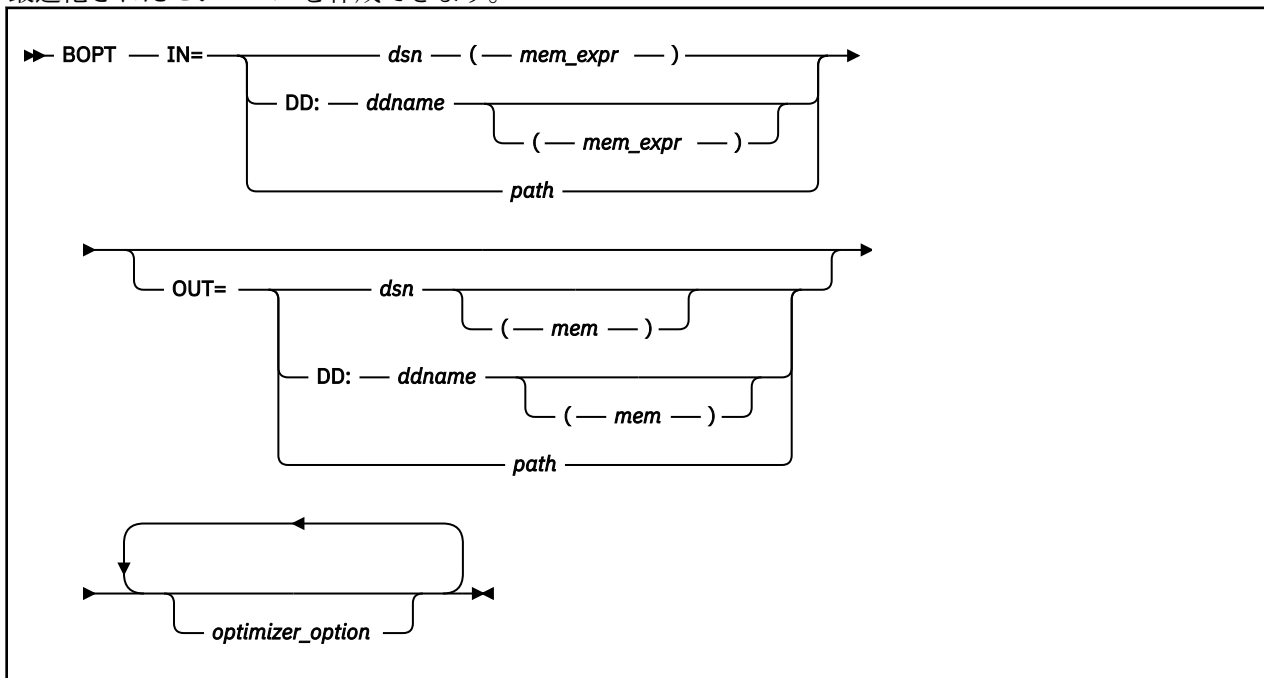
DD 名	タイプ	必須	説明
SYSPRINT	出力	いいえ (LIST オプションが指定されている場合) はい (LIST オプションが指定されていない場合)	リスト変換のデフォルト・ロケーションを指定します。49 ページの『SYSPRINT DD および LIST オプション』も参照してください。
OPTERR	出力	いいえ	例外状況の際に、この DD に最適化の診断情報が書き込まれることを指定します。
CEEDUMP	出力	いいえ	例外状況の際に、この DD に最適化のダンプ情報が書き込まれることを指定します。
CEEOPTS	入力	いいえ (英語メッセージが必要である場合) はい (日本語メッセージが必要である場合)	これは、メッセージに使用される言語を指定します。33 ページの『ABO メッセージに使用される言語を指定』も参照してください。

オブティマイザー・ディレクティブ

BOPT または IEFOPZ を使用して、ABO に指示を出します。

BOPT

BOPT オプティマイザー・ディレクティブを使用すると、明示的な入力指定子および出力指定子に基づいて最適化されたモジュールを作成できます。



IN

最適化する 1 つの入力モジュールを指定するか、ワイルドカードが *mem_expr* 指定子に与えられている場合は複数の入力モジュールを指定します。

OUT

これは、1 つの出力モジュールを指定するか、または *mem* 指定子が省略されている場合は 1 つ以上の出力モジュールに対して 1 つの PDS(E) を指定します。

DD:ddname

DD 名を指定します。

dsn

データ・セット名で、高位修飾子が含まれている必要があります。

mem

データ・セット・メンバー名です。

mem_expr

正規表現を含む可能性があるデータ・セット・メンバー名です。名前ストリングが正規表現に一致するメンバーのみが処理されます。突き合わせで大/小文字は区別されません。

以下の記号が正規表現で受け入れられます。

これは任意のストリングに一致します。

?

これは任意の文字に一致します。

|

これは複数の正規表現の区切り文字として使用できます。複数の正規表現のうち、いずれかの正規表現がストリングに一致すれば、正規表現が一致したとみなされます。

!

これは、後続の式全体を否定します。以下に例を示します。

- MEMA という名前の単一のメンバーをスキップする場合:

```
IN=DD:SYSBIN(!MEMA)
```

- MEMB で始まる名前を持つすべてのメンバーをスキップする場合:

```
IN=DD:SYSBIN(!MEMB*)
```

- SUB1 と SUB2 という名前のメンバーをスキップする場合:

```
IN=DD:SYSBIN(!SUB1|SUB2)
```

path

スラッシュ (/) で始まるフル HFS パス (例えば /home/user1/a.out.opt) です。

optimizer_option

オブティマイザー・オプションです。指定できるオブティマイザー・オプションのリストについては、[19 ページの『オブティマイザー・オプション』](#)を参照してください。

注:

- SCAN オブティマイザー・オプションが Y に設定されている場合、BOPT における OUT オプションの指定は任意です。
- mem_expr* が IN オプションに指定されている場合、*mem_expr* に一致するメンバーすべてが、最適化の対象として選択されます。*mem_expr* が指定されているときに OUT オプション上に *mem* 指定子を組み込まないでください。
- OUT オプション上に *mem* 指定子がない場合、OUT のメンバー名は、IN オプション上の *mem* 指定子または *mem_expr* 指定子に一致するメンバー名であると判断されます。
- IN 指定子、OUT 指定子、およびオブティマイザー・オプションの順序は任意の順序でかまいません。

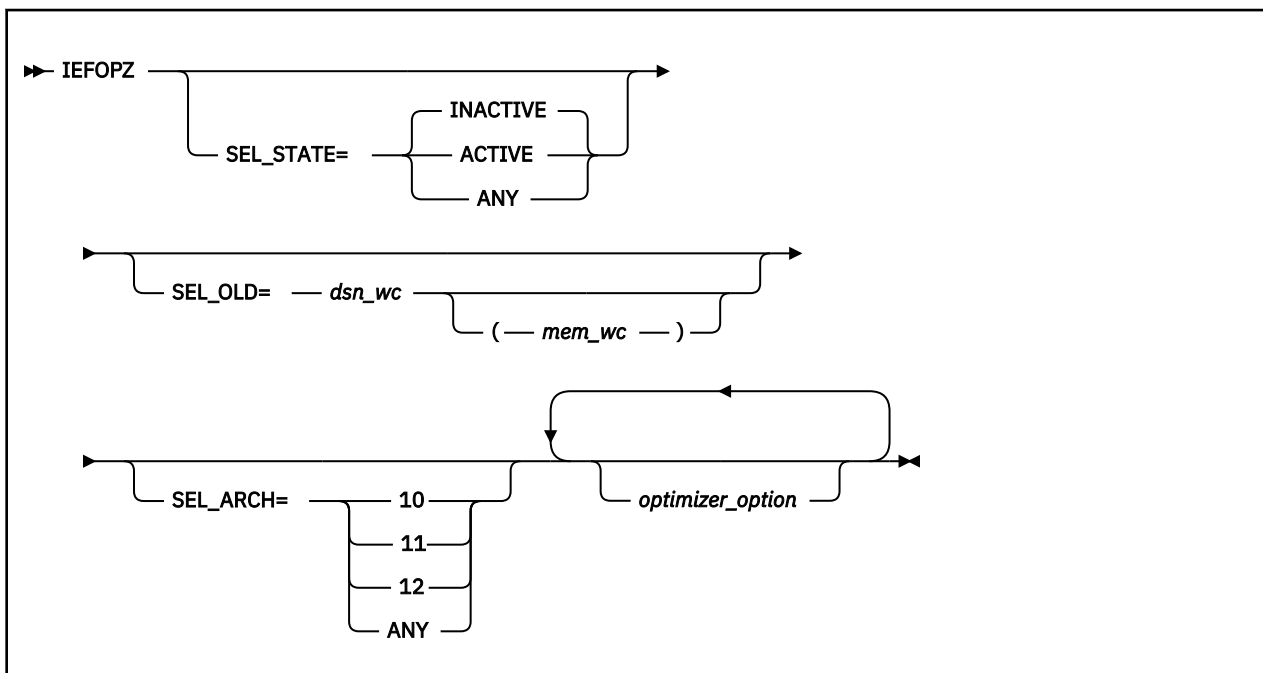
BOPT ディレクティブの例については、28 ページの『JCL の例』を参照してください。BOPT オプティマイザー・ディレクティブの使用におけるサンプル・シナリオについては、54 ページの『シナリオ 1: 静的デプロイメントによる最適化プロセス』および 56 ページの『シナリオ 3: ハイブリッド・アプローチによる最適化プロセス』を参照してください。

IEFOPZ

IEFOPZ オプティマイザー・ディレクティブを使用すると、IEFOPZ 構成に基づいて最適化されたモジュールを作成できます。

注：IEFOPZ は z/OS V2.2 以降でのみサポートされています。z/OS 2.2 では、APAR/PTF OA47689/UA90982 を適用する必要があります。

IEFOPZ 構成については、55 ページの『シナリオ 2: 動的デプロイメントによる最適化プロセス』のステップ 2 を参照してください。



SEL_STATE

指定された状態に一致するマッピングを最適化するように、オプティマイザーに指示します。

ANY

ACTIVE または INACTIVE とマークされたマッピングを最適化するように、オプティマイザーに指示します。

ACTIVE

ACTIVE とマークされたマッピングのみ最適化するように、オプティマイザーに指示します。

INACTIVE

INACTIVE とマークされたマッピングのみ最適化するように、オプティマイザーに指示します。

SEL_OLD

指定されたセクターに一致する OLD データ・セットを持つマッピングに最適化を制限します。

dsn_wc

アスタリスク (*) 記号を使用してワイルドカードを組み込むことができるデータ・セット名。例:
IN.LOAD

mem_wc

アスタリスク (*) 記号を使用してワイルドカードを組み込むことができるデータ・セット・メンバー名。
例: M*

SEL_ARCH

指定されたアーキテクチャーでマークされたマッピングを最適化するように、オプティマイザーに指示します。

10

ARCH(10) とマークされたマッピングのみ最適化するよう、オブティマイザーに指示します。

11

ARCH(11) とマークされたマッピングのみ最適化するよう、オブティマイザーに指示します。

12

ARCH(12) とマークされたマッピングのみ最適化するよう、オブティマイザーに指示します。

ANY

これは、ARCH(10)、ARCH(11)、または ARCH(12) とマークされたマッピングを最適化するようにオブティマイザーに指示します。

optimizer_option

オブティマイザー・オプションです。指定できるオブティマイザー・オプションのリストについては、[19 ページの『オブティマイザー・オプション』](#)を参照してください。

注：

- マッピングとは、IEFOPZ 構成における OLD モジュールから NEW モジュールへの関連付けを指します。
- デフォルトで、IEFOPZ 構成の OLD データ・セット内の INACTIVE モジュールすべてが、NEW データ・セットによって決定され、また IncludeMembers 構成指定子および ExcludeMembers 構成指定子によって決定されたアーキテクチャー・レベルで最適化されます。SEL_OLD= および SEL_ARCH= は、対象のモジュールのサブセットに最適化およびスキャンを制限するために使用できるセクターです。

SEL_STATE= は、ACTIVE モジュールまたは ANY 状態のモジュールに最適化やスキャンを変更するために使用できるセクターです。ACTIVE モジュールのスキャンではリスクはありませんが、ACTIVE モジュールの最適化では、その ACTIVE モジュールを使用するプログラムに関して問題が発生する可能性があります。セクター SEL_STATE=ACTIVE または SEL_STATE=ANY を使用するとき、および (スキャンとは対照的に) 最適化を行うときは、注意が必要です。

- IN 指定子、OUT 指定子、およびオブティマイザー・オプションの順序は任意の順序でかまいません。

IEFOPZ ディレクティブの例については、[28 ページの『JCL の例』](#)を参照してください。IEFOPZ オブティマイザー・ディレクティブの使用におけるサンプル・シナリオについては、[55 ページの『シナリオ 2: 動的デプロイメントによる最適化プロセス』](#)を参照してください。

オブティマイザー・オプション

Automatic Binary Optimizer for z/OS オプションは、BOPT オブティマイザー・ディレクティブと IEFOPZ オブティマイザー・ディレクティブの両方に適用できるオブティマイザー・オプションです。

オブティマイザー・オプションは、SYSIN ファイルの先頭から 1 つ以上の行に、あるいは BOPT オブティマイザー・ディレクティブまたは IEFOPZ オブティマイザー・ディレクティブに置くことができます。

グローバル・オプションは、BOPT オブティマイザー・ディレクティブまたは IEFOPZ オブティマイザー・ディレクティブが含まれない行に指定されるオブティマイザー・オプションです。グローバル・オプションの値はそのオプションのグローバル設定とみなされます。

オブティマイザー・ディレクティブが含まれている特定の行にオブティマイザー・オプションが指定されている場合、値はその最適化オブティマイザー・ディレクティブにのみ適用され、後続のステートメントに関しては値はグローバル設定に戻されます。

[19 ページの表 7](#) に、BOPT および IEFOPZ の両方に適用されるオブティマイザー・オプションの要約があります。

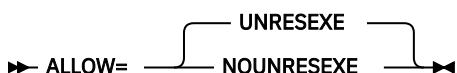
表 7. オブティマイザー・オプション		
オプション	デフォルト	説明
20 ページの『ALLOW』	ALLOW=UNRESEXE	この場合は、ABO が受け入れるプログラム・モジュールのタイプが制御されます。

表 7. オプティマイザー・オプション (続き)		
オプション	デフォルト	説明
20 ページの『ARCH』	ARCH=10	ターゲット・ハードウェア・レベルを指定します。
21 ページの『CSECT』	CSECT オプションが指定されない場合、ABO はすべての適格な CSECT を処理します。	ユーザーは 1 つ以上の CSECT に処理を制限できるようになります。
23 ページの『LIST』	LIST オプションが指定されない場合、リスト変換は、SYSPRINT DD に従ったロケーションに置かれます。	生成されるリスト変換のロケーションを指定します。
24 ページの『LOG』	LOG オプションが指定されない場合、ABO はメンバー・レベルのログ・ファイルを生成しません。	これは、追加で生成されるメンバー・レベル・ログ・ファイルのロケーションを指定します。
25 ページの『REPLACE』	REPLACE=Y	出力モジュールが書き込まれるかどうかを制御します。
26 ページの『SCAN』	SCAN=N	プログラム・モジュールを最適化するかスキャンするかを制御します。
注: ARCH オプションはグローバル・レベルで、BOPT ディレクティブに指定できます。IEFOPZ ディレクティブには指定できません。IEFOPZ では、SEL_ARCH オプションを使用して、SEL_ARCH 値に一致する IEFOPZ 構成内の部分を最適化の対象として選択します。		

ALLOW

目的

ALLOW オプションは、Automatic Binary Optimizer for z/OS が受け入れるプログラム・モジュールのタイプを制御します。



デフォルト

ALLOW=UNRESEXE

USAGE

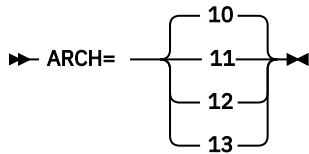
ALLOW=UNRESEXE が指定されている場合、ABO は、完全にバインドされたモジュールでも部分的にバインドされたモジュールでも受け入れます。部分的にバインドされたプログラム・モジュールに関しては、オプティマイザーによって受け入れられる唯一のタイプのプログラム・モジュールは、CALL=NO または NCAL バインダー・オプションでリンクされたプログラム・モジュールです。入力モジュールが完全にバインドされている場合は、最適化された出力モジュールも完全にバインドされます。入力モジュールが部分的にバインドされている場合は、最適化されたモジュールも部分的にバインドされます。

ALLOW=NOUNRESEXE が指定されている場合、ABO は、完全にバインドされたプログラム・モジュール (プログラム・オブジェクトまたはロード・モジュール) のみを受け入れ、完全にバインドされたプログラム・モジュールを常に生成します。ALLOW=NOUNRESEXE が指定されているときに、部分的にバインドされたモジュールが処理されると、メッセージ BOZ1494S が発行されます。

ARCH

目的

ARCH オプションで、ターゲット・ハードウェア・レベルを指定します。



デフォルト

ARCH=10

USAGE

ARCH オプションは、ABO によって生成された最適化済みモジュールがターゲットとするハードウェア・レベルを指定する場合に使用します。

下位の ARCH 設定を使用して生成された最適化済みモジュールは上位の ARCH システムで実行されます。一方、上位の ARCH 設定からの最適化済みモジュールは下位の ARCH システムでは動作しません。

ARCH 設定	IBM Z メインフレーム・レベルでのみ実行可能
ARCH=10	zEC12, zBC12, z13, z13s, z14, z14 ZR1, z15
ARCH=11	z13, z13s, z14, z14 ZR1, z15
ARCH=12	z14, z14 ZR1, z15
ARCH=13	z15

無効な組み合わせが試行されると、プログラムは終了し、次のランタイム LE メッセージが表示される可能性があります。

```
CEE3201S The system detected an operation exception (System Completion Code=0C1)
```

上記の ARCH レベルについて詳しくは、[4 ページの『ターゲット・ハードウェア・レベル』](#)を参照してください。

CSECT

目的

CSECT オプションを使用すれば、ゼロ個以上の CSECT に処理を制限できます。

➡ CSECT= *expr* ➡

デフォルト

CSECT オプションが指定されない場合、ABO はデフォルトですべての適格な CSECT を処理します。

パラメーター

expr

処理対象の CSECT の正規表現。名前ストリングが正規表現に一致する CSECT のみが処理されます。突き合わせで大/小文字は区別されません。

USAGE

以下の記号が正規表現で受け入れられます。

*

これは任意のストリングに一致します。

?

これは任意の文字に一致します。

これは複数の正規表現の区切り文字として使用できます。複数の正規表現のうち、いずれかの正規表現がストリングに一致すれば、正規表現が一致したとみなされます。

これは、後続の式全体を否定します。以下に例を示します。

- PROGA という名前の CSECT をスキップする場合:

```
CSECT=!PROGA
```

- PROGB で始まる名前を持つすべての CSECT をスキップする場合:

```
CSECT=!PROGB*
```

- SUB1 と SUB2 という名前の CSECT をスキップする場合:

```
CSECT=!SUB1|SUB2
```

注:

- 正規表現にスペースや大括弧は使用できません。
- 正規表現はストリング全体に一致しなければなりません。部分的な一致は一致とみなされません。つまり、正規表現 M*2 はストリング MA2 と一致しますが、ストリング MA2A とは一致しません。
- 正規表現で使える特殊文字については各 EBCDIC コード・ページで文字エンコード方式が異なるため、このオプションは以下のコード・ページでのみ使用します。
 - IBM-1047
 - IBM-37/1140
 - IBM-285/1146
 - IBM-924

例 1

次の例では、ワイルドカード・フィルタに一致しない CSECT は処理されません。

JCL コマンド

```
BOPT IN=DD:SYSBIN(*) OUT=DD:SYSBOUT CSECT=SUB*1*
```

出力 (OPTLOG)

```
...
10:46:04 Processing CSECT filter expression 'SUB*1*' on member CALLLITT
10:46:04 CSECT CALLLIT was excluded by filter - skip
10:46:04 Processing CSECT SUB01L00, in member CALLLITT
10:46:04 Optimizing CSECT SUB01L00 for zEC12
10:46:04 Succeeded in optimizing SUB01L00
10:46:04 Generating listing transform into DD:SYSPRINT
10:46:04 CSECT SUB02L00 was excluded by filter - skip
10:46:04 CSECT SUB03L00 was excluded by filter - skip
10:46:04 CSECT SUB04L00 was excluded by filter - skip
10:46:04 CSECT SUB05L00 was excluded by filter - skip
10:46:04 CSECT SUB06L00 was excluded by filter - skip
10:46:04 CSECT SUB07L00 was excluded by filter - skip
10:46:04 CSECT SUB08L00 was excluded by filter - skip
10:46:04 CSECT SUB09L00 was excluded by filter - skip
10:46:04 Processing CSECT SUB10L00, in member CALLLITT
10:46:04 Optimizing CSECT SUB10L00 for zEC12
10:46:04 Succeeded in optimizing SUB10L00
10:46:04 Generating listing transform into DD:SYSPRINT
10:46:04 Finished processing, processed 2 of 11 CSECTs in member CALLLITT
```

例 2

次の例は、突き合わせに複数の正規表現を指定する方法を示したものです。

JCL コマンド

```
BOPT IN=DD:SYSBIN(*) OUT=DD:SYSBOUT CSECT=SUB01L00|SUB02L00
```

OUTPUT

```
...
10:49:13 Processing CSECT filter expression 'SUB01L00|SUB02L00' on member CALLLITT
10:49:13 CSECT CALLLIT was excluded by filter - skip
10:49:13 Processing CSECT SUB01L00, in member CALLLITT
10:49:13 Optimizing CSECT SUB01L00 for zEC12
10:49:13 Succeeded in optimizing SUB01L00
10:49:13 Generating listing transform into DD:SYSPRINT
10:49:13 Processing CSECT SUB02L00, in member CALLLITT
10:49:13 Optimizing CSECT SUB02L00 for zEC12
10:49:13 Succeeded in optimizing SUB02L00
10:49:13 Generating listing transform into DD:SYSPRINT
10:49:13 CSECT SUB03L00 was excluded by filter - skip
10:49:13 CSECT SUB04L00 was excluded by filter - skip
10:49:13 CSECT SUB05L00 was excluded by filter - skip
10:49:13 CSECT SUB06L00 was excluded by filter - skip
10:49:13 CSECT SUB07L00 was excluded by filter - skip
10:49:13 CSECT SUB08L00 was excluded by filter - skip
10:49:13 CSECT SUB09L00 was excluded by filter - skip
10:49:13 CSECT SUB10L00 was excluded by filter - skip
10:49:13 Finished processing, processed 2 of 11 CSECTs in member CALLLITT
```

例 3

次の例では、データ・セット HLQ.IN.LOAD 内の MEM1 に、A1 および B1 という 2 つの CSECT があります。ABO の処理を A1 のみに制限するには、次のように CSECT=A* フィルターを使用します。

JCL コマンド

```
BOPT IN=HLQ.IN.LOAD(MEM1) OUT=HLQ.OUT.LOAD(MEM1) CSECT=A*
```

この CSECT=A* フィルターが機能すると、OPTLOG は次のようになります。

OUTPUT

```
...
17:46:04 Processing CSECT filter expression 'A*' on member MEM1
17:46:04 Processing CSECT A1, in member MEM1
17:46:04 Optimizing CSECT A1 for zEC12
17:46:04 Succeeded in optimizing A1
17:46:04 Generating listing transform into DD:SYSPRINT
17:46:04 CSECT B1 was excluded by filter - skip
17:46:04 Finished processing, processed 1 of 2 CSECTs in member MEM1
...
```

また、CSECT=A* は、後続のすべての BOPT ディレクティブおよび IEFOPZ ディレクティブに適用される (これらのディレクティブが特定のディレクティブによってオーバーライドされる 場合を除く) ようにグローバル・オプションとして指定できます。

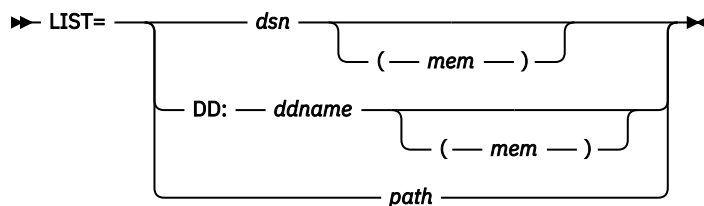
```
//SYSIN DD *
      CSECT=A*
      BOPT IN=HLQ.IN.LOAD(MEM1) OUT=HLQ.OUT.LOAD(MEM1A)
      BOPT IN=HLQ.IN.LOAD(MEM1) OUT=HLQ.OUT.LOAD(MEM1B) CSECT=B*
```

処理が終わると、メンバー MEM1A には、最適化された CSECT A1 とオリジナルの CSECT B1 が含まれ、メンバー MEM1B には、最適化された CSECT B1 とオリジナルの CSECT A1 が含まれます。

LIST

目的

LIST オプションで、生成されるリスト変換のロケーションを指定します。



デフォルト

LIST オプションが指定されないと、リスト変換はデフォルトで、SYSPRINT DD に従ったロケーションに置かれます。

パラメーター

dsn

データ・セット名で、高位修飾子が含まれている必要があります。

mem

データ・セット・メンバー名です。

DD:ddname

DD 名を指定します。

path

スラッシュ (/) で始まるフル HFS パス (例えば /home/user1/a.list) です。

USAGE

LIST オプションのターゲットは、以下の項目のいずれかです。

- 順次データ・セット、または PDSE (PDS ではない) のメンバー。複数の CSECT 最適化の出力は、この単一の順次データ・セットに追加されます。
- PDS または PDSE。CSECT が最適化されると、その CSECT に特定のリスト変換が PDS または PDSE のメンバーに置かれます。メンバー名は CSECT 名に基づいています (大文字変換され、8 文字に切り捨てられます)。メンバーに内容があれば、前の内容がオプティマイザーによって以前の起動で作成されたものであっても、その内容は上書きされます。
- HFS パス。複数の CSECT 最適化の出力は、この HFS ファイルに追加されます。

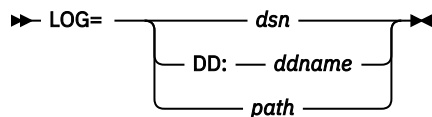
関連情報

SYSPRINT DD

LOG

目的

LOG オプションで、追加で生成されるメンバー・レベルのログ・ファイルのロケーションを指定します。



デフォルト

デフォルトでは、LOG オプションが指定されない場合、ABO はメンバー・レベルのログ・ファイルを生成しません。LOG オプションに関係なく、ABO の起動全体からの出力は、OPTLOG DD に従った位置に常に生成されることに注意してください。

パラメーター

dsn

データ・セット名で、高位修飾子が含まれている必要があります。

DD:ddname

DD 名を指定します。

path

スラッシュ (/) で始まるフル HFS パスです。

USAGE

LOG オプションのターゲットは、以下の項目のいずれかです。

- PDS または PDSE。メンバーが最適化されると、そのメンバーに特定のログ出力が PDS または PDSE のメンバーに置かれます。メンバー名は最適化されたメンバー名に基づいています (大文字変換され、8 文字に切り捨てられます)。メンバーに内容があれば、前の内容が ABO によって以前の起動で作成されたものであっても、その内容は上書きされます。
- ディレクトリーを指す HFS パス。メンバーが最適化されると、そのメンバーに特定のログ出力が、指定の HFS ディレクトリーにファイルで配置されます。ファイル名は最適化済みメンバーの名前に基づきます。ファイル名には `.log` が付加されます。メンバーに内容があれば、前の内容が ABO によって以前の起動で作成されたものであっても、その内容は上書きされます。

注:

1. LOG オプションのターゲットは、メンバーが指定されていない PDS または PDSE であるか、あるいはディレクトリーを指す HFS パスでなければなりません。LOG オプションのターゲットが、メンバーが指定された PDS または PDSE として、順次データ・セットとして、あるいは HFS ファイルとして指定された場合は、エラー・メッセージが発行されます。
2. LOG オプションのターゲット・データ・セットは、[65 ページの表 12](#) の OPTLOG の推奨割り振りパラメーターに従ってなければなりません。

例 1

```
BOPT IN=DD:SYSBIN(*) OUT=DD:SYSBOUT LOG=HLQ.LOG.OUT
```

例 2

```
BOPT IN=DD:SYSBIN(*) OUT=DD:SYSBOUT LOG=DD:LOG
```

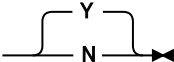
例 3

```
BOPT IN=DD:SYSBIN(*) OUT=DD:SYSBOUT LOG=/home/user1/logdir
```

詳しくは、[41 ページの『ログ・ファイル』](#)を参照してください。

REPLACE**目的**

出力モジュールが書き込まれるかどうかを制御します。

➡ REPLACE=  ➡

デフォルト

REPLACE=Y

USAGE

REPLACE=Y および SCAN=N の場合は、出力モジュールが存在するかどうかに関係なく、最適化されたモジュールが出力モジュールに書き込まれます。

- 出力モジュールが存在しない場合は、作成されて書き込まれます。
- 出力モジュールが存在する場合、その内容は上書きされます。

ただし、最適化の対象として適格な COBOL CSECT がなければ、出力モジュールには何も書き込まれません。

REPLACE=N が指定されると、既に出力モジュールが存在する場合、入力モジュールの最適化またはスキャンはバイパスされます。つまり、出力モジュールには何も書き込まれません。

REPLACE=N を使用すると、既に最適化されているモジュールの最適化をバイパスできます。例えば、バイナリー最適化の後で新規メンバーをオリジナル・データ・セットに追加し、新規メンバーのみ最適化する場合、次のようにメンバー・ワイルドカードを REPLACE=N と一緒に使用することができます。

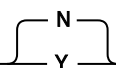
```
BOPT IN=HLQ.IN.LOAD(*) OUT=HLQ.OUT.LOAD REPLACE=N
```

時間超過または他の異常条件が原因で、オブティマイザーが早まって終了する場合があります。この問題を解決するには、最適化されたモジュールのインクリメンタル・ビルドをジョブの順序で実行して、前のジョブで実行した最適化を繰り返すことに時間を浪費しないようにします。または、このような場合には REPLACE=N を使用できます。

SCAN

目的

SCAN オプションで、プログラム・モジュールを最適化するかスキャンするかを ABO に指示します。

➡ SCAN=  ➡

デフォルト

SCAN=N

USAGE

SCAN=N が有効になっていると、ABO は入力プログラム・モジュールに対する最適化を実行します。

SCAN=Y が有効になっていると、ABO は最適化を実行する代わりに入力プログラム・モジュールをスキャンします。出力モジュールは生成されません。

REPLACE=N オプションは、以下のように、スキャンが実行されるかどうかに影響します。

- REPLACE=N が BOPT ディレクティブに指定されていて、OUT オプションの出力モジュールが既に存在する場合、IN オプションのモジュールのスキャンはバイパスされます。
- REPLACE=N が IEFOPZ ディレクティブに指定されていると、NEW データ・セットのメンバーが既に存在する場合、OLD データ・セットのメンバーのスキャンはバイパスされます。

OUT オプションを BOPT ディレクティブから除外すると、REPLACE オプションの値に関係なく、スキャンは常に実行されます。

スキャン・モードでは、オブティマイザーは入力プログラム・モジュールと出力プログラム・モジュールを検査し、モジュール内の CSECT をリストします。スキャン出力は OPTLOG DD に書き込まれます。

SCAN=Y を使用すると、SYSIN セットアップをテストする (存在するモジュールとそれらの内容を調べる) ことができます。使用されているオリジナル・コンパイラーが原因で、プログラムが最適化の対象として不適格である場合、スキャン出力でもそのことが示されます。

詳しくは、[42 ページの『スキャンに関するログ・ファイル』](#)を参照してください。

コメント

コメントを指定するには、(#) 文字から始めます。

(#) 文字を使用するときは、以下の規則が適用されます。

- SYSIN ファイルの行にある最初の非ブランク文字が (#) 文字である場合、その行の残り部分は無視されます。
- ABO が SYSIN ファイルの入力行で (#) 文字を認識し、その (#) 文字の前にあるのがブランクである場合、その行の残り部分は無視されます。

例 1

```
#BOPT IN=DD:SYSBIN OUT=SYSBOUT
#BOPT IN=SYSBIN OUT=SYSBOUT
```

例 1 では、両方の BOPT ディレクティブがコメント化されています。最初の行では、# 文字が 1 桁目にあり、その行の残り部分は無視されます。2 番目の行では、最初の実空白文字が # 文字であり、その行の残り部分は無視されます。

例 2

```
# Optimizing all members of library
# Note: we don't compile a member optimized earlier and found in the OUT dataset
BOPT IN=SYSBIN(*) OUT=SYSBOUT REPLACE=N
```

例 2 は、SYSIN ファイルに情報コメントを 2 行全体にわたって追加する例を示しています。

例 3

```
BOPT IN=SYSBIN(*) OUT=SYSBOUT #
```

例 3 では、行末の # 文字は無視されます。

例 4

```
BOPT IN=SYSBIN(*) OUT=SYSBOUT #optimizing library files
```

例 4 は、ディレクティブの後にコメントを追加する方法を示しています。

行の継続

SYSIN ファイルにおける行の継続は (+) 継続文字または (-) 継続文字で指定されます。この継続文字は、次の行が前の行の一部であるかのように読み取られることを示します。行の継続は必須ではありませんが、行の長さが制限されている可能性がある他のツールで編集を簡単にしたり使い勝手を良くしたりするために長い行の分割に使用できます。

(+) 継続文字または (-) 継続文字は、行にある最後の非空白および非コメントの文字でなければなりません。継続文字を使用するときは、以下の規則が適用されます。

- (-) 文字は、完全なオプション、ディレクティブ、または指定子の後に行を継続する場合にのみ使用できます。行の継続は、次の行の 1 桁目から始まります。(-) 文字の前には 1 つ以上の空白を挿入する必要があります。
- (+) 文字は、完全なオプション、ディレクティブ、または指定子の後に、あるいはオプション、ディレクティブ、または指定子の途中で、行を継続する場合に使用できます。行の継続は、次の行において最初の非空白文字から始まります。完全なオプション、ディレクティブ、または指定子が続ける場合は、(+) 文字の前に 1 つ以上の空白を挿入する必要があります。
- 行継続文字の前にある空白は連結後の行に含まれます。
- コメント文字 (#) は行継続文字に優先します。行継続文字がコメントの一部である場合、その行継続文字は無視され、コメントは続行されません。

予期しない位置に継続文字がある場合は、エラー・メッセージが発行されることがあります。

以下の例は、BOPT オプティマイザー・ディレクティブを使用する JCL を示しています。この例は完全なものではありません。ユーザーが SYSIN ファイルに何を指定しなければならないかを示すことを目的としています。基本 JCL 構成については、65 ページの『付録 A JCL サンプル』を参照してください。

例 1

```
//SYSIN DD *
ARCH=11 -
ALLOW=UNRESEXE
BOPT IN=HLQ.LOAD.APPXYZ1.ORIG(*) -
OUT=HLQ.LOAD.APPXYZ1.ABO
```

この例は、ABO によって以下のように解釈されます。

```
//SYSIN DD *
ARCH=11 ALLOW=UNRESEXE
BOPT IN=HLQ.LOAD.APPXYZ1.ORIG(*) OUT=HLQ.LOAD.APPXYZ1.ABO
```

継続文字の前にあるブランクは、いかなる場合でも連結後のストリングに含まれます。

例 2

```
//SYSIN DD *  
                                AR+  
                                CH=12 +  
                                ALLOW=UNRESEXE  
  
BOPT IN=HLQ.LOAD.APPXYZ1.OR+  
IG(*) -  
OUT=HLQ.LOAD.APPXYZ1.ABO
```

この例では、1 桁目だけではなく次の行において (最初の非ブランク位置で) 任意の場所に (+) 文字によって継続を可能とする方法、および不完全なオプション、ディレクティブ、または指定子を継続するために (+) 文字を使用できる方法が示されています。この例は、ABO によって以下のように解釈されます。

```
//SYSIN DD *  
ARCH=12 ALLOW=UNRESEXE  
BOPT IN=HLQ.LOAD.APPXYZ1.ORIG(*) OUT=HLQ.LOAD.APPXYZ1.ABO
```

例 3

```
//SYSOPTF DD *  
    # some comment +  
                                ARCH=12    +  #other comment -  
                                ALLOW=UNRESEXE  
BOPT IN=HLQ.LOAD.APPXYZ1.ORIG(*) OUT=HLQ.LOAD.APPXYZ1.ABO
```

(1)
(2)
(3)
(4)

この例は、ABO によって以下のように処理されます。

```
//SYSOPTF DD *  
ARCH=12 ALLOW=UNRESEXE  
BOPT IN=HLQ.LOAD.APPXYZ1.ORIG(*) OUT=HLQ.LOAD.APPXYZ1.ABO
```

行 1 の (+) 文字および行 2 の (-) 文字はコメントの一部であるため無視されます。行 2 の (+) 文字は、行にある最後の非コメントおよび非ブランクの文字であるため行継続文字として扱われます。

JCL の例

以下の例に示されているジョブ制御言語 (JCL) ステートメントを使用すると、コンパイル済み COBOL モジュールを Automatic Binary Optimizer for z/OS で処理できます。

BOPT で最適化を指定

以下の例は、BOPT オプティマイザー・ディレクティブを使用する JCL を示しています。

このセクションの例は完全なものではありません。ユーザーが SYSIN ファイルに何を指定しなければならないかを示すことを目的にしています。基本 JCL 構成については、[65 ページの『付録 A JCL サンプル』](#)を参照してください。

例 1. データ・セット名を使用して入力モジュールと出力モジュールを指定

この例では、入力モジュールと出力モジュールは BOPT 行で判別されます。入力データ・セット名と出力データ・セット名は、DD 名に指定されるのではなく、明確に与えられます。

```
...  
//SYSIN DD *  
  ARCH=12  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=HLQ.OUT.LOAD(MEM1)
```

例 2. DD 名を使用して入力モジュールと出力モジュールを指定

次の例では、DD 名を使用して入力モジュールと出力モジュールが指定されています。

```
...  
//SYSBIN DD DSN=HLQ.IN.LOAD,DISP=SHR
```

```
//SYSBOUT DD DSN=HLQ. OUT. LOAD, DISP=SHR
//SYSIN DD *
ARCH=12
BOPT IN=DD:SYSBIN(MEM1) OUT=DD:SYSBOUT(MEM1)
```

例 3. HFS パスを使用して入力モジュールと出力モジュールを指定

次の例では、HFS パスを使用して入力モジュールと出力モジュールが指定されています。スラッシュ (/) で始まる完全修飾 HFS パスを使用する必要があります。

```
...
//SYSIN DD *
ARCH=12
BOPT IN=/home/user1/a.out OUT=/home/user1/a.out.opt
```

例 4. 入力モジュールを指定し、出力メンバー指定子を省略

次の例では、入力モジュールが指定されていて、出力メンバー指定子が省略されています。最適化されたモジュールの出力 PDS(E) 内のメンバー名は、入力 PDS(E) 内で指定されたメンバーと同じ名前になります。

```
...
//SYSBIN DD DSN=HLQ.IN. LOAD, DISP=SHR
//SYSBOUT DD DSN=HLQ. OUT. LOAD, DISP=SHR
//SYSIN DD *
ARCH=10
BOPT IN=DD:SYSBIN(MEM1) OUT=DD:SYSBOUT
```

例 5. 正規表現を使用して複数の入力モジュールを指定

次の例では、正規表現を使用して複数の入力モジュールが指定されています。最適化されたモジュールの出力 PDS(E) 内のメンバー名は、入力 PDS(E) 内で対応するメンバーと同じ名前になります。

MEM で始まる名前を持つモジュールを組み込むには、以下のようになります。

```
...
//SYSBIN DD DSN=HLQ.IN. LOAD, DISP=SHR
//SYSBOUT DD DSN=HLQ. OUT. LOAD, DISP=SHR
//SYSIN DD *
ARCH=10
BOPT IN=DD:SYSBIN(MEM*) OUT=DD:SYSBOUT
```

MEMA という名前の単一のモジュールを除外する場合:

```
...
//SYSBIN DD DSN=HLQ.IN. LOAD, DISP=SHR
//SYSBOUT DD DSN=HLQ. OUT. LOAD, DISP=SHR
//SYSIN DD *
ARCH=10
BOPT IN=DD:SYSBIN(!MEMA) OUT=DD:SYSBOUT
```

MEMB で始まる名前を持つすべてのメンバーを除外する場合:

```
...
//SYSBIN DD DSN=HLQ.IN. LOAD, DISP=SHR
//SYSBOUT DD DSN=HLQ. OUT. LOAD, DISP=SHR
//SYSIN DD *
ARCH=10
BOPT IN=DD:SYSBIN(!MEMB*) OUT=DD:SYSBOUT
```

SUB1 と SUB2 という名前のメンバーを除外する場合:

```
...
//SYSBIN DD DSN=HLQ.IN. LOAD, DISP=SHR
//SYSBOUT DD DSN=HLQ. OUT. LOAD, DISP=SHR
//SYSIN DD *
ARCH=10
BOPT IN=DD:SYSBIN(!SUB1|SUB2) OUT=DD:SYSBOUT
```

例 6. 複数の BOPT オプティマイザー・ディレクティブを使用して、複数の入力モジュールを指定

次の例では、BOPT オプティマイザー・ディレクティブを使用して複数の入力モジュールが指定されています。

```
...  
//SYSIN DD *  
  ARCH=10  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=HLQ.OUT.LOAD(MEM1)  
  BOPT IN=HLQ.IN.LOAD(MEM5) OUT=HLQ.OUT.LOAD(MEM5)
```

例 7. REPLACE オプションで最適化またはスキャンをバイパス

次の例では、REPLACE オプションが使用されています。関連付けられている出力モジュールが既に存在する場合、REPLACE=N で、最適化またはスキャンをバイパスします。この例では、オプティマイザーが最初と 3 番目の BOPT ディレクティブで最適化を行います。オプティマイザーは、2 番目の BOPT ディレクティブにおける最適化と、REPLACE=N が指定された最後の BOPT ディレクティブにおけるスキャンをバイパスします。最初と 3 番目の BOPT ディレクティブにおいて出力モジュールが作成されます。

```
...  
//SYSIN DD *  
  ARCH=10  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=HLQ.OUT.LOAD(MEM1) REPLACE=Y  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=HLQ.OUT.LOAD(MEM1) REPLACE=N  
  BOPT IN=HLQ.IN.LOAD(*) OUT=HLQ.OUT.LOAD REPLACE=Y  
  BOPT IN=HLQ.IN.LOAD(*) OUT=HLQ.OUT.LOAD SCAN=Y REPLACE=N
```

例 8. グローバルおよびローカルのオプティマイザー・オプションの指定

次の例では、グローバル・オプションとローカル・オプションが使用されています。ARCH=11 と REPLACE=N が最初の BOPT ディレクティブに適用され、ARCH=10 と REPLACE=Y が 2 番目の BOPT ディレクティブに適用されます。

```
...  
//SYSBOUT DD DSN=HLQ.OUT.LOAD,DISP=SHR  
//SYSIN DD *  
  ARCH=10 REPLACE=N  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=DD:SYSBOUT(MEM1) ARCH=11  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=DD:SYSBOUT(MEM1) REPLACE=Y
```

例 9. グローバル SCAN を使用してスキャン・モードに切り替える

次の例では、グローバル・オプション SCAN によって、入力モジュールの最適化から入力モジュールのスキャンにモードが変更されます。このスキャン・モードでは、出力モジュールは書き込まれません。OUT オプションは、例えば正しい構文であれば処理されますが、そうではない場合は無視されます。SCAN オプションについて詳しくは、[26 ページの『SCAN』](#)を参照してください。

2 番目の BOPT ディレクティブで、オプティマイザーは最適化モードに戻ります。

```
...  
//SYSBOUT DD DSN=HLQ.OUT.LOAD,DISP=SHR  
//SYSIN DD *  
  ARCH=11 SCAN=Y  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=DD:SYSBOUT(MEM1)  
  BOPT IN=HLQ.IN.LOAD(MEM1) OUT=DD:SYSBOUT(MEM1) SCAN=N
```

例 10. ALLOW オプションの指定

次の例では、2 番目の BOPT ディレクティブに関してはローカル・オーバーライドが行われ、それ以外のディレクティブに関してはグローバル設定 ALLOW=NOUNRESEXE が指定されています。

最初 (MEM1) と 3 番目 (MEM3) の BOPT ディレクティブは ALLOW=NOUNRESEXE を使用して実行されます (つまり、完全にバインドされたプログラム・モジュールのみが受け入れられます)。部分的にバインドされたプログラム・モジュールが検出された場合は、BOZ1494S メッセージが発行されます。

2 番目 (MEM2) の BOPT ディレクティブは ALLOW=UNRESEXE を使用して実行されるため、完全にバインドされたプログラム・モジュールと部分的にバインドされたプログラム・モジュールの両方が入力として受け入れられます。

```
...  
//SYSBOUT DD DSN=HLQ. OUT. LOAD, DISP=SHR  
//SYSIN DD *  
ARCH=11 ALLOW=NOUNRESEXE  
BOPT IN=HLQ.IN. LOAD (MEM1) OUT=DD:SYSBOUT (MEM1)  
BOPT IN=HLQ.IN. LOAD (MEM2) OUT=DD:SYSBOUT (MEM2) ALLOW=UNRESEXE  
BOPT IN=HLQ.IN. LOAD (MEM3) OUT=DD:SYSBOUT (MEM3)
```

IEFOPZ で最適化を指定

以下の例は、IEFOPZ オプティマイザー・ディレクティブを使用する JCL を示しています。

このセクションの例は完全なものではありません。ユーザーが SYSIN ファイルに何を指定しなければならないかを示すことを目的にしています。基本 JCL 構成については、[65 ページの『付録 A JCL サンプル』](#)を参照してください。

例 1. 単一 ARCH 構成

この例は、ABO を IEFOPZ ディレクティブで実行するための、最小の JCL を示しています。

ご使用の IEFOPZ 構成に ARCH レベルが 1 つだけ含まれている場合、この JCL を使用できます。

```
...  
//SYSIN DD *  
IEFOPZ
```

例 2. 複数 ARCH 構成

ご使用の IEFOPZ 構成に複数の ARCH レベルが含まれている場合、リスト・ファイル名の競合を避けるために、IEFOPZ ディレクティブ行を個別に指定してください。

```
...  
//SYSIN DD *  
IEFOPZ SEL_ARCH=11 LIST=HLQ.OUT1.ARCH11.LIST  
IEFOPZ SEL_ARCH=12 LIST=HLQ.OUT1.ARCH12.LIST
```

例 3. SEL_STATE および SEL_ARCH セレクターによる最適化の制限

次の例では、INACTIVE とマークされている OLDNEW マッピング用に最適化されたモジュールが作成されます。例 2 と同様、マッピングに複数の ARCH レベルが含まれている場合、リスト・ファイル名の競合を避けるために LIST オプションが使用されます。

```
...  
//SYSIN DD *  
IEFOPZ SEL_STATE=INACTIVE SEL_ARCH=10 LIST=HLQ.OUT.ARCH10.LIST  
IEFOPZ SEL_STATE=INACTIVE SEL_ARCH=11 LIST=HLQ.OUT.ARCH11.LIST
```

例 4. SEL_OLD セレクターによる最適化の制限

次の例では、指定されたパターンに一致する OLD データ・セットのメンバー用に最適化されたモジュールが作成されます。

1 行目で、HLQ.IN.LOAD のすべてのメンバー用に最適化されたモジュールが作成されます。

2 行目で、HLQ.IN.* に一致する OLD データ・セットにおいて *M に一致するデータ・セット・メンバー用に最適化されたモジュールが作成されます。1 行目のモジュールが再び最適化されることを避けるために、2 行目には REPLACE=N が指定されています。

```
...  
//SYSIN DD *
```

```
IEFOPZ SEL_OLD=HLQ.IN.LOAD  
IEFOPZ SEL_OLD=HLQ.IN*.(M*) REPLACE=N
```

z/OS JCL REGION パラメーターおよび JCL MEMLIMIT パラメーターの推奨設定

ABO 最適化手法

ABO は、Enterprise COBOL V4 以前で実現されていたモジュールを凌駕する最適化済み高性能モジュールを生成するために、拡張分析を行い、かなりのマシン・リソースが必要となるコード最適化手法を使用します。

また、ABO は、コンパイルされた多くのプログラムで同時かつ容易に呼び出されるため、要求される処理の量が原因で、全般的なリソースの使用量が増える可能性があります。完全なデータ・セットは 1 つで多くのモジュールから構成されていることがあります。各モジュールは、それ自体で多くのコンパイル済みプログラム (CSECT) を含んでいる可能性があります。ABO を使用すれば、それらをまとめて最適化できます。そのため、何百何千というコンパイル済みプログラムを同時に最適化するために ABO で必要とされる合計リソースは、一度に 1 つのリソース・ファイルに対して行われるコンパイル処理に比べて多くなります。

モジュールの最適化に必要とされる時間とメモリーは以下の要因によって決定されます。

- モジュールに含まれる CSECT の数。
- 各 CSECT の複雑さ。複雑さに影響を与えるものは、コンパイル済み PROCEDURE DIVISION ステートメントのサイズと、入力プログラムの DATA DIVISION のサイズです。

z/OS JCL REGION パラメーターと JCL MEMLIMIT パラメーターを適切に設定

65 ページの『付録 A JCL サンプル』に示されているように、作動に必要なメモリーを ABO に割り当てるには、JCL REGION パラメーターに OM を設定する必要があります。

ABO は巨大な CSECT を最適化するために、2 GB 境界を超えるストレージを使用します。つまり、ABO 処理を正常に完了させるためには、十分大きい値を z/OS MEMLIMIT パラメーターに設定する必要があります。

OM の推奨 REGION 設定では、MEMLIMIT は NOLIMIT に設定されます。ただし、この NOLIMIT 値は、JOB/EXEC ステートメント指定された MEMLIMIT 設定値によって、または出口ルーチン IEFUSI によって、オーバーライドされて小さい値になる可能性があります。ABO によって要求されるストレージの量は、最適化される CSECT の数とサイズによって異なります。

MEMLIMIT=NOLIMIT がオーバーライドされて小さい値になった場合に、この MEMLIMIT 設定値が十分な大きさではないと、以下のいずれかの ABO メッセージが表示されることがあります。

```
BOZ1145U Insufficient memory in the compiler to continue compilation.  
BOZ1428U Insufficient memory encountered during binder API "&1": return code=&2  
         reason code=&3. Terminating optimizer.  
BOZ1449U Unhandled out of memory exception
```

非常に大きい CSECT を最適化したり、小さい CSECT を非常に多く最適化したりする場合は、MEMLIMIT 設定値として 10GB 以上が必要になる可能性があります。メッセージ BOZ1145U、BOZ1428U、または BOZ1449U のいずれかが表示される場合は、MEMLIMIT 設定値を増やして大きい値にします。

JCL REGION パラメーターおよび JCL MEMLIMIT パラメーターについて詳しくは、「[z/OS MVS Initialization and Tuning Reference](#)」および「[z/OS MVS Initialization and Tuning Guide](#)」を参照してください。

ABO メッセージに使用される言語を指定

CEEOPTS DD は、ABO によって生成されるメッセージの言語を指定するために使用されます。

デフォルトで、メッセージは英語で表示されます。日本語メッセージが表示されるよう指定するには、次のコードを JCL に追加してください。

```
//CEEOPTS DD *  
              NATLANG(JPN)  
/*
```

TSO、REXX、およびアセンブラー・コードからの ABO の起動

このセクションでは、TSO、REXX、およびアセンブラーから ABO を起動する方法について説明します。

TSO のもとでの最適化

TSO のもとでは、TSO コマンド、コマンド・リスト (CLIST)、REXX EXEC、または ISPF を使用すると、従来の MVS データ・セットを使用する COBOL プログラムを最適化することができます。z/OS UNIX ファイルを使用するプログラムを最適化する場合は、TSO コマンドまたは REXX EXEC を使用することができます。

いずれの方式を使用する場合も、以下のように、データ・セットを割り振り、最適化を要求しなければなりません。

1. ALLOCATE コマンドを使用してデータ・セットを割り振ります。データ・セットはどんな順序で割り振っても構いません。ただし、最適化を開始する前に、必要なすべてのデータ・セットを割り振っておく必要があります。
2. SYSIN データ・セット内にオプティマイザー・パラメーターを指定します。
3. READY プロンプトで CALL コマンドを使用して最適化を要求します: CALL 'hlqboz.SB0ZMOD1(B0ZOPT)'

ALLOCATE コマンドおよび CALL コマンドは、TSO コマンド行に指定することができます。また、z/OS UNIX ファイルを使用していない場合は、これらのコマンドを CLIST に指定することができます。

PDS ライブラリーでも PDSE ライブラリーではない z/OS UNIX ファイルは、すべてのオプティマイザー・データ・セットに対して割り振ることができます。例えば、ABO パラメーターが UNIX ファイル /u/myu/abo.parms に格納されている場合、ALLOCATE ステートメントの形式は次のようになります。

```
Allocate File(SYSIN) Path('/u/myu/abo.parms') Pathopts(ORDONLY) Filedata(TEXT)
```

TSO のもとで最適化するための ALLOCATE および CALL

次の例は、TSO のもとで最適化するときの ALLOCATE コマンドおよび CALL コマンドの指定方法を示します。ファイルはいずれも新規ファイルでも既存ファイルでもかまいませんが、入力ファイル SYSIN および SYSBIN は最適化の開始前に存在していなければならないことに注意してください。

```
[READY]  
FREE F(SYSIN) SYSIN OPTLOG OPTERR CEEDUMP SYSBIN  
SYSBOUT  
[READY]  
ALLOC F(SYSIN) DA(ABO.LISTING) SPACE(10,10) CYL NEW CATALOG RECFM(V B) LRECL(512)  
BLKSIZE(27998) DSORG(PS)  
[READY]  
ALLOC FI(OPTLOG) DA(ABO.OPTLOG) SPACE(10,10) CYL NEW CATALOG RECFM(V B) LRECL(512)  
BLKSIZE(27998) DSORG(PS)  
[READY]  
ALLOC FI(OPTERR) DA(ABO.OPTERR) SPACE(10,10) CYL NEW CATALOG RECFM(V B) LRECL(512)  
BLKSIZE(27998) DSORG(PS)  
[READY]  
ALLOC FI(CEEDUMP) DA(ABO.CEEDUMP) SPACE(50,10) CYL NEW CATALOG RECFM(F B) LRECL(133)
```

```

BLKSIZE(27930) DSORG(PS)
[READY]
ALLOC FI(SYSIN) DA(ABO.SYSIN) SHR          /* supply ABO parameters within SYSIN file
*/
[READY]
ALLOC FI(SYSBIN) DA(IN.LOAD) SHR           /* supply COBOL load library to be optimized */
[READY]
ALLOC FI(SYSBOUT) DA(OUT.LOAD) SHR         /* supply load library for optimized load modules */
[READY]
CALL 'hlqboz.SBOZMOD1(BOZOPT)'
[READY]
FREE F(SYSPRINT SYSIN OPTLOG OPTERR CEEDUMP SYSBIN SYSBOUT)

```

TSO のもとで最適化するための CLIST

次の例は、TSO のもとで最適化するための CLIST を示しています。

```

PROC 1 HLQBOZ
FREE F(SYSPRINT SYSIN OPTLOG OPTERR CEEDUMP SYSBIN
SYSBOUT)
ALLOC FI(SYSPRINT) DA(ABO.LISTING) SHR
ALLOC FI(SYSIN) UNIT(SYSDA) SPACE(1,0) TRACKS NEW +
RECFM(F B) LRECL(80) BLKSIZE(800) DSORG(PS)
OPENFILE SYSIN OUTPUT
SET SYSIN = &STR(BOPT IN=DD:SYSBIN(MEMBER) OUT=DD:SYSBOUT(MEMBER))
PUTFILE SYSIN
CLOSEFILE SYSIN
ALLOC FI(OPTLOG) DA(ABO.OPTLOG) SHR
ALLOC FI(OPTERR) DA(ABO.OPTERR) SHR
ALLOC FI(CEEDUMP) DA(ABO.CEEDUMP) SHR
ALLOC FI(SYSBIN) DA(IN.LOAD) SHR
ALLOC FI(SYSBOUT) DA(OUT.LOAD) SHR
CALL '&HLQBOZ..SBOZMOD1(BOZOPT)'
FREE F(SYSIN SYSPRINT OPTLOG OPTERR CEEDUMP SYSBIN SYSBOUT)

```

TSO のもとで最適化するための REXX

次の例は、TSO のもとで最適化するための REXX を示しています。

```

/*=====*>> REXX <<=====*/
Parse Arg hlqboz .                               /* get argument */
IF HLQBOZ = '' THEN DO
SAY 'HLQBOZ ARGUMENT MISSING'
EXIT
END
ADDRESS TSO
msgstat = MSG("OFF")
"FREE FILE (SYSIN SYSPRINT)"
"ALLOC FI(SYSPRINT) DA(ABO.LISTING) SHR"
"ALLOC FI(OPTLOG) DA(ABO.OPTLOG) SHR"
"ALLOC FI(OPTERR) DA(ABO.OPTERR) SHR"
"ALLOC FI(CEEDUMP) DA(ABO.CEEDUMP) SHR"
"ALLOC FI(SYSBIN) DA(IN.LOAD) SHR"
"ALLOC FI(SYSBOUT) DA(OUT.LOAD) SHR"
"ALLOC FI(SYSIN) NEW CYL SPACE(1,1) RECFM(F B)",
" LRECL(80) BLKSIZE(800) DSORG(PS)"
line.1 = 'ARCH=11'
line.2 = 'BOPT IN=DD:SYSBIN(MEMBER) OUT=DD:SYSBOUT(MEMBER)'
"EXECIO 2 DISKW SYSIN (STEM line. FINIS"
"CALL '&hlqboz'.SBOZMOD1(BOZOPT)"
"FREE FI(OPTLOG OPTERR CEEDUMP SYSPRINT SYSBIN SYSBOUT SYSIN)"
msgstat = MSG("ON")

```

TSO バッチでの最適化、OPTLOG および SYSPRINT 出力をライブラリー・メンバーに送信するための REXX。

場合によっては、複数の BOPT ステートメントを作成せずにロード・ライブラリー全体または複数ロード・ライブラリーの各メンバーを最適化しなければならないことがあります。

次の例は、SYSEXEC DD 名によって参照されるデータ・セットに含まれる ABOMEMBS REXX を実行するために使用される TSO バッチ・ジョブを示しています。ABOMEMBS は、SYSBIN データ・セットを連結し、その連結内にある各データ・セット・メンバーを個別に最適化します。

```
//OPTMEMBS EXEC PGM=IKJEFT01,PARM='ABOMEMBS',
REGION=0M
//STEPLIB DD DISP=SHR,DSN= hlqboz.SBOZMOD1
//SYSEXEC DD DISP=SHR,DSN=hlq.CLIST /* supply ABOMEMBS REXX member within SYSEXEC
library */
//SYSTSPRT DD SYSOUT=*,DCB=(LRECL=132,RECFM=FBA,BLKSIZE=1320)
//SYSTSIN DD DUMMY
//OPTERR DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSBIN DD DISP=SHR,DSN=hlq.INLOAD1 /* supply one or more COBOL load libraries to be
optimized */
// DD DISP=SHR,DSN=hlq.INLOAD2
// DD DISP=SHR,DSN=hlq.INLOAD3
//SYSBOUT DD DISP=SHR,DSN=hlq.OUTLOAD
//OPTLOG DD DISP=(,CATLG),DSN=hlq.OPTLOG,UNIT=3390, /* supply new or existing PDS/PDSE or
SEQ file */
// SPACE=(CYL,(5,5,20)),DSNTYPE=LIBRARY,
// DCB=(RECFM=VB,LRECL=512,BLKSIZE=0)
//SYSPRINT DD DISP=(,CATLG),DSN=hlq.SYSPRINT,UNIT=3390, /* supply new or existing PDS/PDSE or
SEQ file */
// SPACE=(CYL,(5,5,20)),DSNTYPE=LIBRARY,
// DCB=(RECFM=VB,LRECL=512,BLKSIZE=0)
//SYSIN DD *
ARCH=11
SCAN=N
# optionally put BOPTs below this line for specific library members optimization
BOPT IN=hlq.INLOAD7(A*) OUT=hlq.OUTLOAD LOG=hlq.OPTLOG LIST=hlq.SYSPRINT
/*
```

上記 SYSBIN DD 連結にあるメンバーごとに ABOMEMBS は対象のメンバーに対して個別に割り振られた SYSIN ファイル、OPTLOG ファイル、SYSBIN ファイル、および SYSPRINT ファイルを使用して ABO を起動します。

例えば、hlq.INLOAD1 データ・セットに含まれるメンバー MEM1 は、上記 JCL 内で割り振られている OPTLOG ファイルと SYSPRINT ファイルが PDS データ・セットまたは PDSE データ・セットである場合に、SYSBIN ファイル hlq.INLOAD1 を割り振り、OPTLOG ファイル hlq.OPTLOG(MEM1) および SYSPRINT ファイル hlq.SYSPRINT(MEM1) を再割り振りします。それ以外の場合、それらのファイルの JCL 割り振りが引き続き使用されます。

MEM1 用の SYSIN ファイルを構成するためには、上の JCL SYSIN ファイルで最初の BOPT ステートメントまでにリストされている ABO パラメーターがすべて使用されます。その後で、その SYSIN ファイルが、内部的に生成された BOPT ステートメントで追加されます。以下は、MEM1 の SYSIN ファイルの例です。

```
=====
ARCH=11
SCAN=N
BOPT IN=DD:SYSBIN(MEM1) OUT=DD:SYSBOUT(MEM1)
=====
```

個々の SYSIN ファイル、OPTLOG ファイル、SYSPRINT ファイル、および SYSBIN ファイルがすべて割り振られると、MEM1 を最適化するために ABO が呼び出されます。

ABOMEMBS exec は、このプロセスを SYSBIN 連結内のメンバーごとに繰り返します。上記 JCL SYSIN ファイルに BOPT ステートメントが含まれている場合は、SYSBIN 連結内のメンバーがすべて最適化された後で、ABOMEMBS は OPTLOG ファイルと SYSPRINT ファイルを SYSOUT に再割り振りし、オリジナルの JCL SYSIN ファイルと同じステートメントをすべて含む SYSIN ファイルを再割り振りし、もう一度 ABO を起動して、SYSIN ファイルで明示的に指定されたパラメーターを使用して処理を続行します。

この方法は、すべてのライブラリー・メンバーを最適化する場合に役立ちます。その際に行うことは、追加の BOPT を明示的に指定せずにライブラリー・メンバーを SYSBIN 連結に組み込み、オプションで、一部のライブラリー・メンバーのみを最適化する必要がある場合に追加の BOPT ステートメントを組み込むだけです。上記 JCL 内の JCL カードは任意の順序で指定できます。

ABOMEMBS は以下の SYSTSPRT 出力を生成します。

```
===== Processing hlq.INLOAD1 data set =====
Member ADD10 processed, rc=4
Member CALLEE processed, rc=0
Member CALLEE1 processed, rc=0
...
===== Processing hlq.INLOAD2 data set =====
```

```

Member COBPGM processed, rc=0
Member DJSIEV85 processed, rc=0
Member IAMASM processed, rc=4
...
===== Processing hlq.INLOAD3 data set =====
Member TEMPNAM6 processed, rc=12
Member TEMPNAM7 processed, rc=0
Member TEMPNAM8 processed, rc=0
...
===== Processing SYSIN statements =====
ARCH=11
SCAN=N
BOPT IN=hlq.INLOAD7(A*) OUT=hlq.OUTLOAD LOG=hlq.OPTLOG LIST=hlq.SYSPRINT
=====
READY
END

```

以下は、ABOMEMBS REXX ソース・コードの例です。

```

BROWSE      hlq.CLIST(ABOMEMBS)
Command ==>
***** Top of Data
/*=====> REXX <=====*/
Parse Arg

X = LISTDSI(STEPLIB FILE)
If x > 0 Then Do
    Say 'Check STEPLIB allocation' ; Exit
End
optimizer = ""||SYSDSNAME||'(BOZOPT)'||""
flaglog = 0 /* assume OPTLOG is sequential */
X = LISTDSI(OPTLOG FILE)
IF x = 0 & SYSDSORG = 'PO' Then Do
    optlogds = SYSDSNAME ; flaglog = 1
End
flaglist = 0 /* assume SYSPRINT is sequential */
X = LISTDSI(SYSPRINT FILE)
IF x = 0 & SYSDSORG = 'PO' Then Do
    sysprtds = SYSDSNAME ; flaglist = 1
End
"EXECIO * DISKR SYSIN (STEM line. FINIS" /* read in SYSIN parameters into
line. array */
n = line.0 + 1 /* assume no explicit BOPT specified. n is per-member BOPT
line number */
Do l = 1 To
line.0
    PARSE UPPER VALUE line.l WITH
line.l
    If POS('BOPT ',line.l) > 0 Then
Do
    n = l ; flagbopt = 1 ; leave /* leave when first BOPT found
*/
End
    line.l = line.l /* copy all lines before first BOPT into linea. array
*/
End

X = LISTDSI(SYSBIN
FILE)
If x > 0 Then
Do
    Say 'Check SYSBIN allocation' ;
Exit
End

x =
outtrap(concatl.)
lista
status

```

```

      x =
outtrap(off)
      /* i2 is a line with last dsn in SYSBIN concatenation
*/
      i2 = concatl.0 - 1          /* assume that SYSBIN concatenation is a last in JCL
*/
      Do i = 1 To
concatl.0
          If substr(concatl.i,3,6) = 'SYSBIN' Then
Do
            i1 = i - 1 ; leave    /* i1 line with first dsn in concatenation
*/
End
      End
      k = i + 2
      If k < concatl.0 Then Do
          Do i = k To concatl.0 by 2
              If substr(concatl.i,3,1) <> ' ' Then Do
                  i2 = i - 3 ; leave /* i2 line with last dsn in concatenation */
              End
          End
      End
      Else i2 = i1                /* when SYSBIN DD with a single data set is a last JCL card
*/

      msgstat = MSG("OFF")

      Do i = i1 To i2 by 2 /* loop trough SYSBIN concatenation */
          sysbinds = strip(concatl.i)
          Say '===== Processing 'sysbinds' data set ====='
          x=outtrap('row.')
              Address TSO "LISTDS '"sysbinds'" members"
          x=outtrap('off')
          If row.0 < 6 Then Do
              Say 'Check SYSBIN file' ; Exit
          End
          DO J = 7 TO row.0 /* loop trough member list */
              PARSE VALUE row.J WITH memn alias
              "FREE FI(SYSIN)"
              "ALLOC FI(SYSIN) NEW CYL SPACE(1,1) RECFM(F B)",
              " LRECL(80) BLKSIZE(800) DSORG(PS)"
              linea.n = 'BOPT IN=DD:SYSBIN('||memn||') ',
              'OUT=DD:SYSBOUT('||memn||')'
              "EXECIO * DISKW SYSIN (STEM linea. FINIS"
              If flaglog = 1 Then Do
                  "FREE FI(OPTLOG)"
                  optlogm = "||optlogds||('||memn||')||"
                  "ALLOC FI(OPTLOG) DA("optlogm") SHR"
              End
              If flaglist = 1 Then Do
                  "FREE FI(SYSPRINT)"
                  sysprtm = "||sysprtds||('||memn||')||"
                  "ALLOC FI(SYSPRINT) DA("sysprtm") SHR"
              End
              "FREE FI(SYSBIN)"
              "ALLOC FI(SYSBIN) DA('"sysbinds"') SHR"
              "CALL "optimizer /* optimize member */
              Say 'Member 'memn' processed, rc='rc
          END /* end of member list loop */
      End /* end of concatenation list loop */
      If flagbopt = 1 Then Do
          "FREE FI(SYSIN)"
          "ALLOC FI(SYSIN) NEW CYL SPACE(1,1) RECFM(F B)",
          " LRECL(80) BLKSIZE(800) DSORG(PS)"
          "EXECIO * DISKW SYSIN (STEM line. FINIS"
          "FREE FI(OPTLOG)"
          "ALLOC FI(OPTLOG) SYSOUT"
          "FREE FI(SYSPRINT)"
          "ALLOC FI(SYSPRINT) SYSOUT"
          "CALL "optimizer /* optimize with BOPTs provided in SYSIN */

```

```

    Say '===== Processing SYSIN statements ====='
    Do i = 1 To line.0
        Say line.i
    End
    Say '===== '
End
msgstat = MSG("ON")
EXIT

```

アセンブラー・プログラムからオブティマイザーを開始する

HLASM プログラム内からプログラマチックに ABO を起動できます。

最適化を始める前に、以下のステップを実行してください。

1. アセンブラー・プログラム内で動的割り振りを使用するか、またはアセンブラー・プログラム起動に使用されるジョブ JCL で DD カードを指定して、必要なデータ・セットをすべて割り振ります。割り振りが必要となる DD 名は、SYSPRINT、SYSBIN、SYSBOUT、SYSIN、OPTLOG、OPTERR、CEEDUMP です。
2. SYSIN データ・セット内に ABO パラメーターを指定します。

LINKX マクロまたは ATTACHX マクロを使用してアセンブラー・プログラム内から ABO を開始できます。これは、これら 2 つのマクロが 64 ビット・モード互換であり、ABO が AMODE 64 で実行されるためです。

以下は、リスト形式での LINKX マクロの例です。

```
symbol {LINKX} EP=BOZOPT,AMODE64OK=YES,PLIST8=YES,SF=L
```

以下は、実行形式での LINKX マクロの例です。

```
LINKX EP=BOZOPT,AMODE64OK=YES,PARAM=(addr),PLIST8=YES,      *
      MF=(E,#LINKX),SF=(E,#LINK2)
```

は接頭部記号として使用されます。

EP

ABO のシンボル名を指定します。

PARAM

アセンブラー・プログラムから ABO に渡されるアドレス・パラメーター・リストを指定します。この例では、*addr* は、SYSIN ファイルで指定されたパラメーターを直接読み取る BOZOPT プログラムによって無視されるため任意の値にすることができます。

PLIST8=YES

これは、PARAM キーワードに基づいて LINKX によって各エントリー 8 バイトのパラメーター・リストとして作成されるパラメーター・リストのパラメーター・リスト・エントリーのサイズを定義します。

AMODE64OK=YES

これは、AMODE 24 ルーチンまたは AMODE 31 ルーチンから AMODE 64 ターゲット・ルーチンにリンクしようとする試みをシステムが受け入れるように指示します。

SF=L

LINKX マクロのリスト形式を指定します。

ABO は、処理を完了すると、レジスター 15 に戻りコードを書き込みます。

オブティマイザーを開始するアセンブラー・プログラム

以下の例はアセンブラー・プログラムからの ABO 起動を示しています。

```

//JOB CARD JOB
//*
//ASM HCL  PROC  MAC='SYS1.MACLIB',MAC1='SYS1.MODGEN',U=3390,
//              MAC2='SYS1.MACLIB'

```

```

/*-----***
/*          A S M H C L      H-ASSEMBLER          ***
/*          IBM-PROCEDURE:  COMPILE  +  LINK          ***
/*-----***
//ASM      EXEC  PGM=ASMA90,PARM=OBJECT,REGION=0M
//SYSLIB   DD    DSN=&MAC,DISP=SHR
//         DD    DSN=&MAC1,DISP=SHR
//         DD    DSN=&MAC2,DISP=SHR
//SYSUT1   DD    DSN=&&SYSUT1,UNIT=&U,SPACE=(TRK,(60,45))
//SYSPRINT DD    SYSOUT=*,DCB=BLKSIZE=1089
//SYSPUNCH DD    DUMMY
//SYSLIN   DD    DSN=&&OBJSET,UNIT=&U,SPACE=(80,(2000,500)),
//         DISP=(MOD,PASS)
//LKED     EXEC  PGM=HEWLH096,
//         PARM='XREF,LET,LIST,AC=0,FILL=NONE',
//         COND=(8,LT,ASM)
//SYSLIN   DD    DSN=&&OBJSET,DISP=(OLD,DELETE)
//         DD    DDNAME=SYSIN
//SYSLMOD  DD    DSN=h1q.CALLABO.LOAD,DISP=SHR
//SYSUT1   DD    DSN=&&SYSUT1,UNIT=(&U,SEP=(SYSLIN,SYSLMOD)),
//         SPACE=(1024,(50,20))
//SYSPRINT DD    SYSOUT=*
//         PEND
//
//          EXEC  ASMHCL
//ASM.SYSIN DD
*
          YREGS
CALLABO   CSECT
CALLABO   AMODE 31
CALLABO   RMODE ANY
          STM    R14,R12,12(R13)
          USING  CALLABO,R12
          LR     R12,R15      LOAD BASE REGISTER
          LA     R9,SAVE      POINT TO CURRENT SAVEAREA
          ST     R9,8(R13)    A(CURRENT_SA) IN OLD_SA
          ST     R13,4(R9)    A(OLD_SA) IN CURRENT_SA
          LR     R13,R9       R13=A(CURRENT_SAVEAREA)
*-----*
          LLGTR  R12,R12      allow 64-bit
addressability
          BAL    R14,LINKOPT  invoke BOZOPT
*-----*
DONE      DS     0H
          L      R13,4(R13)
          ST     R15,16(R13)
          LM     R14,R12,12(R13)
          BR     R14
*-----*
LINKOPT   DS     0H
          XR     R15,R15      nullify parm addr, BOZOPT ignores
it
          LLGTR  R15,R15
          SAM64
          LINKX  EP=BOZOPT,AMODE640K=YES,PARAM=((R15)),PLIST8=YES,
          MF=(E,#LINKX),SF=(E,#LINK2)
          SAM31
          B      DONE
*-----*
SAVE      DS     0F
          DC     XL72'00'
#LINKX    LINKX  EP=BOZOPT,AMODE640K=YES,PLIST8=YES,SF=L
#LINK2    LINKX  EP=BOZOPT,AMODE640K=YES,PLIST8=YES,SF=L
          LTORG
          END
//LKED.SYSIN DD *
          PAGE  CALLABO
          NAME  CALLABO(R)
//

```

次のサンプル JCL は CALLABO プログラムを実行します。

```

//OPT      EXEC  PGM=CALLABO,REGION=0M
//STEPLIB  DD    DSN=h1qboz.SBOZMOD1,DISP=SHR
//         DD    DSN=h1q.CALLABO.LOAD,DISP=SHR
//OPTLOG   DD    SYSOUT=*
//OPTERR   DD    SYSOUT=*
//CEEDUMP  DD    SYSOUT=*
//SYSPRINT DD    SYSOUT=*
//SYSBIN   DD    DSN=h1q.IN.LOAD,DISP=SHR
//SYSBOUT  DD    DSN=h1q.OUT.LOAD,DISP=SHR

```

```
//SYSIN      DD *  
# supply ABO directives down this line  
BOPT IN=DD:SYSBIN(*) OUT=DD:SYSBOUT
```

第 6 章 最適化処理からの出力を理解

ログ・ファイル

IBM Automatic Binary Optimizer for z/OS は、問題の識別および解決に役立つログ・ファイルを生成します。

ABO によって OPTLOG DD に無条件に生成されるログ・ファイルを調べることによって、最適化の時点で問題を診断できます (15 ページの表 6 を参照)。ログ・ファイルには、最適化およびスキャンに関する診断情報が入っています。

追加メンバー・レベル・ログ・ファイルを生成するには、LOG オプションを使用します。

最適化に関するログ・ファイル

最適化がモジュールに対して実行される (SCAN=N オプティマイザー・オプションが有効になっている) 場合、ログ・ファイルに入る情報は以下のとおりです。

- 処理されている入力モジュールのファイル名の情報
- 最適化されている CSECT の名前
- 最適化された CSECT ごとの、リスト変換のファイル名の情報
- 最適化された出力モジュールのファイル名の情報
- OPTLOG の各行のタイム・スタンプ、およびその日付のヘッダー
- エラー・メッセージを含む、その他の診断情報

例 1:

次のログ・ファイルは、データ・セット *HLQ.IN1.LOAD* のメンバーである MEMA および MEMB にある COBOL CSECT が最適化されていることを示しています。モジュール MEMA には、SUB1 および SUB2 という名前の、最適化される 2 つの COBOL CSECT があります。モジュール MEMB には、PROGB という名前の、最適化される 1 つの COBOL CSECT があります。リスト変換はすべて、デフォルト *SYSPRINT DD* に置かれます。最適化されたモジュールは、*HLQ.OUT1.LOAD* データ・セットに書き込まれます。

```
5697-AB2 IBM Automatic Binary Optimizer for z/OS 2.1.0

===== Sept 24 2019 =====
10:53:16 Optimizer build level: tr_v21_binopt_20190730_1658_t54Rw7MMEem46oTHworqTQ
(Jul 30 2019 20:05:19)
10:53:16 Processing HLQ.IN1.LOAD, member MEMA
10:53:16   Processing CSECT SUB1, in member MEMA
10:53:16     Optimizing CSECT SUB1 for zEC12
10:53:16     Succeeded in optimizing SUB1
10:53:16     Generating listing transform into DD:SYSPRINT
10:53:16   Processing CSECT SUB2, in member MEMA
10:53:16     Optimizing CSECT SUB2 for zEC12
10:53:16     Succeeded in optimizing SUB2
10:53:16     Generating listing transform into DD:SYSPRINT
10:53:16   Finished processing, processed 2 of 2 CSECTs in member MEMA
10:53:16   Save HLQ.OUT1.LOAD (MEMA) succeeded
10:53:16 Processing HLQ.IN1.LOAD, member MEMB
10:53:16   Processing CSECT PROGB, in member MEMB
10:53:16     Optimizing CSECT PROGB for zEC12
10:53:16     Succeeded in optimizing PROGB
10:53:16     Generating listing transform into DD:SYSPRINT
10:53:16   Finished processing, processed 1 of 1 CSECTs in member MEMB
10:53:16   Save HLQ.OUT1.LOAD (MEMB) succeeded
10:53:16 Exiting with return code: 0
```

例 2:

以下のログ・ファイルは、LOG=HLQ.LOG.OUT が指定された前の例と同じ HLQ.IN1.LOAD によって生成されたメンバー・レベル・ログ・ファイルの出力を示しています。

HLQ.LOG.OUT(MEMA) の内容:

```
5697-AB2 IBM Automatic Binary Optimizer for z/OS 2.1.0
===== Sept 24 2019 =====
10:53:16 Optimizer build level: tr_v21_binopt_20190730_1658_t54Rw7MMEem46oTHworqTQ (Jul 30 2019 20:05:19)
10:53:16 Processing HLQ.IN1.LOAD, member MEMA
10:53:16   Processing CSECT SUB1, in member MEMA
10:53:16     Optimizing CSECT SUB1 for zEC12
10:53:16     Succeeded in optimizing SUB1
10:53:16     Generating listing transform into DD:SYSPRINT
10:53:16   Processing CSECT SUB2, in member MEMA
10:53:16     Optimizing CSECT SUB2 for zEC12
10:53:16     Succeeded in optimizing SUB2
10:53:16     Generating listing transform into DD:SYSPRINT
10:53:16   Finished processing, processed 2 of 2 CSECTs in member MEMA
10:53:16   Save HLQ.OUT1.LOAD (MEMA) succeeded
10:53:16 Exiting with return code: 0
```

HLQ.LOG.OUT(MEMB) の内容:

```
5697-AB2 IBM Automatic Binary Optimizer for z/OS 2.1.0
===== Sept 24 2019 =====
10:53:16 Optimizer build level: tr_v21_binopt_20190730_1658_t54Rw7MMEem46oTHworqTQ (Jul 30 2019 20:05:19)
10:53:16 Processing HLQ.IN1.LOAD, member MEMB
10:53:16   Processing CSECT PROGB, in member MEMB
10:53:16     Optimizing CSECT PROGB for zEC12
10:53:16     Succeeded in optimizing PROGB
10:53:16     Generating listing transform into DD:SYSPRINT
10:53:16   Finished processing, processed 1 of 1 CSECTs in member MEMB
10:53:16   Save HLQ.OUT1.LOAD (MEMB) succeeded
10:53:16 Exiting with return code: 0
```

スキャンに関するログ・ファイル

スキャンが実行される (SCAN=Y オプティマイザー・オプションが有効になっている) 場合、ログ・ファイルの内容は以下のとおりです。

- スキャンされている入力モジュールのファイル名の情報
- スキャンされているモジュールの CSECT の名前
- その他の診断メッセージ

例:

次のログ・ファイルは、データ・セット HLQ.MEM1.LOAD のモジュールである MEMA および MEMB がスキャンされていることを示しています。スキャン出力は、モジュール内の CSECT それぞれを示しています。コンパイルに使用された COBOL コンパイラーのバージョンと、CSECT から抽出された「Signature information bytes」が、COBOL CSECT の SUB、SUB2、および PROGB について表示されます。「Signature information bytes」は COBOL プログラミング・ガイド で説明されていて、コンパイル済みプログラムに関する情報を示します。

```
5697-AB2 IBM Automatic Binary Optimizer for z/OS 2.1.0
===== Sept 24 2019 =====
10:58:23 Optimizer build level: tr_v21_binopt_20190730_1658_t54Rw7MMEem46oTHworqTQ (Jul 30 2019 20:05:19)
10:58:23 Processing HLQ.IN1.LOAD, member MEMA
      Language ID Records:
          id 5688187 v21 m00 2015281 resident EDCOEXTS
          id 5655S7100 v42 m00 2015281 resident SUB
      Enterprise COBOL V4: start=0x10, length=4.19 (kBytes)
      Signature information bytes:
```

```

a0487d4c 20000000 00880100 00000040
08000000 000000 00008004 1400
id 5655S7100 v42 m00 2015281 resident SUB2
Enterprise COBOL V4: start=0x10d8, length=4.20 (kBytes)
Signature information bytes:
a0487d4c 20000000 00880100 00000040
08000000 000000 00008004 1400
id 569623400 v01 m06 2013071 resident CEESG005
id 569623400 v01 m06 2013072 resident CEEBETBL
id 569623400 v01 m06 2013072 resident CEESTART
10:58:23 Processing HLQ.IN1.LOAD, member MEMB
Language ID Records:
id 5688187 v21 m00 2015281 resident EDCOEXTS
id 5655S7100 v42 m00 2015281 resident PROGB
Enterprise COBOL V4: start=0x10, length=8.19 (kBytes)
Signature information bytes:
a0487d4c 20000000 00880100 00000040
08000000 000000 00008004 1400
id 569623400 v01 m06 2013072 resident CEESTART
10:58:23 Exiting with return code: 0

```

リスト変換

リスト変換は、既にコンパイルされているプログラム・モジュールに Automatic Binary Optimizer for z/OS が行った変更を記述するものです。入力バイナリーにおける命令が、最適化された命令にどのようにマッピングされているかを示します。リスト変換は、最適化される CSECT ごとに作成されます。リスト変換の目的は、入力バイナリーがソースから最初にコンパイルされたときに生成された、コンパイラー・リストを補完することです。リスト変換はコンパイラー・リストに依存しませんが、リスト変換とコンパイラー・リストを一緒に使用すれば、どのようにして IBM Automatic Binary Optimizer for z/OS がバイナリーを変換したのかを詳しく理解できます。

リスト変換は、最適化された CSECT ごとに無条件で生成されます。つまり、特殊なオプションやフラグを指定する必要はありません。デフォルトで、リスト変換は SYSPRINT DD に生成されます。

また、リスト変換は [61 ページの『Application Delivery Foundation for z Systems』](#) によっても使用されます。

リスト変換の内容

リスト変換は、最適化済みプログラムの実行時に検出された問題を診断するときに役立てるために提供されています。リスト変換は、IBM Debug for z Systems などのデバッグ・ツールで使用するためのものです。

リスト変換には、以下の情報が入っています。

- 最適化オプションの要約
- 入力命令が散在する、最適化された命令
- ABO によって作成された新しいリテラルが入っているリテラル・プール
- 最適化された CSECT に関する情報が入っている PPA4 セクション
- ABO によって作成された新しいスタック・シンボルの、動的ストレージ域 (DSA) マップとも呼ばれる自動マップ
- 最適化対象の CSECT に対する命令の完全なリストとコンパイラー・オプションが含まれている入力命令セクション

最適化パラメーターの要約

このセクションには、プログラムが最適化される対象のアーキテクチャー・レベルの名前、および入力バイナリーとそのバイナリーを作成したコンパイラーの両方の日時スタンプがあります。出力バイナリーの日時スタンプも示されています。

例:

```
Invocation Parameters:
  Architecture Level: zEC12

Input IDRL Record: 5655S7100 v42 m00 2013122
  Name: Enterprise COBOL V4
  Version: 42
  Mod Level: 00
  Compiled Date (YYYYDDD): 2013122

Output IDRL Record: 5697-AB2 v21 m00 2019256
  Name: IBM Automatic Binary Optimizer for z/OS
  Version: 21
  Mod Level: 00
  Optimized Date (YYYYDDD): 2019256
```

最適化された命令

このセクションは、リスト変換のデータの大多数を構成しています。これは、さまざまな IBM COBOL コンパイラ (IBM Enterprise COBOL V5 および V6) によって生成されるリストのオブジェクト・コード・セクションに似ています。最適化された各 CSECT は、PROC 疑似命令コード、およびそのオペランドとなる CSECT の名前で始まります。

例:

(1)	(2)	(3)	(4)	(5)
000258		000000	PROC	PROGA
000258	183F	000000	LR	R3,R15
00025A	5800 3008	000000	L	R0, 8(,R3)
00025E	1E01	000000	ALR	R0,R1
000260	5500 C00C	000000	CL	R0, 12(,R12)
000264	0DF0	000000	BASR	R15,R0
000266	47D0 F00C	000000	BC	R13, 12(,R15)
00026A	58F0 C300	000000	L	R15, 768(,R12)
00026E	0DEF	000000	BASR	R14,R15
000270	181F	000000	LR	R1,R15
000272	50D0 1004	000000	ST	R13, 4(,R1)
000276	5000 104C	000000	ST	R0, 76(,R1)
00027A	D203 1000 3058	000000	MVC	0(4,R1), 88(R3)
000280	D703 1084 1084	000000	XC	132(4,R1), 132(R1)
000286	5090 105C	000000	ST	R9, 92(,R1)
00028A	18D1	000000	LR	R13,R1
00028C	41A0 D120	000000	LA	R10, 288(,R13)

この例は、PROGA という名前の CSECT を対象に作成された、最適化された命令を示しています。最適化された命令は次の 5 つのセクションで構成されています。

1. 最適化された命令の CSECT における 16 進オフセット
2. 命令バイトの 16 進表記
3. 最適化された命令が生成されたときの対象であった「ソース」命令の 16 進 CSECT オフセット
4. 命令の命令コード
5. 命令のオペランド

最適化された命令の間に散在しているのは、それらの最適化された命令の生成対象である「ソース」命令です。列 5 から始まっている行は最適化された命令で、列 1 から始まっている行は「ソース」命令です。下の例では、列 1 から始まる先頭 2 行は PACK ソース命令と OI ソース命令 (それぞれ 16 進オフセット 00042C および 16 進オフセット 000432) です。列 5 から始まる 3 行目は、ABO によって生成された命令「CDZT」です。CDZT の「ソース」16 進オフセットは 00042C で、これは、入力モジュールにおける PACK 命令を対象に生成されたことを示しています。

例:

00042C	PACK	272(4,13),0(7,8)
000432	OI	276(13),15

0004C4	ED07	4000 00AA	00042C	CDZT	FP0, _WSA[0x12c] 0(8,R4),0x0
000436	PACK	280(4,13),8(7,8)			
00043C	OI	284(13),15			
0004CA	ED07	4008 10AA	000436	CDZT	FP1, _WSA[0x12c] 8(8,R4),0x0
000440	AP	272(4,13),280(4,13)			
000446	UNPK	16(7,8),272(4,13)			
0004D0	B3D2	1000	000440	ADTR	FP0,FP0,FP1
00044C	OI	23(8),240			
000450	L	2,248(0,,13)			
0004D4	ED07	4010 00A8	00044C	CZDT	FP0, 16(8,R4),0x0

リテラル・プール

ABO は、新しいリテラルを作成する場合、それをコード・セクションの末尾に置きます。このようなセクションは、リスト変換において定数データ・スニペット (*Constant Data Snippet*) と呼ばれます。定数データ・スニペットは、リスト内に存在しない場合もあれば、複数存在する場合があります。それらの内容は、オリジナル・コンパイラーによって作成されるリテラル・プールによく似ています。オリジナル・リテラル・プールは変更されないままとなり、入力バイナリーの場合と同じく、引き続き使用されます。

例:

(1)	(2)	(3)	(4)
		L0032:	# Constant Data Snippet
000550	4040 4040 4040 4040	DC	X'4040404040404040'
000558	4040 4040 4040 4040	DC	X'4040404040404040'
000560	4040 4040 4040 4040	DC	X'4040404040404040'
000568	4040 4040 4040 0000	DC	X'4040404040400000'
000570	8000 0000 0000 0000	DC	X'8000000000000000'
000578	0000 0000 0000 0000	DC	X'0000000000000000'
000580	0000 0001 0000 0000	DC	X'0000000100000000'
000588	C9C7 E9E2 D9E3 C3C4	DC	X'C9C7E9E2D9E3C3C4'
000590	E2E8 E2D6 E4E3 4040	DC	X'E2E8E2D6E4E34040'
000598	0E00 0000 0000 0000	DC	X'0E00000000000000'

データは次の 4 つのセクションで構成されています。

1. CSECT の先頭からリテラル・プール内のそれらのバイトまでの 16 進オフセット
2. リテラル・プール内のバイトの 16 進表記
3. リテラル・プールの先頭を示すラベル
4. これらのバイトのアセンブラー構文

追加割り振り DSA/TGT バイトのセクション

このセクションには、オプティマイザーによって割り振られたすべての追加 DSA/TGT バイトのバイト数が 16 進数で表示されます。

例:

DSA WILL BE ALLOCATED FOR AN ADDITIONAL 000001A8 BYTES
TGT WILL BE ALLOCATED FOR AN ADDITIONAL 00000000 BYTES

PPA4 セクション

PPA4 セクションには、プログラム・モジュールの最適化に関する情報があります。例えば、最適化の日時や、コード・セクションの長さなどの情報が含まれています。

例:

(1)	(2)	(3)	(4)
	PPA4: Entry Point Constants		
046668	00000000	=X'00000000'	Flags 1
04666C	00000300	=X'00000300'	Flags 2
046670	F2F0F1F6	=C'2019'	Compiled Year
046674	F0F8F2F1	=C'0913'	Compiled Date MMDD

046678	F1F2F3F0F2F3	=C'123023'	Compiled Time HHMMSS
04667E	F0F1F0F2F0F0	=C'010200'	Compiler Version
046684	0004706A	=F'290922'	Code Length
046688	0B020000	=X'0B020000'	Options
04668C	00000028	=X'00000028'	A(PPA4-ListName)
PPA4 End			

PPA4 は次の 4 つのセクションで構成されています。

1. CSECT of PPA4 セクション項目の CSECT におけるオフセット (16 進)
2. PPA4 セクション内のバイトの 16 進表記
3. PPA4 内のバイトのアセンブラー構文
4. PPA4 内のデータの説明

自動マップ

自動マップには、ABO によって作成されたシンボルのオフセットおよびサイズ (16 進数) が含まれています。これらのオフセットは、オリジナル DSA の末尾に確立されたベースに相対しています。自動マップは、オリジナル・プログラムにおける自動化も、オリジナル・コンパイラによって作成されるテンポラリーも表示しません。ABO は、「新しい」DSA の開始オフセットを入れるための汎用レジスター (GPR) を確立します。新しく作成された自動化はすべて、そのベースとして新しいレジスターで参照されます。

例:

(1)	(2)	(3)
	*****	A U T O M A T I C M A P *****
OFFSET (HEX)	LENGTH (HEX)	NAME
0	4	_GPR0
4	4	_GPR1
8	4	_GPR2
C	4	_GPR3
10	4	_GPR4
14	4	_GPR5
18	4	_GPR6
1C	4	_GPR7
20	4	_GPR8
24	4	_GPR9
28	4	_GPR10
2C	4	_GPR11
30	4	_GPR12
34	4	_GPR13

自動マップは次の 3 つのセクションで構成されています。

1. 新しいスタックの先頭に相対するスタック・シンボルの 16 進オフセット
2. シンボルの長さ (バイト) の 16 進値
3. シンボルの名前

入力命令

このセクションには、入力モジュールにある命令の完全なリストが含まれています。最適化された CSECT ごとにセクションが 1 つあります。これらの入力命令は、『最適化された命令』で既に表示されている入力命令と同じです。そのセクションでは、入力命令が、対応する最適化済み命令と混在した状態で示されていて、それ自体は完全な番号付きリストになっていません。

入力命令セクションの先頭には、コンパイルに使用された COBOL コンパイラのバージョンと、CSECT から抽出された「Signature information bytes」が記載されています。「Signature information bytes」は COBOL プログラミング・ガイドで説明されていて、コンパイル済みプログラムに関する情報を示します。これらの情報バイトはデコードされ、対応する使用済みコンパイラ・オプションが印刷されます。デコードされたコンパイラ・オプションは、内容やフォーマットにおいて、オリジナルのコンパイラ・リストに表示されているものと正確には一致しない可能性があります。これは、ABO が、オリジナルのソース全体や、オリジナルのコンパイル処理中に指定されたオプションではなく、入力 CSECT に含まれる「Signature information bytes」のみに基づいてオプションをデコードするためです。

次の例では、PROGB という COBOL CSECT が最適化されました。

```

(1)      (2)      (3)

*****  INPUT  INSTRUCTIONS  *****

id 5655S7100  v42 m00 2018005 resident PROGB
Enterprise COBOL V4: start=0x10, length=9.15 (kBytes)
Signature information bytes:
a0487dcc 20000000 00880108 00000040
08000000 00000008 00f00114 00
Compiler Options in effect:
    ADV
    ARITH(COMPAT)
    NOAWO
    NOBLOCK0
    コンパイル単位はプログラム
    NOCICS
    NOCURRENCY
    Default DDNAME for OUTDD will be used
    DATA(31)
    NODATEPROC
    DBCS
    NODECK
    NODLL
    NODUMP
    NODYNAM
    NOEXPORTALL
    NOFASTSRT
    INTDATE(ANSI)
    NOLIB
    LIST
    NOMAP
    NOMDECK
    NONAME
    NONUM
    NUMCLS(PRIM)
    NUMPROC(NOPFD)
    OBJ
    NOOFFSET
    NOOPTIMIZE
    QUOTE
    RENT
    RMODE(ANY)
    SEQUENCE
    SIZE(MAX)
    SOURCE
    NOSQL
    SQLCCSID
    NOSSRANGE
    TERM
    TEST(HOOK,NOSEPARATE,EJPD)
    NOTHREAD
    TRUNC(STD)
    NOVBREF
    NOWORD
    XMLPARSE(XMLSS)
    XREF
    YEARWINDOW(1900)
    ZWB

0002BA  LR      3,15
0002BC  LA      0,440(0,,1)
0002C0  CL      0,12(0,,12)
0002C4  BASR    15,0
0002C6  BC      13,12(0,15)
0002CA  L       15,768(0,,12)
0002CE  BASR    14,15
0002D0  LR      1,15
0002D2  ST      13,4(0,,1)
0002D6  ST      0,76(0,,1)
0002DA  MVC     0(4,1),88(3)
0002E0  XC      132(4,1),132(1)
0002E6  ST      9,92(0,,1)
0002EA  LR      13,1
0002EC  L       12,232(0,,9)
0002F0  LR      1,2
0002F2  ST      13,88(0,,13)
0002F6  L       10,36(0,,12)
0002FA  L       8,300(0,,9)

```

0002FE	STM	14,12,12(13)
000302	MVC	136(4,13),16(10)
000308	LA	2,280(0,,13)
00030C	ST	2,140(0,,13)
000310	ICM	2,15,364(9)
000314	L	11,40(0,,12)
000318	BC	8,214(0,11)
00031C	L	3,92(0,,9)
000320	L	15,244(0,,3)
000324	LA	1,271(0,,10)
000328	BASR	14,15
00032A	A	2,0(0,,12)
00032E	ST	2,364(0,,9)
000332	MVC	256(4,13),252(13)
000338	LA	2,240(0,,11)
00033C	ST	2,252(0,,13)
000340	BC	15,84(0,11)
000344	MVC	252(4,13),256(13)
00034A	MVC	260(4,13),248(13)
000350	L	11,40(0,,12)
000354	LA	2,268(0,,11)
000358	ST	2,248(0,,13)
00035C	BC	15,0(0,11)
000360	MVC	248(4,13),260(13)
000366	TM	87(9),64
00036A	L	11,40(0,,12)
00036E	BC	1,306(0,11)
000372	TM	84(9),32
000376	BC	14,298(0,11)
00037A	OI	132(13),32
00037E	OI	87(9),64
000382	BC	15,306(0,11)
000386	OI	348(9),64
00038A	OI	132(13),1
00038E	BAS	14,316(0,,9)
000392	BAS	14,318(0,,9)
000396	AP	0(8,8),16(8,8)
00039C	ZAP	0(8,8),0(8,8)
0003A2	BAS	14,318(0,,9)
0003A6	ZAP	16(8,8),0(8,8)
0003AC	BAS	14,318(0,,9)
0003B0	CP	0(8,8),114(2,10)
0003B6	L	11,40(0,,12)
0003BA	BC	7,392(0,11)
0003BE	BAS	14,316(0,,9)
0003C2	BAS	14,318(0,,9)
0003C6	L	2,92(0,,9)
0003CA	L	15,44(0,,2)
0003CE	LA	1,259(0,,10)
0003D2	BASR	14,15
0003D4	L	11,40(0,,12)
0003D8	BC	15,414(0,11)
0003DC	BAS	14,316(0,,9)
0003E0	BAS	14,318(0,,9)
0003E4	L	2,92(0,,9)
0003E8	L	15,44(0,,2)
0003EC	LA	1,247(0,,10)
0003F0	BASR	14,15
0003F2	BAS	14,316(0,,9)
0003F6	BAS	14,318(0,,9)
0003FA	CP	16(8,8),114(2,10)
000400	L	11,40(0,,12)
000404	BC	7,466(0,11)
000408	BAS	14,316(0,,9)
00040C	BAS	14,318(0,,9)
000410	L	2,92(0,,9)
000414	L	15,44(0,,2)
000418	LA	1,235(0,,10)
00041C	BASR	14,15
00041E	L	11,40(0,,12)
000422	BC	15,488(0,11)
000426	BAS	14,316(0,,9)
00042A	BAS	14,318(0,,9)
00042E	L	2,92(0,,9)
000432	L	15,44(0,,2)
000436	LA	1,223(0,,10)
00043A	BASR	14,15
00043C	BAS	14,316(0,,9)
000440	BAS	14,318(0,,9)
000444	BAS	14,316(0,,9)
000448	L	11,40(0,,12)
00044C	BC	15,534(0,11)

000450	BAS	14,316(0,,9)
000454	TM	132(13),32
000458	BC	14,534(0,11)
00045C	L	2,92(0,,9)
000460	L	15,244(0,,2)
000464	LA	1,205(0,,10)
000468	BASR	14,15
00046A	L	2,364(0,,9)
00046E	S	2,0(0,,12)
000472	ST	2,364(0,,9)
000476	NI	348(9),191
00047A	TM	85(9),64
00047E	BC	14,572(0,11)
000482	LA	1,8(0,,0)
000486	L	2,92(0,,9)
00048A	L	15,32(0,,2)
00048E	BASR	14,15
000490	TM	84(9),40
000494	BC	7,602(0,11)
000498	L	2,296(0,,9)
00049C	LH	15,8(0,,2)
0004A0	L	13,4(0,,13)
0004A4	L	14,12(0,,13)
0004A8	LM	0,12,20(13)
0004AC	BCR	-1,14
0004AE	MVC	264(12,13),177(10)
0004B4	L	3,296(0,,9)
0004B8	LH	2,8(0,,3)
0004BC	ST	2,276(0,,13)
0004C0	LA	1,264(0,,13)
0004C4	L	2,92(0,,9)

***** END OF INPUT INSTRUCTIONS *****

入力命令セクションに含まれる 3 つのセクションは次のような内容になっています。

1. オリジナル命令の入力 CSECT における 16 進オフセット
2. 命令ニーモニック
3. 命令のオペランド

SYSPRINT DD および LIST オプション

SYSPRINT DD または LIST オプションを使用して、生成されるリスト変換のロケーションを指定します。

SYSPRINT または LIST のターゲットは、以下の項目のいずれかです。

- 順次データ・セット、または PDSE (PDS ではない) のメンバー。複数の CSECT 最適化の出力は、この順次データ・セットに、最適化の順に追加されます。
- PDS または PDSE。CSECT が最適化されると、その CSECT に特定のリスト変換が PDS または PDSE のメンバーに置かれます。メンバー名は CSECT 名に基づいています (大文字変換され、8 文字に切り捨てられます)。メンバーに内容があれば、前の内容が ABO によって以前の起動で作成されたものであっても、その内容は上書きされます。
- HFS パス。複数の CSECT 最適化の出力は、この HFS ファイルに追加されます。

LIST オプションは、SYSPRINT DD よりも優先されます。LIST オプションを指定すると、このオプションは SYSPRINT DD をオーバーライドします。LIST オプションが指定されていれば、SYSPRINT DDname を省略できます。

注：

ABO リストには、詳細な変換情報が含まれています。そのため、ABO リスト非常に大きくなる可能性があります。SYSPRINT DD ターゲットを SYSOUT として指定すると、JES2 スプールがシステム指定の行制限に達する可能性があります。スプール行限度に達すると、JES2 はインストール・システム出口ルーチンに制御を渡しますが、ABO ジョブは終了することも終了しないこともあります。JOBPARM L オプションを使用してスプール行制限を緩和することは可能ですが、それでも最大 L 設定値 999999 は ABO リストにとって十分な大きさではない可能性があります。

この問題を回避するには、このセクションで説明されているように、SYSPRINT で PDS、PDSE、または HFS の位置が指定されるようすることをお勧めします。

例

次の JCL の例では、SYSPRINT DD に PDSE が使用されていて、リスト変換が PDSE のメンバーに書き込まれるようになっています。

```
//SYSIN DD *  
BOPT IN=HLQ.IN.LOAD(MOD*) OUT=HLQ.OUT.LOAD  
...  
//SYSPRINT DD DSN=HLQ.LIST.PDSE,DISP=SHR
```

この例では、入力プログラム・モジュールが **HLQ.IN.LOAD(MOD*)** として指定されています。つまり、名前が「MOD」で始まる **HLQ.IN.LOAD** 内の適格なメンバーがすべて最適化されます。

入力データ・セットには、2つのメンバー **MOD1** および **MOD2** があります。これらの2つのプログラム・モジュールの中には、さまざまな CSECT が入っています。

表 8. 入力モジュールと、それらに含まれている CSECT	
HLQ.IN.LOAD	CSECT
MOD1	PROG1A
	PROG1B
	PROG1C
MOD2	PROG2A
	PROG2B

ABO は、これらの CSECT を 1 つずつ最適化して、CSECT ごとに以下の 2 つの出力を作成します。

1. 最適化された CSECT
2. CSECT 用のリスト変換

最適化された CSECT は入力 CSECT と同じ名前になります。また、最適化された CSECT は、入力プログラム・モジュールと同じメンバー名を持つプログラム・モジュール内に配置されます。ただし、新しいプログラム・モジュールは、「**HLQ.OUT.LOAD**」という名前の新しい PDSE に置かれます。

表 9. 出力 1: 最適化されたモジュールと、それらの CSECT	
HLQ.OUT.LOAD	CSECT
MOD1	PROG1A
	PROG1B
	PROG1C
MOD2	PROG2A
	PROG2B

最適化された CSECT ごとに生成されるリストは、個別のメンバーとして PDSE 「**HLQ.LIST.PDSE**」に置かれます。このような PDSE メンバーはそれぞれ、入力 CSECT と同じ名前を持ちます。結果として、**HLQ.LIST.PDSE** には 5 つのメンバー **PROG1A**、**PROG1B**、**PROG1C**、**PROG2A**、および **PROG2B** が入ります。

表 10. 出力 2: リスト変換	
HLQ.LIST.PDSE	
PROG1A	

表 10. 出力 2: リスト変換 (続き)
HLQ.LIST.PDSE
PROG1B
PROG1C
PROG2A
PROG2B

第7章 最適化の管理、および最適化されたモジュール・デプロイメント・プロセスの管理

ABO 使用時に段階的反復アプローチを使用

ABO を使用するための段階的反復アプローチは、ABO によって生成されて最適化されたプログラムを実行することで得られるメリットに合わせて最適化プロセスのコストのバランスを取るための推奨アプローチです。

例えば、最初に、上位 x% のコントリビューターを含むモジュールを CPU 時間に最適化します。ABO によって生成されたモジュールを使用して影響 (例えば、CPU 時間の短縮) を測定し、パフォーマンス目標が満たされるまで次の上位 x% の CPU コントリビューターについて操作を繰り返します。

IBM Application Performance Analyzer (APA) for z/OS などのパフォーマンス測定/レポート作成ツールは、上位 CPU コントリビューターの判別に役立つ可能性があります。使用可能なパフォーマンス測定ツールを所有していない場合、アプリケーション実行時に最も実行された COBOL モジュールを判別するのであれば、ABO に付属する [RTI Profiler](#) を使用できます。

ABO からの恩恵を最も受けるプログラムの特性

コンパイルされたプログラムのいくつかは、他のプログラムよりも多くの恩恵を ABO から受けます。これらのプログラムの重要な特性をいくつか知っておくことは、コンパイルした COBOL プログラムで ABO を段階的に使用するときにも役立つ可能性があります。

ABO は、コンパイラが生成したオリジナルのコード、および一部の Language Environment (LE) 選択ルーチンに関してのみ、パフォーマンスを向上させることができます。ただし、ABO には、他のサブシステム (CICS、Db2、IMS など) で時間が費やされた場合にパフォーマンスを向上させる手段がありません。

ABO を使用した最適化からより多くの恩恵を受ける可能性があるプログラムの重要な特性:

- アプリケーションの実行時間の大部分が COBOL コードで費やされます (CICS、Db2、IMS など、他のサブシステムで費やされるものではありません)。
- COBOL コードで非常に多くの計算が行われます。例えば、COBOL コード自体で実際の作業が行われていて、COBOL コード自体が単に他のプログラムやサブシステムの「ドライバー」プログラムとして機能しているのではないプログラムなどです。
 - ソース・レベルでは、恩恵を受ける可能性が最も高いステートメントとしては COMPUTE、IF、MOVE、ADD、SUBTRACT、MULTIPLY、DIVIDE、REMAINDER などがありますが、これらに限られるわけではありません。
 - また、一部の Language Environment (LE) ルーチンも ABO によって最適化できます。そのようなルーチンは、多種多様な変換、移動、および算術演算を行います。このようなルーチンには、IGZCSH2、IGZCFPC、IGZCONV、IGZCVMO、IGZCXPR、IGZCXMU、IGZCXDI などがあります。ABO はこのようなルーチンを最適化します。そのために、ABO は、そのようなルーチンの作業をより効率的に、最適化されたコードで直接実行したり、より効率的な LE ルーチンと呼び出したりします。

注: COBOL ソースを調べるだけでは、実際にアプリケーションで費やされる時間は分からないため、APA などの分析ツールで作成されたパフォーマンス・レポートも一緒に調べる必要があります。

- アプリケーション内に含まれる多くの COBOL モジュールが ABO による最適化に適格です。すなわち、そのようなモジュールは適格な COBOL コンパイラでコンパイルされたものであり、ABO によってサポートされている言語機能を含んでいます。

最適化およびデプロイメントの使用法のシナリオ

このセクションには、IBM Automatic Binary Optimizer for z/OS の使用法についての標準的なシナリオが 3 つ含まれています。これらのシナリオでは、IBM Automatic Binary Optimizer for z/OS を使用して、既にコンパイルされている IBM COBOL プログラムのパフォーマンスを向上させるために取り得ることができる方法について説明します。それぞれのシナリオにはステップバイステップの説明があり、コンパイル済みの IBM COBOL プログラムを最適化できるようになっています。

シナリオ 1: 静的デプロイメントによる最適化プロセス

この使用法についてのシナリオでは、BOPT ディレクティブを使用してオブティマイザーへの入力モジュールを JCL に指定し、オリジナル・モジュールが含まれているデータ・セットを識別する既存の JCL をすべてデプロイメントのために更新します。

手順

静的デプロイメントによって最適化を実行するには、以下の手順に従ってください。

1. 新規データ・セットを作成します。例えば、このシナリオ用には以下のデータ・セットが作成されます。*HLQ* は、定義する高位修飾子です。
 - *HLQ.OUT.LOAD.Z14*: このデータ・セットには、z14 マシンをターゲットにして最適化されたバイナリーが取り込まれます。
 - *HLQ.OUT.LOAD.Z15*: このデータ・セットには、z15 マシンをターゲットにして最適化されたバイナリーが取り込まれます。
2. ABO を実行して、新規データ・セットに取り込みます。オブティマイザーを実行するには、新しい JCL を作成してください。SYSIN で始まるインストリーム行で、BOPT オブティマイザー・ディレクティブを使用します。IN オプションで、最適化するコンパイル済み COBOL プログラムを選択します。例えば次の JCL は、*HLQ.IN.LOAD* にある、M で始まる名前のメンバーすべてを最適化するように、オブティマイザーに指示します。z14 および z15 をそれぞれターゲットにして最適化されたバイナリーは、個別に *HLQ.OUT.LOAD.Z14* および *HLQ.OUT.LOAD.Z15* に置かれます。

```
//SYSIN DD *  
BOPT IN=HLQ.IN.LOAD(M*) OUT=HLQ.OUT.LOAD.Z14 LIST=HLQ.OUT.LIST.Z14 ARCH=12  
BOPT IN=HLQ.IN.LOAD(M*) OUT=HLQ.OUT.LOAD.Z15 LIST=HLQ.OUT.LIST.Z15 ARCH=13
```

この例は、ユーザーが SYSIN ファイルに何を指定しなければならないかを示すことを目的にしています。基本 JCL 構成については、65 ページの『付録 A JCL サンプル』を参照してください。静的デプロイメント・シナリオで使用できる、これ以外のサンプル JCL については、28 ページの『BOPT で最適化を指定』を参照してください。

3. 最適化されたプログラムを実行するには、オリジナル・プログラムを実行するために使用される JCL を変更します。その JCL は、オリジナル・モジュールが入っているデータ・セットを識別しています。STEPLIB 設定では、ターゲット・アーキテクチャーごとに、最適化されたモジュールのデータ・セットをオリジナル・モジュールのデータ・セットの前に置く必要があります。以下のスニペットは、最適化されたプログラム・バイナリーとオリジナルのプログラム・バイナリーを指し示す JCL 内の変更部分を示しています。

z14 でオリジナル・プログラムを実行するために使用される JCL 内の変更部分:

```
//STEPLIB DD DSN=HLQ.OUT.LOAD.Z14,DISP=SHR  
// DD DSN=HLQ.IN.LOAD,DISP=SHR  
...
```

z15 でオリジナル・プログラムを実行するために使用される JCL 内の変更部分:

```
//STEPLIB DD DSN=HLQ.OUT.LOAD.Z15,DISP=SHR  
// DD DSN=HLQ.IN.LOAD,DISP=SHR  
...
```

シナリオ 2: 動的デプロイメントによる最適化プロセス

この使用法についてのシナリオでは、IEFOPZxx SYS1.PARMLIB のメンバー内の出力モジュールに入力モジュールをマッピングしてから、IEFOPZ オプティマイザー・ディレクティブを使用して、入力モジュールの最適化を出力モジュールに指定します。バイナリー最適化が完了した後、オリジナル・プログラムを実行するために使用された既存の JCL に変更を加えずに、最適化されたプログラムを実行します。

このタスクについて

IEFOPZxx には、古い COBOL ライブラリーと目的の新しいライブラリーとの組み合わせのリスト (目的のアーキテクチャー・レベルごとに 1 つずつ) を提供する、データ・セット最適化構成を定義するステートメントが入っています。また IEFOPZxx は、処理 (最適化) する対象のメンバーを指定します。詳しくは、「z/OS 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。

手順

動的デプロイメントによってバイナリー最適化を実行するには、以下の手順に従ってください。

1. 新規データ・セットを作成します。例えば、このシナリオ用には以下のデータ・セットが作成されます。*HLQ* は、定義する高位修飾子です。
 - *HLQ.OUT.LOAD.Z14*: このデータ・セットには、z14 マシンをターゲットにして最適化されたバイナリーが取り込まれます。
 - *HLQ.OUT.LOAD.Z15*: このデータ・セットには、z15 マシンをターゲットにして最適化されたバイナリーが取り込まれます。
2. IEFOPZ 構成を定義します。
 - a. IEFOPZxx メンバーを作成します。
 - b. 最適化するコンパイル済みモジュールが入っている古いデータ・セットごとに、OLD/NEW ペアを IEFOPZxx メンバーに定義します。OLD/NEW ペアを INACTIVE とマークして、予期せずにシステムが OLDNEW 処理を実行しないようにします。次の例を参照してください。

```
MAXARCH(13)
CHECKALL
OWNER(IBM) MINARCH(12)
OLDNEW(
  OWNER(IBM)
  OLD (DSN(HLQ.IN.LOAD))
  NEW (DSN(HLQ.OUT.LOAD.Z14)  ARCH(12))
  NEW (DSN(HLQ.OUT.LOAD.Z15)  ARCH(13))
  INCLUDEMEMBERS(M*) //Identifies to process all members beginning with M
  INACTIVE
)
```

注: OLD/NEW ペアは、1 つ以上の IEFOPZxx メンバーに定義できます。

3. IEFOPZ 構成をアクティブにするには、次の z/OS MVS System コマンドを実行してください。

```
SET IEFOPZ=( $x_1, \dots, x_n$ )
```

ここで $x_1, \dots, x_n$ は、IEFOPZxx メンバーの接尾部 xx です。前のステップでメンバーを 1 つだけ作成した場合、コマンドは次のようになります。

```
SET IEFOPZ= $x_1$ 
```

注: SET コマンドの変更は現行 IPL セッションでのみ保持されます。そのため、通常、この変更は新規構成即時テストに使用されたり、現行 IPL セッションでいくつかの永続定義をオーバーライドするために使用されます。永続構成定義については、ステップ 5 を参照してください。

4. IBM Automatic Binary Optimizer for z/OS を実行して、新規データ・セットに取り込みます。

オプティマイザーを実行するには、JCL を以下のように作成してください。SYSIN で始まるインストリーム・データで、IEFOPZ ディレクティブを使用します。

```
//*SYSIN DD *
```

```
IEFOPZ SEL_ARCH=12 LIST=HLQ.B0ZOPT.ARCH12.LIST
IEFOPZ SEL_ARCH=13 LIST=HLQ.B0ZOPT.ARCH13.LIST
```

この例は、ユーザーが SYSIN ファイルに何を指定しなければならないかを示すことを目的にしています。基本 JCL 構成については、65 ページの『付録 A JCL サンプル』を参照してください。動的デプロイメント・シナリオで使用できる、これ以外のサンプル JCL については、31 ページの『IEFOPZ で最適化を指定』を参照してください。

5. それ以降の IPL で、目的の IEFOPZ 構成が正しくアクティブになるよう、ご使用の IEASYSxx SYS1.PARMLIB メンバーを IEFOPZ システム・パラメーターで更新します。例えば、メンバー IEFOPZ99 で指定された IEFOPZ 構成を後続の各 IPL で自動的に活動化するには、IEFOPZ=99 ステートメントを IEASYSxx メンバーに入れます。ただし、ステップ 3 で説明されている SET コマンドは、例えば SET IEFOPZ=99 のように発行されているのであれば、現行 IPL セッションでのみ、必要な IEFOPZ99 メンバーをアクティブにします。
6. OLD/NEW ペアを ACTIVE として再定義します。OLD/NEW 処理を JOBLIB および STEPLIB 以外の DDNAME に対して実行する場合は、DDNAME ステートメントを使用して、それらの DDNAME を IEFOPZxx parmlib メンバー内に定義します。その後で、更新済み IEFOPZ 構成をアクティブにしてください。
7. オリジナル・プログラムを実行するために使用された既存の JCL を使用して、最適化されたプログラムを実行します。

関連資料

87 ページの『関連資料』

シナリオ 3: ハイブリッド・アプローチによる最適化プロセス

ハイブリッド・アプローチでは、シナリオ 1 で行ったように、最適化する入力バイナリーを明示的に JCL に指定しますが、さらに、シナリオ 2 で説明した動的デプロイメントを結合します。動的デプロイメントによって、既存の JCL を変更することなく、最適化されたモジュールを実行します。

手順

ハイブリッド・アプローチを使用してバイナリー最適化を実行するには、以下の手順に従ってください。

1. 新規データ・セットを作成します。例えば、このシナリオ用には以下のデータ・セットが作成されます。*HLQ* は、定義する高位修飾子です。
 - *HLQ.OUT.LOAD.Z14*: このデータ・セットには、z14 マシンをターゲットにして最適化されたバイナリーが取り込まれます。
 - *HLQ.OUT.LOAD.Z15*: このデータ・セットには、z15 マシンをターゲットにして最適化されたバイナリーが取り込まれます。
2. IBM Automatic Binary Optimizer for z/OS を実行して、新規データ・セットに取り込みます。オブティマイザーを実行するには、新しい JCL を作成してください。SYSIN で始まるインストリーム・データで、BOPT オプティマイザー・ディレクティブを使用して、最適化するコンパイル済み COBOL モジュールを選択します。例えば次の JCL は、*HLQ.IN.LOAD* にある、文字 M で始まるメンバーすべてを最適化するよう、オブティマイザーに指示します。z14 および z15 をそれぞれターゲットにして最適化されたバイナリーは、個別に *HLQ.OUT.LOAD.Z14* および *HLQ.OUT.LOAD.Z15* に置かれます。

```
//SYSIN DD *
BOPT IN=HLQ.IN.LOAD(M*) OUT=HLQ.OUT.LOAD.Z14 ARCH=12
BOPT IN=HLQ.IN.LOAD(M*) OUT=HLQ.OUT.LOAD.Z15 ARCH=13
```

この例は、ユーザーが SYSIN ファイルに何を指定しなければならないかを示すことを目的にしています。基本 JCL 構成については、65 ページの『付録 A JCL サンプル』を参照してください。ハイブリッド・シナリオで使用できる、これ以外のサンプル JCL については、28 ページの『BOPT で最適化を指定』を参照してください。

3. IEFOPZ 構成を定義します。
 - a. IEFOPZxx メンバーを作成します。

- b. 最適化するコンパイル済みモジュールが入っている古いデータ・セットごとに、OLD/NEW ペアを IEFOPZxx メンバーに定義します。OLD/NEW ペアを ACTIVE とマークします。以下の例を参照してください。

```
OLDNEW (
  OLD( DSNNAME (HLQ.IN.LOAD) )
  NEW( DSNNAME (HLQ.OUT.LOAD.Z14) ARCH(12) )
  NEW( DSNNAME (HLQ.OUT.LOAD.Z15) ARCH(13) )
  INCLUDEMEMBERS(M*) //Identifies to process all members beginning with M
  ACTIVE )
```

注: OLD/NEW ペアは、1 つ以上の IEFOPZxx メンバーに定義できます。

- c. OLDNEW 処理を JOBLIB および STEPLIB 以外の DDNAME に対して実行する場合は、DDNAME ステートメントを使用して、それらの DDNAME を IEFOPZxx parmlib メンバー内に定義します。その後で、更新済み IEFOPZ 構成を次のコマンドでアクティブにしてください。

```
SET IEFOPZ=(x1, ..., xn)
```

ここで $x_1, \dots, x_n$ は、IEFOPZxx メンバーの接尾部 xx です。前のステップでメンバーを 1 つだけ作成した場合、コマンドは次のようになります。

```
SET IEFOPZ=x1
```

4. オリジナル・プログラムを実行するために使用された既存の JCL を使用して、最適化されたプログラムを実行します。

関連資料

87 ページの『関連資料』

テスト情報

ABO が生成する最適化済みモジュールは、オリジナルの COBOL モジュールに比べて高速に実行されますが、動作は同じです。ただし、エラー・メッセージおよび異常終了コードが異なるものもまれにあります。ABO がこれを行うことができるのは、ABO が COBOL モジュール内のバイナリー・コードを処理するためプログラムの低レベル・ロジックを同じままにしておくことができるからです。つまり、ABO のユーザーは、ABO 最適化済みモジュールの完全な機能検査を行う必要がありません。ABO 最適化済みモジュールを実稼働環境にデプロイする前に ABO 最適化済みモジュールを使用してアプリケーションの基本機能を保証するには、限られた特定のテストが推奨されます。

パフォーマンス・テストは、オリジナル・アプリケーションと一緒に使用されたり、ABO 最適化済みモジュールを含むアプリケーションと一緒に使用されたりする同じ入力データを使用して制御環境で実行されなければなりません。他の実行中アプリケーションが可能な限り少ないマシンや LPAR を使用すれば、安定した再現可能なパフォーマンス結果を得ることができます。オリジナルのアプリケーションと最適化済みアプリケーションとの間で CPU 時間を比較することは、パフォーマンス向上を確認する最良の方法です。

第 8 章 最適化に関する問題と、最適化済みモジュールのデプロイメントに関する問題を解決

最適化時に発生した問題を解決

ABO が z/OS に渡す戻りコードは、最適化時に問題が発生したかどうかを示す標識です。戻りコード値がゼロの場合、最適化は正常に行われ、問題は発生しなかったことになります。戻りコード値がゼロ以外の場合、何か予期しなかったことが発生したか、または問題が発生したことになります。戻りコードの詳細については、67 ページの『付録 B 戻りコード』を参照してください。

ABO は、問題の診断に使用できる出力ファイルを作成します。

COBOL プログラム最適化の間に見つかった問題を診断するために、以下のファイルが役立ちます。

- ログ・ファイルに、何が最適化またはスキャンされたかに関する要約と、該当する場合は、エラー・メッセージが書き込まれます。詳しくは、[メッセージ](#)を参照してください。最適化プロセスの間に問題が検出された場合、まず最初に調べるべき対象はログ・ファイルです。
- 例外的な状況では、メッセージは、OPTERR DD によって指定されたファイルに書き込まれます。OPTERR DD が指定されていない場合、メッセージは JOBLLOG に書き込まれます。
- ABO 実行時のプログラム例外などの状況では、メッセージは、CEEDUMP DD によって指定されたファイルに書き込まれます。CEEDUMP ファイルは Language Environment (LE) によって作成されます。ここには、異常終了の時点で実行されていたプロシーチャーのトレースバックといった情報が入ります。
- JOBLLOG には、他の場所に書き込まれたエラー・メッセージを補完する追加診断メッセージが含まれます。または、JOBLLOG を、例外的な状況で発生したエラーのデフォルト・ロケーションにすることができます。

実行時に発生した問題を解決

一般的な実行エラーの解決方法

インストールされている「Language Environment Automatic Binary Optimizer Runtime Engine」PTF が入手可能な最新 PTF ではない場合、ABO 最適化済みモジュールは U4038 異常終了で失敗する可能性があります。この場合、そのモジュールは次のいずれかのメッセージを出力します。

```
IGZ0153S Program BOZSRC1 was compiled with a level of the compiler that requires service to be
installed on Language Environment.
IGZ0355S Program BOZSRC1 was optimized with a level of the Automatic Binary Optimizer that
requires service to be installed on Language Environment.
```

最初のメッセージは z/OS 2.2 および 2.3 の PTF によって発行されるものです。2 番目のメッセージは z/OS 2.4 の PTF によって発行されるものです。この問題を解決するには、プログラム・ディレクトリーにリストされている最新の「Language Environment Automatic Binary Optimizer Runtime Engine」PTF をインストールする必要があります。また、ABO PTF に関する最新情報について、[フィックス・リストと新規フィーチャーのページ](#)を参照してください。

ABO で生成されたモジュールを、ABO でサポートされていないシステムで実行しようとした場合、OC1 異常終了が発生します。サポート対象のシステムについては、4 ページの『ターゲット・ハードウェア・レベル』を参照してください。

実行エラー診断

Application Delivery Foundation for z Systems において IBM から提供されている問題判別ツールを使用すれば、ABO 最適化済みモジュールが含まれるアプリケーションで発生した実行時間問題の原因を判別できます。使用可能な問題判別ツールがない場合は、ABO から生成されたリスト変換が実行時間問題の判別に役立つ可能性があります。

実行時間問題を引き起こしている ABO 最適化済みモジュールが診断で判明した場合は、オリジナル COBOL モジュールに戻り、IBM サービスに連絡して問題を報告してください。ABO の問題を IBM に報告するために収集する必要がある情報については、[ABO Mustgather](#) のページを参照してください。

最適化後の COBOL モジュール・サイズの変更

通常、最適化済みモジュールのサイズは、ABO がパフォーマンス向上のために行う最適化のタイプが原因で、オリジナルのモジュールよりも大きくなります。

モジュール・サイズが大きくなる一般的な理由をいくつか以下に示します。

- 4 バイトまたは 2 バイトの長さ (多くの低位 ARCH 命令) に対して高位の ARCH 命令 (通常は 6 バイト) が使用されている。次に例を示します。
 - パフォーマンス向上のためにパック/ゾーン 10 進数演算機構に 10 進数浮動小数点 (DFP) が使用されている
 - オリジナルのモジュールの「ベース・ロケーター」ポインターの代わりに、効率は良いがサイズが大きい長変位命令が使用されている
 - 1 つのメモリー内移動の代わりに複数の即値命令が使用されている
- ABO における一部の最適化により、生成されるコードは大きくなるが、パスの長さが短くなり、パフォーマンスが向上する。以下に例を示します。
 - 数値編集変数の処理効率の向上
 - 長さは短くなるが速度は格段に劣る命令の代わりに長移動比較演算をアンロール
 - 2 進データの 10 進精度を条件付きで訂正
- 各種ランタイム・ライブラリー・ルーチンの動作をインライン化したために、最適化済みモジュールにおいてコードが大きくなった。ただし、多くの場合に、パフォーマンスが格段に向上する。

上記の理由、および同様の理由により、多くの場合、ABO によって生成される最適化済みモジュールは、オリジナルのモジュールよりも大きくなり、必要とされるディスク・ストレージ上の領域が増えます。ただし、最適化済みプログラム自体によって実行時に使用されるメモリーの量は、オリジナルのモジュールによって使用されるメモリーの量と同じです。最適化済みプログラムの実行時に観測される EXCP カウントが多少高くなることはありますが、これは単に、より大きなモジュールをロードするためにほんのわずかの追加出力命令が必要になるためです。

最適化済みモジュールは、最適化されたコードがオリジナルのコードより小さくなることがあっても、そのサイズを未変更のまま維持します。

エラー・メッセージおよび異常終了コードの相違

ABO によって生成された最適化済みモジュールは、ほとんどすべてのケースで、対応するオリジナル・モジュールと機能的に同等です。ただし、まれに、ABO によって生成されたモジュールは、オリジナル・モジュールとは異なる Language Environment (LE) ランタイム・メッセージや CICS 異常終了コードを出力することがあります。

このような状況が発生する可能性があるのは、巨大なデータ項目に対する除算、および他の複雑な演算が、LE ライブラリー・ルーチンや非効率的なマシン命令によって処理されるのではなく、より効率的な処理のために ABO によって生成されたコードにおいてインライン化/最適化されたときです。

非 CICS アプリケーションでは、ABO によって生成されたモジュールは以下の処理を行うことがあります。

- オリジナルのモジュールが 10 進数除算例外 (CEE3211S) または IGZ0061S メッセージを生成した場所で固定小数点除算例外 (CEE3209S) メッセージを生成することがあります。
- オリジナルのモジュールが IGZ0061S メッセージを生成した場所で 10 進数除算例外 (CEE3211S) を発生させることがあります。

参考のために、上記の異なる例外を表す LE ランタイム・メッセージの全文を以下に示します。

```
CEE3211S システムで 10 進数除算例外が検出されました (システム完了コード = 0CB)。  
(CEE3211S The system detected a decimal-divide exception (System Completion Code=0CB).)  
IGZ0061S プログラム「program-name」において変位「displacement」でゼロ除算が行われました。  
(IGZ0061S Division by zero occurred in program 'program-name' at displacement 'displacement'.)  
CEE3209S システムで固定小数点除算例外が検出されました (システム完了コード = 0C9)。  
(CEE3209S The system detected a fixed-point divide exception (System Completion Code=0C9).)
```

CICS アプリケーションに関しては、「CICS ASSIGN ABCODE」で返される異常終了コードが「1061」(オリジナル・モジュールのもの)から「ASRA」(ABO によって生成されたモジュールのもの)に変更される可能性があります。

Application Delivery Foundation for z Systems

ABO によって生成されたモジュール上では、Application Delivery Foundation for z Systems (ADFz) を使用できます。

Application Delivery Foundation for z Systems について詳しくは、<https://www.ibm.com/ca-en/marketplace/app-delivery-foundation-on-zsystems> を参照してください。

ABO によって生成された COBOL モジュールでは、以下の Application Delivery Foundation for z Systems ファミリーの問題判別ツールを使用できます。

- Developer for z Systems Enterprise Edition。これには、IBM Debug for z Systems (以前は IBM Debug Tool (DT) for z/OS と呼ばれていた) が含まれます。
- Fault Analyzer for z/OS (FA)
- Application Performance Analyzer for z/OS (APA)

これらのツールをより効果的に使用するためには、最適化されたプログラムごとに LANGX サイド・ファイルを作成する必要があります。DT、FA、および APA はサイド・ファイルを活用して、このツールをより効果的なものにします。例えば、LANGX サイド・ファイルが提供されていれば、Debug for z Systems でソース・レベルのデバッグが可能になります。サイド・ファイルがなければ、ソース・レベルのデバッグは不可です。

LANGX サイド・ファイルの作成

IPVLANGO は、IBM Problem Determination Tools Common Component for z/OS V1.7 に付属する新しいツールです。これは Application Delivery Foundation for z/OS ツールによって共有されます。IPVLANGO は、SYSDEBUG データ・セット/コンパイラー・リスト、またはオリジナルのコンパイル済みプログラムの LANGX サイド・ファイルを ABO リスト変換と結合して、ABO によって生成されたモジュールに適した新しい LANGX サイド・ファイルを作成します。ABO によって生成されたモジュールに対して DT、FA、または APA を使用するときは、この新しい LANGX ファイルを使用してください。

Run Time Instrumentation Profiler

IBM Run Time Instrumentation (RTI) Profiler は、ご使用の z/OS アプリケーション群の実行時間 CPU パフォーマンス特性に関して収集と報告を行うためのパフォーマンス分析ツールです。

概要

RTI Profiler は、zEC12 で追加された Runtime Instrumentation Facility を使用して、低オーバーヘッドにおけるバッチ・プログラムの CPU パフォーマンス特性に関する高精度情報を収集します。

RTI Profiler を使用するためにサポートされている z/OS バージョンは以下のとおりです。

- z/OS バージョン 2.4
- z/OS バージョン 2.3
- z/OS バージョン 2.2

全体的な CPU のパフォーマンスを完全に把握するために、ご使用のアプリケーションにおけるすべての Language Environment (LE) CSECT のプロファイルが作成されます。これには、IBM COBOL、C/C++、および PL/I コンパイラでコンパイルされたプログラムや、ABO によって最適化された COBOL プログラムが含まれます。LE ライブラリー・ルーチンで費やされた時間も収集されて報告されます。

RTI Profiler からはテキスト・ファイルが出力されます。そのテキスト・ファイルには、コンパイル/最適化された CSECT ごとの詳細が含まれるほか、その CSECT 内において、プログラムが実行中に時間を費やした場所のオフセットも含まれます。

プロファイル作成対象のプログラムに関する相応のリスト・ファイルとともに、RTI Profiler の出力は、ご使用のプログラムの特定部分をマシン命令に至るまで判別するときに役立つ可能性があります。そこでは、アプリケーション CPU パフォーマンスを向上させることができる可能性があります。

詳細レベルの情報が生成されるため、RTI Profiler を対象とした基本ユース・ケースは、IBM サポート/開発担当員にデータを渡してパフォーマンス調査を手伝ってもらうことです。このシナリオでは、IBM サポート/開発担当員は、RTI Profiler を使用すること、およびその出力を、コンパイル/最適化されたオリジナル・プログラムから生成された他の成果物 (リスト・ファイルなど) とともに IBM に送信することを要求します。

より完成度の高いフル装備のアプリケーション・プロファイル作成が必要な場合は、ADFz で [Application Performance Analyzer for z/OS](#) などのプロファイル作成ツールを使用することをお勧めします。

システムの要件と制限

RTI Profiler は zEC12 システム、zBC12 システム、z13 システム、z13s システム、z14 システム、z14 ZR1 システム、および z15 システムでのみ使用できます。このような制限が存在する理由は、RTI Profiler が使用する Runtime Instrumentation Facility が、zEC12 システム以降で初めて追加されたためです。

ご使用の z/OS オペレーティング・システムが z/VM[®] で稼働している場合、RTI Profiler は機能しません。z/VM ゲストで稼働している場合は、メッセージ RISTART: AUTH REQUEST FAILED がジョブ・ログに出力されます。

RTI Profiler は、IMS バッチや、Db2[®] と対話するバッチ・プログラムなど、すべてのバッチ・アプリケーションで使用できます。

RTI Profiler は CICS アプリケーションに対しては機能せず、非バッチ IMS 環境に属するアプリケーションに対してはサポートされません。

使用の説明

RTI Profiler は、ABO がインストールされたときと同じデータ・セットに組み込まれた 2 つのメンバー BOZBXITA および BOZRIDT で構成されます。

- BOZBXITA: これは、CEEBOXITA と関連プロファイル作成ルーチンをアプリケーションのメインプログラムにリンクします。このステップでは、プロファイル作成を開始/停止できるようになるだけでなく、プログラム実行時に RTI Profiler のバッファをモニター/管理できるようにもなります。
- BOZRIDT: これは、RTI Profiler バッファ・データを処理してテキスト・ファイル・レポートを生成します。

RTI Profiler を使用する前に、プロファイル作成結果を保持するために PDS/PDSE データ・セットを割り振る必要があります。RTI Profiler で生成されたレポートは、このデータ・セットのメンバーに保管されます。次の JCL 例では、このデータ・セットの名前は HLQ.SYSPROFD です。

HLQ.SYSPROFD に推奨される割り振りパラメーターを次の表に示します。

表 11. 推奨される割り振りパラメーター	
データ・セット	推奨される割り振りパラメーター
HLQ.SYSPROFD (PDS として)	Space units : CYLINDER Primary quantity : 10 Secondary quantity : 10 Directory blocks : 10 Record format : FB Record length : 80 Block size : 27920 Data set name type PDS
HLQ.SYSPROFD (PDSE として)	Space units : CYLINDER Primary quantity: 10 Secondary quantity : 10 Directory blocks : 10 Record format : FB Record length : 80 Block size : 27920 Data set name type LIBRARY

RTI Profiler を使用するには、以下の手順に従います。

1. リンク・エディット・ステップで、既存のプログラムを再バインドして BOZBXITA を組み込みます。
2. 実行ステップで、プロファイル作成結果が保持されるデータ・セットのロケーションを指定します。

下にある手順と JCL 例で、\$HLQBOZ.BOZ210.SBOZMOD1 は ABO に対して選択されたインストール・ロケーションです。

ステップ 1. 既存のプログラムを再バインドして BOZBXITA を組み込む

最初のステップでは、プログラムの実行時に RTI Profiler が使用可能になるように、既存のプログラムを再バインドして BOZBXITA を組み込みます。

この再バインドを実行するには、メイン・エントリー・ポイントが含まれるプログラムのリンク・エディット・ステップを、ご使用のアプリケーションに合わせて変更します。

- \$HLQBOZ.BOZ210.SBOZMOD1 を SYSLIB としてリンク・エディット・ステップに追加する
- BOZBXITA をリンク・エディット・ステップへの追加入力として組み込む

このステップの JCL 例を以下に示します。

```
//LKED EXEC PGM=IEWL,PARM=$PARM
//SYSLIB DD DISP=SHR,DSN=$HLQBOZ.BOZ210.SBOZMOD1 <-- add $HLQBOZ.BOZ210.SBOZMOD1 as SYSLIB
//      DD DSN=$HLQCEE.SCEELKED,DISP=SHR
//      DD DSN=$HLQCEE.SCEELKEX,DISP=SHR
//LOAD DD DISP=SHR,DSN=$LOAD
//SYSLMOD DD DISP=SHR,DSN=$SYSLMOD($PROG)
//SYSPRINT DD SYSOUT=*
//SYSLIN DD *
//      INCLUDE LOAD($PROG)
//      INCLUDE SYSLIB(BOZBXITA) <-- add INCLUDE for BOZBXITA
//      ENTRY $PROG
//      NAME $PROG(R)
```

ステップ 2. プロファイル作成結果が保持されるデータ・セットのロケーションを指定する

2 番目のステップでは、RTI Profiler によって生成されたプロファイル作成結果が保持されるデータ・セットのロケーションを指定します。

プログラム実行用の既存 JCL において以下の操作を行います。

- \$HLQBOZ.BOZ210.SBOZMOD1 を既存の STEPLIB に追加する
- プロファイル作成結果を受け取るために DDNAME SYSPROFD を追加する

このステップの JCL 例を以下に示します。

```
//GO EXEC PGM=$PROG,REGION=0M
//STEPLIB DD DSN=$SYSLMOD,DISP=SHR <-- $SYSLMOD is the output dataset from the rebind in step 1
// DD DSN=$HLQB0Z.B0Z210.SB0ZMOD1,DISP=SHR <-- add $HLQB0Z.B0Z210.SB0ZMOD1 to STEPLIB
//SYSPROFD DD DSN=$HLQ.SYSPROFD($PROG),DISP=SHR <-- add SYSPROFD DD
```

プログラムの実行が完了すると、プロファイル作成結果が HLQ.SYSPROFD(\$PROG) に含まれます。

SYSPROFD が実行ステップで追加されていないと、メッセージ RIDATA: OPENING SYSPROFD FAILED がジョブ・ログに生成されます。また、異常終了コード ABEND=S000 U1130 REASON=000000000 が生成されます。

注:

1. ステップ 1 に関しては、RTI Profiler 用に再バインドを行うときにオリジナルのモジュールと同じリンク・オプション・セットを使用する必要があります。同じリンク・オプション・セットが使用されたことを確認するには、オリジナルのモジュールと、再バインドされたモジュールで、AMBLIST を使用します。COBOL プログラムに関して不一致が発生する可能性が最も高いのは、再バインド時に AMODE を 24 から 31 にうっかり変更する場合です。以下の JCL の例では、AMODE=24 のリンク・オプションの設定方法が示されています。

```
//LKED EXEC PGM=IEWL,PARM='LIST,MAP,AMODE=24'
```

2. プロファイル作成対象のプログラムが VS COBOL II または ABO 最適化 VS COBOL II コンパイル・プログラムを使用してコンパイルされた場合は、モジュールを再バインドしてブートストラップ・ルーチン IGZEBST を LE からの現行バージョンで置き換えるために、追加のステップが必要になることがあります。ステップ 1 の後で SYSPROFD の位置にプロファイル作成の出力が表示されない場合に、VS COBOL II または ABO 最適化 VS COBOL II コンパイル・プログラムのプロファイルを作成しようとするのであれば、以下の手順に従って IGZEBST ルーチンを置き換え、RTI プロファイル作成にモジュールを使用できるようにします。

```
//LKED EXEC PGM=IEWL,PARM=$PARM
//SYSLIB DD DISP=SHR,DSN=$HLQB0Z.B0Z210.SB0ZMOD1 <- location of B0ZRIDT and B0ZBXITA (i.e.
ABO install location)
// DD DSN=$HLQCEE.SCEELKED,DISP=SHR
// DD DSN=$HLQCEE.SCEELKEX,DISP=SHR
//LOAD DD DISP=SHR,DSN=$LOAD <- original module that will be linked with new IGZEBST and
BOZBXITA
//SYSLMOD DD DISP=SHR,DSN=$SYSLMOD($PROG) <- output module location
//SYSPRINT DD SYSOUT=*
//SYSLIN DD *
INCLUDE SYSLIB(IGZEBST) <- new bootstrap to include from CEE
INCLUDE LOAD($PROG) <- original member from LOAD to link new parts into
REPLACE -IMMED,IGZEBST <- bootstrap member to replace
INCLUDE SYSLIB(BOZBXITA) <- RTI prof member to include
ENTRY $PROG
NAME $PROG(R)
//*
```

付録 A JCL サンプル

次の JCL サンプルが、IBM Automatic Binary Optimizer for z/OS インストール・パッケージに用意されています。

```
//BOZJCLE JOB <job parameters>
//*****
//* Job Name: BOZJCLE *
//* *
//* Licensed Materials - Property of IBM *
//* 5697-AB2 *
//* Copyright IBM Corp. 2019 *
//* *
//* US government users restricted rights *
//* use, duplication or disclosure restricted *
//* by GSA ADP schedule contract with IBM Corp. *
//* *
//*****
//OPT EXEC PGM=BOZOPT,REGION=0M
//STEPLIB DD DSN=$HLQBOZ.BOZ210.SBOZMOD1,DISP=SHR
//OPTLOG DD DSN=$HLQ.BOZOUT.OPTLOG($BOZJOBID),DISP=SHR
//OPTERR DD DSN=$HLQ.BOZOUT.OPTERR($BOZJOBID),DISP=SHR
//CEEDUMP DD DSN=$HLQ.BOZOUT.CEEDUMP($BOZJOBID),DISP=SHR
//SYSPRINT DD DSN=$HLQ.BOZOUT.LISTING,DISP=SHR
//SYSBIN DD DSN=$SYSBIN,DISP=SHR
//SYSBOUT DD DSN=$SYSBOUT,DISP=SHR
//SYSIN DD *
ARCH=$ARCH
BOPT IN=DD:SYSBIN($MEMBER) OUT=DD:SYSBOUT($MEMBER)
```

この JCL サンプルでは、\$HLQBOZ.BOZ210.SBOZMOD1 が、オブティマイザー用に選択されたインストール・ロケーションです。

この例では、以下のデータ・セットが前もって割り振られている必要があります。

- HLQ.BOZOUT.OPTLOG
- HLQ.BOZOUT.OPTERR
- HLQ.BOZOUT.LISTING
- HLQ.BOZOUT.CEEDUMP

これらのデータ・セットは、下の表にある推奨パラメーターで割り振ることができます。

表 12. 推奨される割り振りパラメーター	
データ・セット	推奨される割り振りパラメーター
HLQ.BOZOUT.OPTLOG HLQ.BOZOUT.OPTERR HLQ.BOZOUT.LISTING	Space units: CYLS Primary quantity: 50 Secondary quantity: 50 Directory blocks: 10 Record format: VB ¹ Record length: 512 Block size: 27998 Data set name type: Library
HLQ.BOZOUT.CEEDUMP	Space units: CYLS Primary quantity: 10 Secondary quantity: 10 Directory blocks: 10 Record format: FB Record length: 133 Block size: 27930 Data set name type: Library

表 12. 推奨される割り振りパラメーター (続き)

データ・セット	推奨される割り振りパラメーター
注: 1. <i>HLQ.BEZOUT.OPTLOG</i> および <i>HLQ.BEZOUT.OPTERR</i> には、正常に開かれる VB のレコード・フォーマットが必要です。 <i>HLQ.BEZOUT.LISTING</i> に関しては、レコード・フォーマット FB も許可されますが、レコード長が短すぎる場合はリストから情報が切り捨てられます。切り捨てが行われないようにするには、レコード長として少なくとも 133 が推奨されます。	

\$BOZJOBID は、ユーザーによって選択された、このジョブの固有 ID です。 *HLQ.BEZOUT.OPTLOG*、*HLQ.BEZOUT.OPTERR*、および *HLQ.BEZOUT.CEEDUMP* データ・セットのそれぞれで、これはメンバー名として使用されます。 \$BOZJOBID は有効なメンバー名でなければなりません。

この JCL サンプルは、1 つのモジュールを最適化するための SYSIN DD の定義を示しています。詳しくは、28 ページの『JCL の例』および 54 ページの『最適化およびデプロイメントの使用法のシナリオ』を参照してください。例で使用されている DD 名の説明については、15 ページの『必須 DD ステートメント』を参照してください。

付録 B 戻りコード

IBM Automatic Binary Optimizer for z/OS はメッセージを発行して、情報を提供したり、可能性のある警告を提供したり、エラーを報告したりします。メッセージごとに「メッセージ戻りコード」があります (69 ページの『付録 C メッセージ』を参照)。ABO が終了時に z/OS に渡す戻りコード値は、発行された全メッセージの「メッセージ戻りコード」値の最大値です。メッセージが発行されなかった場合は、戻りコード 0 が z/OS に返されます。

表 13. IBM Automatic Binary Optimizer for z/OS 戻りコード	
戻りコード (10 進数)	説明
0	すべての処理が正常終了しました。1 つ以上の情報メッセージが発行された可能性があります。
4	正常終了しましたが、通常とは異なる状態が検出されました。1 つ以上の警告メッセージが発行された可能性があります。
12	BOPT/IEFOPZ ディレクティブまたはグローバル・オプションの処理時にエラーが検出されました。1 つ以上のメッセージが発行された可能性があります。 • 入力行の構文処理時にエラーが発生すると、行の残りはリジェクトされ、ABO は先に進んで次の入力行を処理します。 • BOPT/IEFOPZ ディレクティブの処理時にエラーが発生すると、ABO は先に進んで、次に適用可能な BOPT/IEFOPZ ディレクティブ・モジュールを処理します。BOPT/IEFOPZ ディレクティブについて処理する入力モジュールがこれ以上なくなると、そのディレクティブの処理は終了し、ABO は次の入力行に進んで次のディレクティブを処理するか、または入力行がこれ以上なければ ABO は終了します。
16	リカバリー不能エラーが検出されました。1 つ以上のメッセージが発行され、ABO は即時に処理を終了します。

付録 C メッセージ

このセクションで説明するメッセージは、OPTLOG DD に書き込まれます。いくつかの例外的なケースとして、メッセージが OPTLOG DD に書き込まれないことがあります。その場合は OPTERR DD に書き込まれ、OPTERR DD にも書き込まれない場合は JOBLOG に書き込まれます。下にリストされているメッセージにはそれぞれ、67 ページの『付録 B 戻りコード』で説明されている、z/OS に返された戻りコードを調べるための「メッセージ戻りコード」があります。

このセクションにある各 ABO メッセージの形式は BOZnnnnX です。BOZ はメッセージが ABO メッセージであることを示します。nnnn はメッセージ番号です。X は重大度の指標です。

重大度の指標は I、W、E、S、U のいずれかです。

I

通知メッセージ (RC=0)

W

警告メッセージ (RC=4)

E

エラー・メッセージ (RC=8)

S

重大エラー・メッセージ (RC=12)

U

リカバリー不能エラー・メッセージ (RC=16)

BOZ1003U プログラムがシグナル &1 を catch
 しました。戻りコード 16 で終了し
 ます。(Program caught signal &1,
 exiting with return code 16.)

説明:

予期しない状態が処理中に検出されたため、オブティマイザーは処理を続行できませんでした。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

これ以前の問題が原因で、予期しない問題が発生した可能性があります。ログ・ファイルで報告されている問題を修正し、最適化プロセスを再試行してください。問題が解消されない場合は、IBM サービスに援助を依頼してください。

メッセージ戻りコード

16

BOZ1031S An error occurred while
 attempting to open "&1".

説明:

オブティマイザーは、&1 で示されているファイルを開くことができませんでした。

システムの処置:

このオープン障害が必須のオブティマイザー DD の 1 つ (SYSIN DD など) に関連している場合、オブティマイザーは戻りコード 16 ですぐに終了します。そうでは

なく、ファイルが SYSIN 入力ファイル (グローバル・オプション、あるいは BOPT ディレクティブまたは IEFOPZ ディレクティブ) の行に指定されている場合、その行の処理は終了し、オブティマイザーは SYSIN 入力ファイル内の次の行を処理します。

ユーザーの処置:

ファイル名が正しいこと、また適切なレコード・フォーマットと適切なレコード長でファイルが割り振られていることを確認してください。

メッセージ戻りコード

必須の DD を開こうと試みている場合は 16 で、そうではない場合は 12 です。

BOZ1145U コンパイルを続行するためのメモリー
 がコンパイラーに不足していま
 す。

説明

メモリー不足が原因で、オブティマイザーは操作を続行できませんでした。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

JCL MEMLIMIT パラメーターまたは JCL REGION パラメーターを使用して、オブティマイザーが使用するメモリーを増やしてください。詳しくは、『z/OS MVS Initialization and Tuning Reference』および『z/OS MVS Initialization and Tuning Guide』を参照してください。

メッセージ戻りコード

16

BOZ1400S Directive is missing "&1" specifier.

説明

オブティマイザーが、&1 指定子を必要とする BOPT ディレクティブまたは IEFOPZ ディレクティブを検出しましたが、この指定子が欠落していました。

システムの処置:

オブティマイザーはディレクティブを破棄し、SYSIN 入力ファイル内の次の行を処理します。

ユーザーの処置:

該当する &1 指定子を追加し、ディレクティブを修正してください。

メッセージ戻りコード

12

BOZ1401S "&1" directive must be specified at start of line.

説明

SYSIN 入力ファイルの行において、&1 ディレクティブの前にオプションがあるか、その行に &1 ディレクティブが欠落しています。

システムの処置:

オブティマイザーはディレクティブを破棄し、SYSIN 入力ファイル内の次の行を処理します。

ユーザーの処置:

&1 ディレクティブを行の先頭に指定し、その行を修正してください。

メッセージ戻りコード

12

BOZ1402S Invalid specifier in "&1".

説明

BOPT ディレクティブまたは IEFOPZ ディレクティブの &1 オプションに、無効な指定子が含まれていました。例えば、オプション「SCAN=H」において「H」は無効な指定子です。

システムの処置:

オブティマイザーは、無効な指定子を持つディレクティブを破棄し、SYSIN 入力ファイル内の次の行を処理します。

ユーザーの処置:

オプションの中の指定子を有効なものに変更してください。

メッセージ戻りコード

12

BOZ1403S Invalid option "&1".

説明

SYSIN 入力ファイルの処理中に &1 が検出されましたが、&1 はオブティマイザー・ディレクティブではなく、&1 は有効なオプションではありません。

システムの処置:

オブティマイザーは、無効なオプションを持つ行を破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

正しいスペルのディレクティブまたはオプションを指定し、SYSIN の行を修正してください。

メッセージ戻りコード

12

BOZ1404S "&1" can only be specified on "&2" directive.

説明

&1 オプションが、&2 ディレクティブではないディレクティブに指定されました。例えば、SEL_ARCH は IEFOPZ ディレクティブにのみ適用可能であるため、SEL_ARCH を BOPT ディレクティブに指定することはできません。

システムの処置:

オブティマイザーは、無効なオプションを持つ行を破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

正しいオプションまたはディレクティブを指定し、&1 を持つ行を修正してください。

メッセージ戻りコード

12

BOZ1405S "&1" not allowed on "&2" directive.

説明

&2 ディレクティブにオプション &1 が含まれていますが、このオプションは &2 ディレクティブに適用可能ではありません。例えば、「IN」オプションは IEFOPZ ディレクティブに適用可能ではないため、このディレクティブには指定できません。

システムの処置:

オブティマイザーは、無効なオプションを持つ行を破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

&2 ディレクティブに適用される正しいオプションを指定し、行を修正してください。

メッセージ戻りコード

12

BOZ1406S Wildcards not supported in member specifier of "&1".

説明

SYSIN の行に、ワイルドカードが含まれているメンバー指定子 (&1) を持つ「IN」オプションと、データ・セット・メンバー指定子が含まれている「OUT」オプションが見つかりました。ワイルドカードが「IN」オプションで使用されている場合、「OUT」オプションにメンバー指定子を入れないでください。

システムの処置:

オブティマイザーは、無効なオプションを持つ行を破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

ワイルドカードを指定しないように「IN」オプションを変更するか、またはメンバー指定子を「OUT」オプションから取り除いてください。

メッセージ戻りコード

12

BOZ1407S Output specifier "&1" invalid when using wildcards on input.

説明

オブティマイザーが、ワイルドカード指定子が「IN」オプションにあって、「OUT」オプションに USS パス &1 が指定されている BOPT ディレクティブを検出しました。メンバー・ワイルドカードが「IN」オプションで使用されている場合、「OUT」オプションには、USS パスではなくデータ・セットを指定する必要があります。

システムの処置:

オブティマイザーは、無効な「OUT」オプションを持つ BOPT オプションを破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

ワイルドカードを指定しないように「IN」オプションを変更するか、またはデータ・セットを指定するように「OUT」オプションを変更してください。

メッセージ戻りコード

12

BOZ1408S Module specifier "&1" is an existing directory.

説明

オブティマイザーが、BOPT ディレクティブの「IN」オプションまたは「OUT」オプションのモジュール・ロケーションとして、HFS ディレクトリー &1 が指定されていることを検出しました。HFS では、モジュールは標準ファイルであって、ディレクトリーではありません。

システムの処置:

オブティマイザーはディレクティブを破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

「IN」オプションまたは「OUT」オプションのパス指定子を、ディレクトリーではなく標準ファイルを指すように変更してください。

メッセージ戻りコード

12

BOZ1409S Output specifier "&1" is not an existing PDS(E).

説明

オブティマイザーが、以下のどちらかを検出しました。

- IEFOPZ 構成での NEW データ・セットとしての &1
- SYSIN ファイルの (メンバー名が含まれている) ファイル指定子の DD 名またはデータ・セット名としての &1
- LOG オプションのデータ・セット指定子としての &1

しかし、&1 に関連付けられているデータ・セットが存在しなかったか、または PDS(E) ではなく順次データ・セットでした。

システムの処置:

IEFOPZ のケースでは、オブティマイザーは NEW データ・セットを無視し、IEFOPZ 構成の処理を続行します。そうではない場合は、オブティマイザーはディレクティブを破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

データ・セット・ロケーションを既存の PDS(E) に変更するか、またはオブティマイザーを実行する前に PDS(E) を割り振ってください。

メッセージ戻りコード

12

BOZ1410I Output module "&1" cannot be replaced as REPLACE=N is in effect.

説明

REPLACE=Y オプションが指定されている場合、オブティマイザーは、同じ名前の出力モジュール (&1) が既に存在することを検出したときに、この通知メッセージを出します。

システムの処置:

オブティマイザーは入力モジュールの最適化をバイパスし、次のモジュールまたは次のディレクティブを処理します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ14115 **Error getting member list from dataset specifier "&1".**

説明

オブティマイザーが、以下のどちらかを処理していました。

- PDS(E) (&1) が (メンバー・ワイルドカードが含まれている) 「IN」 オプションに指定され、PDS(E) にメンバーがない場合の BOPT ディレクティブ
- IEFOPZ 構成の OLD データ・セット (&1) にメンバーがない場合の IEFOPZ ディレクティブ

システムの処置:

BOPT ディレクティブのケースでは、オブティマイザーはディレクティブを破棄し、SYSIN ファイル内の次の行を処理します。IEFOPZ ディレクティブのケースでは、オブティマイザーは OLD データ・セットを無視し、IEFOPZ 構成の残りの部分を処理します。

ユーザーの処置:

適切なデータ・セットが BOPT ディレクティブに指定されていること、また適切なデータ・セットが IEFOPZ 構成に指定されていることを確認してください。

メッセージ戻りコード

12

BOZ14125 **IEFOPZ is not available on this system.**

説明

オブティマイザーは、IEFOPZ 機能がない z/OS システム上で IEFOPZ ディレクティブを処理していました。

システムの処置:

オブティマイザーは IEFOPZ ディレクティブを破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

IEFOPZ 機能は、z/OS V2R2 以降でのみ使用可能です。オブティマイザーが、V2R2 より前の z/OS システムで動作している場合、IEFOPZ ディレクティブを指定しないように SYSIN を変更してください。オブティマイザーが z/OS V2R2 以降で動作している場合、IEFOPZ 機能に必要な PTF をインストールするよう、システム・プログラマーに依頼してください。

メッセージ戻りコード

12

BOZ14135 **Problem with IEFOPZQ system service (return code="&1", reason code="&2"): &3.**

説明

オブティマイザーが、IEFOPZ ディレクティブの処理中、IEFOPZ 構成を読み取っているときに問題を検出しました。&1 はエラー戻りコードを示します。&2 は、構成を読み取るために使用された IEFOPZQ システム・サービスのエラー理由コードを示します。&3 は、理由コードの簡略説明です。

システムの処置:

オブティマイザーは IEFOPZ ディレクティブを破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

エラーが有効であるかどうかを調べるには、このエラー・メッセージをシステム・プログラマーに報告してください。IEFOPZ の使用法に問題がない場合、このオブティマイザー・メッセージと (該当する場合は) 他の IEFOPZ 構成情報を IBM サービスに報告してください。

メッセージ戻りコード

12

BOZ14145 **Input specifier "&1" is not an existing PDS(E).**

説明

オブティマイザーが、以下のどちらかを検出しました。

- IEFOPZ 構成での OLD データ・セットとしての &1
- SYSIN ファイルの (メンバー名が含まれている) ファイル指定子の DD 名またはデータ・セット名としての &1

しかし、&1 に関連付けられているデータ・セットが存在しなかったか、または PDS(E) ではなく順次データ・セットでした。

システムの処置:

IEFOPZ のケースでは、オブティマイザーは OLD データ・セットを無視し、IEFOPZ 構成の処理を続行します。そうではない場合は、オブティマイザーはディレクティブを破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

データ・セット名を既存の PDS(E) に変更するか、または PDS(E) を割り振ってください。

メッセージ戻りコード

12

BOZ14155 **No DD definition is supplied for "&1".**

説明

オブティマイザーが必須のオブティマイザー DD (&1) の DD 定義を見つけれなかったか、または SYSIN ファイルで使用されている DD (&1) の DD 定義が指定されていませんでした。

システムの処置:

DD の対象がオブティマイザーの必須の DD であれば、オブティマイザーは戻りコード 16 ですぐに終了します。そうではない場合は、オブティマイザーは DD 定義が含まれている SYSIN ファイルの行を破棄し、SYSIN の次の行を処理します。

ユーザーの処置:

エラーがある DD に DD 定義を指定してください。

メッセージ戻りコード

&1 が必須の DD である場合は 16 で、そうではない場合は 12 です。

BOZ1416S **A member name is not specified for PDS(E) specifier "&1".**

説明

オブティマイザーが、PDS(E) (&1) を指定していて、メンバー名を必要とする「IN」オプションまたは「OUT」オプションを検出しましたが、そのオプションにメンバーが組み込まれていませんでした。

システムの処置:

オブティマイザーは、無効な「IN」オプションまたは「OUT」オプションを持つディレクティブを破棄し、SYSIN ファイル内の次の行を処理します。

ユーザーの処置:

データ・セット・メンバーを含むように「IN」オプションまたは「OUT」オプションを変更してください。

メッセージ戻りコード

12

BOZ1417S **File "&1" does not exist.**

説明

入力ファイル &1 が見つかりませんでした。原因として、一般的に以下の 2 つがあります。

1. 既存の入力モジュール PDS(E) のメンバーが存在しない
2. 無効な HFS パスが指定された

システムの処置:

オブティマイザーは、無効なファイル指定が使用されていたディレクティブ (または入力モジュール) の処理を無視し、次の入力モジュールまたは次のディレクティブを処理します。

ユーザーの処置:

既存のデータ・セット・メンバーを指定して問題を修正するか、または既存の HFS ファイルを指すようにパス指定を修正してください。

メッセージ戻りコード

12

BOZ1418S **Invalid file specification "&1".**

説明

ファイル &1 の指定が正しくありません。正しくない指定の例は、以下のとおりです。

1. メンバー名が、以下のように 2 つ指定されている:

- DD の定義に 1 つ
- DD 定義が含まれているオブティマイザー・オプション/ディレクティブに 2 つ目

2. HFS パスが指定されているが、そのパスにあるディレクトリーが存在しないか、そのパスがアクセス可能ではない

3. DD 名またはデータ・セット名が長すぎる

システムの処置:

オブティマイザーは、無効なファイル指定が使用されていたディレクティブ (または入力モジュール) の処理を無視し、次の入力モジュールまたは次のディレクティブを処理します。

ユーザーの処置:

ファイル指定 (&1) を正しいフォーマットで指定してください。

メッセージ戻りコード

12

BOZ1419S **Output of load module(s) to "&1" is not supported when input has program object format.**

説明

入力モジュールは、より新しいプログラム・オブジェクト・フォーマットですが、最適化されたモジュール (&1) は、より古いロード・モジュール・フォーマットをターゲットにしています。この状況が発生するのは、入力モジュールが PDSE のメンバーであるか、または HFS パスに含まれるファイルであるにもかかわらず、最適化されたモジュールが PDS のメンバーをターゲットにしていたり順次データ・セットをターゲットにしていたりする場合です。

システムの処置:

オブティマイザーは入力モジュールの処理を終了し、次の入力モジュールまたは次のディレクティブを処理します。

ユーザーの処置:

最適化されたモジュールの出力ロケーション (&1) を、PDSE のメンバーまたは HFS パスに修正してください。

メッセージ戻りコード

12

BOZ1420S **Path "&1" must be absolute and begin with a "/".**

説明

入出力ファイルの指定 (&1) は HFS ファイルを指していますが、絶対パスが &1 に指定されていません。絶対ファイル指定は「/」文字で始まっている必要があります。このエラーは、例えば、オブティマイザーが入力モジュール、出力モジュール、またはリスト変換の HFS 指定を処理しているときに発生する可能性があります。

システムの処置:

オブティマイザーは、無効なパスが指定されている場合はモジュールの最適化をバイパスし、次の入力モジュールまたは次のディレクティブを処理します。

ユーザーの処置:

パス (&1) の指定を絶対パスに修正してください。

メッセージ戻りコード

12

BOZ1421S **Binder API "&1" failed: return code=&2 reason code=&3.**

説明

バインダー API (&1) を使用してモジュールを処理しているときに、このバインダー API が、予期しない戻りコード (&2) および理由コード (&3) を返しました。

システムの処置:

多くの場合、オブティマイザーは入力モジュールの処理を中止し、次の入力モジュールまたは次のディレクティブを処理します。いくつかのケース (例えば戻りコードが 4 で、理由コードが 0x83000526) では、バインダーが問題 (このケースであれば予期しない入力) をリカバリーでき、入力モジュールの最適化は続行します。

ユーザーの処置

理由コードについては、バインダーの資料を参照してください。理由コードの情報が、問題の原因を判別するために役立つ場合があります。例えば、理由コードによって、最適化の対象の入力ファイルが正しいロード・モジュール/プログラム・オブジェクト・ファイルではないことが分かる場合があります。この場合、正しい入力モジュールを指定するように JCL または SYSIN ファイルを修正してください。バインダー API 戻りコードおよび理由コードについては、「[z/OS MVS Program Management: Advanced Facilities](#)」を参照してください。

通常、このメッセージの前に BOZ4116 バインダー・メッセージが表示されます。このバインダー・メッセージには、応答において役立つ追加情報が提供されている可能性があります。

メッセージ戻りコード

オブティマイザーが処理を中止する場合は 12 で、そうではない場合は 4 です。

BOZ1422S

Module cannot be processed as it is not marked executable.

説明

オブティマイザーが、「実行可能」とマークされていない入力モジュールを検出しました。オブティマイザーによる最適化処理が成功するためには、モジュールが「実行可能」とマークされている必要があります。

システムの処置:

オブティマイザーは入力モジュールの処理を中止し、次の入力モジュールまたは次のディレクティブを処理します。

ユーザーの処置:

この問題がモジュールにおいて予期されている場合、このメッセージを無視するか、またはモジュールを除外するようにオブティマイザー・ディレクティブを変更してください。この問題が予期されるものではない場合は、バインドによる結果のモジュールが「実行可能」とマークされるよう、入力モジュールを作成するバインド・ステップを修正してください。

メッセージ戻りコード

12

BOZ1423S

Module cannot be processed as it was linked EDIT=NO or otherwise cannot be reprocessed.

説明

オブティマイザーが、編集できない入力モジュールを検出しました。この問題が発生する最も一般的なケースは、モジュールを作成するために使用されたバインド・ステップに EDIT=NO バインダー・オプションが組み込まれていた場合です。編集できないモジュールには、オブティマイザーが必要とする重要な情報が欠落しています。

システムの処置:

オブティマイザーは入力モジュールの処理を中止し、次の入力モジュールまたは次のディレクティブを処理します。

ユーザーの処置:

この問題がモジュールにおいて予期されている場合、このメッセージを無視するか、またはモジュールを除外するようにオブティマイザー・ディレクティブを変更してください。この問題が予期されるものではない場合は、入力モジュールを作成するバインド・ステップから EDIT=NO オプションを取り除いてください。

メッセージ戻りコード

12

BOZ1424S

Module cannot be appropriately processed as program is SIGNed.

説明

オブティマイザーが、「SIGN 対象」とマークされている入力モジュールを検出しました。

システムの処置:

オブティマイザーは「SIGN 対象」モジュールをサポートしていないため、その入力モジュールの処理を中止し、次の入力モジュールまたは次のディレクティブを処理します。

ユーザーの処置:

この問題がモジュールにおいて予期されている場合、このメッセージを無視するか、またはモジュールを除外するようにオブティマイザー・ディレクティブを変更してください。この問題が予期されるものではない場合は、モジュールが「SIGN 対象」とマークされないよう、モジュールを作成するバインド・ステップを修正してください。

メッセージ戻りコード

12

BOZ1428U	Insufficient memory encountered during binder API "&1": return code=&2 reason code=&3. Terminating optimizer.
-----------------	--

説明

バインダー API (&1) を使用してモジュールを処理しているときに、メモリー不足が原因で、このバインダーが処理を続行できませんでした。バインダーは、メモリー問題を示す戻りコード (&2) および理由コード (&3) を生成します。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置

JCL MEMLIMIT パラメーターまたは JCL REGION パラメーターを使用して、最適化プロセスの間に使用されるメモリーを増やしてください。詳しくは、「[z/OS MVS Initialization and Tuning Reference](#)」および「[z/OS MVS Initialization and Tuning Guide](#)」を参照してください。

通常、このメッセージの前に BOZ4116 バインダー・メッセージが表示されます。このバインダー・メッセージには、応答において役立つ追加情報が提供されている可能性があります。

メッセージ戻りコード

16

BOZ1429U	"&1" I/O error encountered during binder API "&2": return code=&3 reason code=&4. Terminating optimizer.
-----------------	---

説明

バインダー API (&2) を使用してモジュールを処理しているときに、このバインダーが、タイプ &1 の入出力エラーを検出しました。バインダー API は、戻りコード (&3) および理由コード (&4) を返しました。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置

入出力問題のタイプ (&1) および理由コード (&4) は、この問題を診断および修正するためのステップを決定するために役立ちます。理由コードの説明については、バインダーの資料を参照してください。ここでは、出力 PDS(E) またはファイル・システムがフルであることが原因での、最適化されたモジュールの「WRITE」(&1) エラーを例として挙げます。バインダー API 情報 (&2) または理由コード (&4) が、「WRITE」問題の原因を確認または調査する際に役立つ場合があります。PDS(E) (またはファイル・システム) のサイズを大きくすることで、「WRITE」問題を修正できる可能性もあります。バインダー API 戻りコードおよび理由コードについては、「[z/OS MVS Program Management: Advanced Facilities](#)」を参照してください。

通常、このメッセージの前に BOZ4116 バインダー・メッセージが表示されます。このバインダー・メッセージには、応答において役立つ追加情報が提供されている可能性があります。

メッセージ戻りコード

16

BOZ1430U	Unrecoverable "&1" error encountered during binder API "&2": return code=&3 reason code=&4. Terminating optimizer.
-----------------	---

説明

バインダー API (&2) を使用してモジュールを処理しているときに、このバインダーが、タイプ &1 のエラーを検出しました。バインダー API は、戻りコード (&3) および理由コード (&4) を返しました。

システムの処置:

バインダーは、戻りコード 16 ですぐに終了します。

ユーザーの処置

問題のタイプ (&1) および理由コード (&4) は、この問題を診断および修正するためのステップを決定するために役立ちます。問題を診断できない場合は、IBM サービスに援助を依頼してください。バインダー API 戻りコードおよび理由コードについては、「[z/OS MVS Program Management: Advanced Facilities](#)」を参照してください。

通常、このメッセージの前に BOZ4116 バインダー・メッセージが表示されます。このバインダー・メッセージには、応答において役立つ追加情報が提供されている可能性があります。

メッセージ戻りコード

16

BOZ1431S	Input module with unsupported feature (&1) encountered during binder API "&2": return code=&3 reason code=&4. Module bypassed.
-----------------	---

説明

バインダー API (&2) を使用してモジュールを処理しているときに、このバインダーが、サポートされていない機能 &1 がモジュールに含まれていることが原因で、モジュールを最適化できないことを検出しました。バインダー API は、戻りコード (&3) および理由コード (&4) を返しました。

サポートされていない機能の例として、入力がオブジェクト・モジュールである (入力がロード・モジュールまたはプログラム・オブジェクトであることに對して) 場合が挙げられます。また、この問題のもう 1 つの例として、入力モジュールが完全にはバインドされておらず、「UNRESOLVED」参照を含んでいる場合が挙げられます。

システムの処置:

ユーザーの処置

この入力モジュールはサポートされていないため、選択肢は以下のとおりです。

- メッセージを無視する
- モジュールの最適化を行わないようにオブティマイザー入力を変更する
- 問題を修正する: 例えば、「UNRESOLVED」参照を持つモジュールのケースでは、そのモジュールが完全にバインドされるよう、モジュール作成に使用されるビルド・ステップを変更してください。

バインダー API 戻りコードおよび理由コードについては、[「z/OS MVS Program Management: Advanced Facilities」](#)を参照してください。

通常、このメッセージの前に BOZ4116 バインダー・メッセージが表示されます。このバインダー・メッセージには、応答において役立つ追加情報が提供されている可能性があります。

メッセージ戻りコード

12

BOZ1432S	Output module size exceeded module format limitations and cannot be saved.
-----------------	---

説明

オブティマイザーが最適化されたモジュールを書き込もうと試みましたが、出力フォーマットの制限に達しました。PDS メンバー (または順次データ・セット) に保管されるロード・モジュールは、最も制限的なフォーマットです。(PDSE または HFS パスに書き込まれる) プログラム・オブジェクトでは、フォーマット制限が発生することはほとんどありません。

システムの処置:

オブティマイザーは入力モジュールの最適化を終了し、次のモジュールまたは次のディレクティブを処理します。

ユーザーの処置:

出力モジュールを PDS メンバーまたは順次データ・セットに保管したい場合、出力ファイル・ロケーションを PDSE のメンバーに変更してください。そうではない場合は、プログラムを複数のモジュールに分割してください。

メッセージ戻りコード

12

BOZ1436S	Invalid ARCH specification : &1
-----------------	--

説明

以下のいずれかのケースにおいて、無効な、またはサポートされていないアーキテクチャー指定 (&1) が検出されました。

1. SYSIN ファイルの処理中における ARCH オプションまたは SEL_ARCH オプション
2. IEFOPZ 構成での NEW データ・セットの処理中

システムの処置:

無効な指定が SYSIN ファイルの行で検出された場合、オブティマイザーは無効なオプションを持つ行を破棄し、SYSIN ファイル内の次の行を処理します。無効な指定が IEFOPZ 構成における NEW データ・セットの処理で検出された場合、オブティマイザーは NEW データ・セットを無視し、IEFOPZ 構成の残りの部分を処理します。

ユーザーの処置:

オブティマイザーによってサポートされている ARCH レベルを指定し、SYSIN ファイルまたは IEFOPZ 構成を修正してください。

メッセージ戻りコード

12

BOZ1437S	No BOPT or IEFOPZ directive found
-----------------	--

説明

オブティマイザーは BOPT も IEFOPZ ディレクティブも検出できません。

システムの処置:

オブティマイザーは実行を終了し、戻りコード 12 をオペレーティング・システムに返します。

ユーザーの処置:

JCL に少なくとも 1 つの BOPT または IEFOPZ ディレクティブが含まれていることを確認してください。

メッセージ戻りコード

12

BOZ1438U	dynfree dyn failed: rc=&1 for DD &2
-----------------	--

説明

オブティマイザーが、最適化プロセスの一部として使用していた内部入力 DD 名 (&2) を解放しようと試みているときに、エラーを検出しました。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

この問題は、これ以前の問題が原因で発生した可能性があります。ログ・ファイルで報告されている問題を修正し、最適化プロセスを再試行してください。問題が解消されない場合は、IBM サービスに援助を依頼してください。

メッセージ戻りコード

16

BOZ1439U	dynfree saveDyn failed: rc=&1 for DD &2
-----------------	--

説明

オブティマイザーが、最適化プロセスの一部として使用していた内部出力 DD 名 (&2) を解放しようと試みているときに、エラーを検出しました (dynfree サービスが &1 を返しました)。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

この問題は、これ以前の問題が原因で発生した可能性があります。ログ・ファイルで報告されている問題を修正し、最適化プロセスを再試行してください。問題が解消されない場合は、IBM サービスに援助を依頼してください。

メッセージ戻りコード

16

BOZ1446U	An I/O error occurred while writing &1
-----------------	---

説明

オブティマイザーが、&1 に書き込んでいるときに入出力エラーを検出しました。&1 は「リスト変換」または「ログ・ファイル」のいずれかです。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

PDS(E) またはファイル・システムがフルであるかどうかを確認し、より大きなファイルを PDS(E) に割り振るか、またはファイル・システムのサイズを大きくしてください。また、データ・セットが正しいレコード・フォーマットおよびレコード長で割り振られたかどうかを確認してください。

メッセージ戻りコード

16

BOZ1447U	An Unexpected I/O error occurred
-----------------	---

説明

オブティマイザーが、実行中に入出力エラーを検出しました。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

出力ファイルの定義を調べて (出力モジュールの除外が可能である必要があります)、正しいレコード長およびレコード・フォーマットが使用されていることを確認し、さらにファイルがフルであるかどうかを確認してください。

メッセージ戻りコード

16

BOZ1449U	Unhandled out of memory exception
-----------------	--

説明

メモリー不足が原因で、オブティマイザーは操作を続行できませんでした。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

JCL MEMLIMIT パラメーターまたは JCL REGION パラメーターを使用して、オブティマイザーが使用するメモリーを増やしてください。詳しくは、「[z/OS MVS Initialization and Tuning Reference](#)」および「[z/OS MVS Initialization and Tuning Guide](#)」を参照してください。

メッセージ戻りコード

**BOZ1450U Assertion failure, check logs for
 traceback**

説明

処理中に予期しない条件が検出されたため、オブティマイザーは操作を続行できませんでした。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

これ以前の問題が原因で、予期しない問題が発生した可能性があります。ログ・ファイルで報告されている問題を修正し、最適化プロセスを再試行してください。問題が解消されない場合は、IBM サービスに援助を依頼してください。

メッセージ戻りコード

16

**BOZ1451S dynalloc(): failed for DSN &1 with
 DD &2, errcode &3, info code &4**

説明

オブティマイザーが、内部 DD (&2) をデータ・セット (&1) に割り振っているときにエラーを検出しました。&3 は、MVS 動的割り振り関数によって返されたエラー・コードです。&4 は、MVS 動的割り振り関数によって返された情報コードです。

システムの処置:

オブティマイザーは入力モジュールの最適化を終了し、次のモジュールまたは次のディレクティブを処理します。

ユーザーの処置:

データ・セット (&2) が存在していてアクセス可能であることを確認し、同じ DD (&2) の定義が JCL に組み込まれていないことを確認してください。また、これ以前の問題が原因で、予期しない問題が発生した可能性があります。ログ・ファイルで報告されている問題を修正し、最適化プロセスを再試行してください。問題が解消されない場合は、IBM サービスに援助を依頼してください。

メッセージ戻りコード

12

**BOZ1452S dynalloc(): failed for path &1 with
 DD &2, errcode &3, info code &4**

説明

&3 は、MVS 動的割り振り関数によって返されたエラー・コードです。&4 は、MVS 動的割り振り関数によって返された情報コードです。

システムの処置:

オブティマイザーは入力モジュールの最適化を終了し、次のモジュールまたは次のディレクティブを処理します。

ユーザーの処置:

パスがアクセス可能で書き込み可能であることを確認し、同じ DD (&2) の定義が JCL に組み込まれていないことを確認してください。また、これ以前の問題が原因で、予期しない問題が発生した可能性があります。ログ・ファイルで報告されている問題を修正し、最適化プロセスを再試行してください。問題が解消されない場合は、IBM サービスに援助を依頼してください。

メッセージ戻りコード

12

**BOZ1453U dynalloc(): failed for DUMMY DD
 &1 errcode &2, info code &3**

説明

オブティマイザーが、最適化プロセスに必要な必須 DUMMY DD (&1) を割り振っているときにエラーを検出しました。&2 は、MVS 動的割り振り関数によって返されたエラー・コードです。&3 は、MVS 動的割り振り関数によって返された情報コードです。

システムの処置:

オブティマイザーは実行をすぐに終了し、戻りコード 16 をオペレーティング・システムに返します。

ユーザーの処置:

同じ DD (&1) の定義が JCL に組み込まれていないことを確認してください。また、これ以前の問題が原因で、予期しない問題が発生した可能性があります。ログ・ファイルで報告されている問題を修正し、最適化プロセスを再試行してください。問題が解消されない場合は、IBM サービスに援助を依頼してください。

メッセージ戻りコード

16

BOZ1455W Unsupported feature "&1" found

説明

このメッセージは、次のいずれかの状態のときに出力されます。

1. ABO で使用するには不適格であるコンパイラーによって作成された COBOL CSECT (つまり、コンパイルされた COBOL プログラム) を ABO が検出したとき、または ABO でサポートされていない COBOL 言語機能が CSECT に含まれているとき。
2. 複雑すぎて確実に最適化できない CSECT を ABO が検出したとき。

最初の状態では、ABO は、処理対象の CSECT でサポートされていない機能「&1」を検出しました。ABO での使用に適格であるコンパイラー、および ABO でサポー

トされていない COBOL 言語機能については、『COBOL モジュール要件』を参照してください。

2 番目の状態では、対象の CSECT が複雑すぎて実際には最適化できないと ABO は判断し、その CSECT はスキップされました。ABO が CSECT を最適化するのは、オリジナルのコンパイル済みプログラムと同じロジックで、最適化されたプログラムが実行されるのを ABO が保証できる場合に限られます。CSECT が複雑すぎて ABO がこれを保証できない場合、ABO は最適化処理を停止して、この CSECT をスキップします。当該モジュールに含まれる他の適格 CSECT はいずれも引き続き処理されます。

このメッセージは通知のためにのみ発行されます。このメッセージには、ABO に関する機能的な問題は示されていません。

システムの処置:

ABO は CSECT の最適化をバイパスし、次の CSECT の処理に進みます。

ユーザーの処置:

非常に多くのモジュールを最適化しているときに、サポートされない特定機能についてメッセージ BOZ1455 が発行される場合は、その機能がサポートされていないために ABO を有効に使用できないことを示す RFE を開いてください。

メッセージ戻りコード

4

BOZ1456S	"&1" cannot be both optimizer input and optimizer output.
-----------------	--

説明

オブティマイザーは、データ・セットまたはファイルをオブティマイザーとの間の入力として使用することを許可していません。例えば、ある PDS(E) が入力モジュールのソースである場合、その PDS(E) に最適化されたモジュールを書き込むことはできません。オブティマイザーへの入力のロケーションと、オブティマイザーからの出力のロケーションの両方として &1 が使用されている場合、このメッセージが出されます。

システムの処置:

オブティマイザーは入力モジュールの最適化を終了し、次のモジュールを処理します。

ユーザーの処置:

出力データ・セットと入力データ・セットが異なるよう、JCL または SYSIN ファイルを修正してください。

メッセージ戻りコード

12

BOZ1457S	Invalid filter expression "&1".
-----------------	--

説明

オブティマイザーは、BOPT オプティマイザー・ディレクティブの *mem_expr* パラメーターや CSECT オプティマイザー・オプションの *expr* パラメーターを処理できません。無効な構文またはワイルドカードが式に含まれているため、この式は記述どおりには処理できません。

システムの処置:

オブティマイザーは現行ディレクティブの最適化を終了し、次のディレクティブを処理します。

ユーザーの処置:

演算式を訂正してください。16 ページの『BOPT』の *mem_expr* パラメーターの説明、および『CSECT』の *expr* パラメーターの説明を参照してください。

メッセージ戻りコード

12

BOZ1490W	Warning: AMODE/RMODE conflicts encountered during binder API "&1": return code=&2 reason code=&3. Operation performed and processing continues.
-----------------	--

説明

最適化プロセスにおいて、バインダーが AMODE 設定および RMODE 設定との競合を検出しました。この問題は、バインダーが戻りコード (&2) および理由コード (&3) を出した対象のバインダー API (&1) によって検出されました。バインダー理由コードの記述と理由コード (&3) を調べると、競合の本質を詳しく理解できます。通常、この競合は入力モジュールに既に存在し、最適化プロセスによって生じたものではありません。

システムの処置:

バインダーはこの BOZ1490 警告メッセージを出し、入力モジュールの最適化を続行します。

ユーザーの処置:

この警告メッセージは、最適化される入力モジュールの問題を示す標識である可能性があります。この問題を修正するには、入力モジュールを作成するために使用されたビルド・ステップの修正が必要になる場合があります。バインダー API 戻りコードおよび理由コードについては、『z/OS MVS Program Management: Advanced Facilities』を参照してください。

メッセージ戻りコード

4

BOZ1491W	Warning: problems encountered adding aliases to directory during binder API "&1": return code=&2 reason code=&3. Module saved and processing continues.
-----------------	--

説明

最適化プロセスにおいて、別名をディレクトリーに追加しているときにバインダーが問題を検出しました。この問題は、バインダーが戻りコード (&2) および理由コード (&3) を出した対象のバインダー API (&1) によって検出されました。バインダーは、別名と同じ名前のメンバーが PDS(E) ディレクトリーに存在する場合は、その別名を PDS(E) ディレクトリーに追加することはありません。

システムの処置:

バインダーはこの BOZ1491 警告メッセージを出し、入力モジュールの最適化を続行します。

ユーザーの処置

この問題を解決するためには、既存のメンバーが別名と同じ名前で出力 PDS(E) に存在する理由を理解しておくことが重要です。次に例を示します。

- BOPT ディレクティブの IN オプションのメンバー指定子に別名を指定しないでください。別名が指定された場合、その BOPT ディレクティブを削除し、そのメンバーをターゲット・データ・セットから削除してください。
- BOPT ディレクティブの OUT オプションに正しくないメンバー指定子があると、別名との競合の原因になります。BOPT ディレクティブの OUT オプションに、IN オプションで指定されたメンバー指定子と同じメンバー指定子があることを確認してください。
- 複数の入力データ・セットから別名をマージすると、2つのデータ・セットにおいて、別名とメンバー名の競合が発生する可能性があります。入力データ・セットごとに異なる出力データ・セットを使用するようお勧めします。バインダー API 戻りコードおよび理由コードについては、「z/OS MVS Program Management: Advanced Facilities」を参照してください。

メッセージ戻りコード

4

BOZ1492W	Warning: input module "&1" with exported symbols is saved to different named module "&2".
-----------------	--

説明

オブティマイザーが、エクスポート済みシンボルが含まれている PDS(E) で入力モジュール (&1) を検出しましたが、最適化されたモジュールが出力 PDS(E) において異なる名前のメンバー (&2) に書き込まれていました。

システムの処置:

バインダーはこの BOZ1492 警告メッセージを出し、入力モジュールの最適化を続行します。

ユーザーの処置:

この問題を修正するには、最適化されたモジュールのメンバー名が入力モジュールのメンバー名と同じになるよう、JCL または SYSIN ファイルを変更してください。

い。これを行わないと、メンバー名の変更が原因で、最適化されたモジュールを探しているときにランタイム問題が発生する可能性があります。

メッセージ戻りコード

4

BOZ1493S	Concatenated DD "&1" encountered and not is allowed for "&2".
-----------------	--

説明

オブティマイザーが、複数のデータ・セットの連結である入力または出力の DD 定義 (&1) を検出しました。&2 は、DD が使用されたときのコンテキストです。例えば &2 は、DD が入力モジュール・ロケーションとして、出力モジュール・ロケーションとして、または出力リスト変換ロケーションとして使用されたことを示します。

システムの処置:

オブティマイザーは連結 DD 定義を含むディレクティブをバイパスし、次のディレクティブを処理します。

ユーザーの処置:

入力モジュール、出力モジュール、およびリスト変換の連結 DD 定義を含まないよう、JCL を修正してください。

メッセージ戻りコード

12

BOZ1494S	Module not processed as it is not fully bound.
-----------------	---

説明

最適化プロセスにおいて、完全にはバインドされていないモジュールが入力に見つかりましたが、ALLOW=NOUNRESEXE オプションが指定されていた。オブティマイザーは、ALLOW=UNRESEXE オプションが指定されていない限り、完全にはバインドされていないモジュールを処理することはありません。

システムの処置:

オブティマイザーは入力モジュールの最適化を終了し、次のモジュールまたは次のディレクティブを処理します。

ユーザーの処置:

部分的にバインドされたモジュールを最適化することが目的であれば、ALLOW=NOUNRESEXE オプションを取り除いてください。完全にバインドされたモジュールのみ最適化することが目的であれば、このエラーを無視するか、または完全にバインドされたモジュールのみ処理するよう、JCL または SYSIN ファイルを修正してください。

メッセージ戻りコード

12

BOZ4089I **IEFOPZ: did not get ARCH=&1
match for dataset: '&2' which has
ARCH=&3.**

説明

IEFOPZ オプティマイザー・ディレクティブの処理中、オプティマイザーは、IEFOPZ オプティマイザー・ディレクティブに指定された SEL_ARCH セレクター (&1) に一致しない ARCH 指定 (&3) を持つ NEW データ・セット (&2) が IEFOPZ 構成に見つかるたびに、この通知メッセージを出します。

システムの処置:

オプティマイザーは NEW データ・セットをバイパスし、構成における次の NEW データ・セットを処理します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ4091I **IEFOPZ: did not get STATE=&1
match for dataset: '&2' which has
STATE=&3.**

説明

IEFOPZ オプティマイザー・ディレクティブの処理中、オプティマイザーは、IEFOPZ オプティマイザー・ディレクティブに指定された SEL_STATE セレクター (&1) に一致しない STATE 指定 (&3) を持つ NEW データ・セット (&2) が IEFOPZ 構成に見つかるたびに、この通知メッセージを出します。

システムの処置:

オプティマイザーは NEW データ・セットをバイパスし、構成における次の NEW データ・セットを処理します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ4092I **IEFOPZ: did not get DSN='&1'
match for dataset '&2'.**

説明

IEFOPZ オプティマイザー・ディレクティブの処理中、オプティマイザーは、IEFOPZ オプティマイザー・ディレクティブに指定された SEL_OLD セレクター値 (&1) に一致しない OLD データ・セット (&2) が IEFOPZ 構成に見つかるたびに、この通知メッセージを出します。

システムの処置:

オプティマイザーは OLD データ・セットをバイパスし、構成における次の OLD データ・セットを処理します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ4097I **No members in dataset '&1' to
process**

説明

このメッセージは、処理対象のデータ・セットにメンバーが含まれていない場合に表示されます。

システムの処置:

オプティマイザーは、次のデータ・セットの処理を続行します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ4101W **No applicable COBOL code section
found, return code 4**

説明

このメッセージは、以下のケースで出されます。

1. オプティマイザーがロード・モジュールを検出したが、そのロード・モジュール内のどの CSECT も最適化しない場合 (注: REPLACE=Y オプションが指定されておらず、最適化されたモジュールが既に存在していれば、このメッセージを表示されません)
2. メンバー・ワイルドカードが「IN」オプションにある BOPT ディレクティブの後 (「IN」データ・セット内のモジュールは最適化されなかった)
3. OLD データ・セットが処理された後 (OLD データ・セット内のモジュールは最適化されなかった)
4. IEFOPZ ディレクティブが処理された後 (モジュールは最適化されなかった)

システムの処置:

オプティマイザーは、入力における次のモジュールの処理を続行します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

4

BOZ4107I **INFO: IDRL record not added to
CSECT &1 as load module format
does not support three IDRLs. 処理
を続行します。**

説明

オブティマイザーは、ロード・モジュール内の最適化された CSECT に 2 つの IDRL レコードが既に存在することが原因で、その CSECT に (バイナリー・オブティマイザー それ自体の) IDRL レコードを追加できなかった場合に、この通知メッセージを出します。CSECT ごとに最大 2 つの IDRL が、PDS 内のロード・モジュールの制限です (PDSE 内のプログラム・オブジェクトの制限ではありません)。

システムの処置:

オブティマイザーは、出力ロード・モジュールの処理を続行します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ4109I	INFO: Adding a third IDRL to load module CSECT "&1".
-----------------	---

説明

ロード・モジュール内の CSECT (&1) の処理中、オブティマイザーは、その言語レコードを CSECT (&1) の 3 番目の IDRL として追加するときに、この通知メッセージを出します。バインダーが 3 番目の IDRL を CSECT に正しく追加できるよう、バインダーを更新する必要があります。バインダー更新がシステムにインストールされていない場合、オブティマイザーは、最適化ロード・モジュールの保管を試みるときに、続けて警告メッセージを出します。

システムの処置:

オブティマイザーがモジュールの保管中に警告メッセージを出した場合、言語レコードは CSECT に追加されていない可能性があります。オブティマイザーはその CSECT の処理を続行します。そうではない場合、CSECT の処理は正常に実行されています。

ユーザーの処置:

オブティマイザーがモジュールの保管中に警告メッセージを出した場合、バインダー更新のインストールをシステム・プログラマーに依頼し、最適化プロセスをもう一度実行する必要があります。そうではない場合は、ユーザーによる処置は必要ありません。このメッセージが出されたときに必要になるバインダー更新は、APAR OA50460 で行われます。詳しくは、[3 ページの『サポート対象のオペレーティング・システム』](#)を参照してください。

メッセージ戻りコード

0

BOZ4110I	INFO: performing a second bind to handle private section "&1" in class "&2" referring to ENTRY "&3" at offset &4.
-----------------	--

説明

オブティマイザーは、COBOL ENTRY ステートメント (&3) を持つ CSECT の処理中、オフセット (&4) のクラス (&2、通常は C_WSA) にある私用セクション (&1) から ENTRY (&3) への参照があったときに、このメッセージを出します。2 番目のバインドが正常に行われるよう、バインダーを更新する必要があります。バインダー更新がない場合、オブティマイザーに問題が生じる可能性があります。

システムの処置:

バインダー更新が使用可能であれば、処理は正常に完了します。ただし、バインダー更新が使用可能でない場合、2 番目のバインドが正常に完了したように見えても、ランタイム・エラーが発生する場合があります。

ユーザーの処置:

バインダー更新がシステムにインストールされていれば、処置は必要ありません。そうではない場合、バインダー更新のインストールをシステム・プログラマーに依頼し、最適化プロセスをもう一度実行してください。このメッセージが出されたときに必要になるバインダー更新は、APAR OA50460 で行われます。詳しくは、[3 ページの『サポート対象のオペレーティング・システム』](#)を参照してください。

メッセージ戻りコード

0

BOZ4111I	INFO: performing update to private section "&1" in class "&2" referring to ENTRY "&3" at offset &4.
-----------------	--

説明

オブティマイザーは、COBOL ENTRY ステートメント (&3) を持つ CSECT の処理中、オフセット (&4) のクラス (&2) にある私用セクション (&1) から ENTRY (&3) への参照があったときに、このメッセージを出します。2 番目のバインドが正常に行われるよう、バインダーを更新する必要があります。バインダー更新がない場合、オブティマイザーは参照の処理中にエラー・メッセージを出します。

システムの処置:

バインダー更新が使用可能でない場合、オブティマイザーは参照の処理中にエラーを出します。そうではない場合、バインダーは参照を正常に処理します。

ユーザーの処置:

オブティマイザーが参照の処理中にエラーを出した場合、バインダー更新のインストールをシステム・プログラマーに依頼し、最適化プロセスをもう一度実行してください。そうではない場合、処置は必要ありません。このメッセージが出されたときに必要になるバインダー更新は、APAR OA50460 で行われます。詳しくは、[3 ページの『サポート対象のオペレーティング・システム』](#)を参照してください。

メッセージ戻りコード

0

BOZ4113I	CSECT &1 was excluded by filter - skip
-----------------	---

説明

このメッセージは、CSECT オプティマイザー・オプションにおける式が原因でオプティマイザーが CSECT を除外する場合に表示されます。

システムの処置:

オプティマイザーは、次の CSECT の処理を続行します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ4114I	INFO: processing module that is not fully bound with ALLOW=UNRESEXE option in effect.
-----------------	--

説明

ALLOW=UNRESEXE オプションが指定されている場合、オプティマイザーは、完全にはバインドされていないモジュールを入力で検出したときに、この通知メッセージを出します。

モジュールが完全にバインドされている場合、このメッセージは出されません。

このメッセージで、オプティマイザーによって処理された、部分的にバインドされたモジュールを判別できます。

システムの処置:

バインダー更新が使用可能でない場合、オプティマイザーは参照の処理中にエラーを出します。そうではない場合、バインダーは参照を正常に処理します。

ユーザーの処置:

オプティマイザーは、部分的にバインドされたモジュールを処理し、最適化された状態で出力します。

メッセージ戻りコード

0

BOZ4116I	Binder message "&1"
-----------------	--------------------------------

説明

オプティマイザーはバインダー・サービスを使用するため、モジュールの最適化を妨げるサービスは失敗する可能性があります。バインダー・サービスが失敗すると、ABO は BOZ1429 などのメッセージを生成し、何らかの方法で処理を終了します。BOZ1429 メッセージには、バインダー・サービスの失敗理由に関する詳細情報が含

まれていないことがあります。例えば、BOZ1429 メッセージには、バインダーが入力モジュールに関する何らかの問題を検出したことが示されていますが、バインダーが検出した正確な入力問題は含まれていません。オプティマイザーは、失敗したバインダー・サービスに関する詳細情報を提供するために、重大バインダー・メッセージを取り込み、バインダー・メッセージのテキストを BOZ4116 通知メッセージの &1 内に組み込みます。すなわち、バインダー・サービスが失敗すると、ABO は通常は 2 つのメッセージを表示します。

1. 重大バインダー・メッセージのテキストが &1 に含まれている BOZ4116 メッセージ。BOZ4116 メッセージの後に次のメッセージが表示されます。

2. BOZ1429 などの要約メッセージ。これは、バインダー失敗の一般的性質を示します。

システムの処置:

オプティマイザーのアクションを判別するには、要約メッセージの『システム処置』セクションを参照してください。

ユーザーの処置:

実行すべき処置を判断するには、要約メッセージの『ユーザー応答』セクションを参照してください。

BOZ4116 メッセージからは、応答の手引きとなる情報を得ることができる場合があります。

メッセージ戻りコード

0

BOZ4117I	Member "&1" was excluded by filter - skip
-----------------	--

説明

このメッセージは、BOPT ディレクティブの IN オプションにおける式が原因でオプティマイザーがモジュールを除外する場合に表示されます。

システムの処置:

オプティマイザーは、次のモジュールの処理を続行します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

BOZ4119S	Continuation indicated on SYSIN line &1. Unable to read SYSIN line &2.
-----------------	---

説明

SYSIN の行 &1 で継続が指示され、行 &1 の非ブランクの最終文字が継続文字 (「+」または「-」) でした。オプティマイザーは SYSIN の読み取り時に行 &2 を読み取れませんでした。

システムの処置:

オブティマイザーは、この行を破棄します。

ユーザーの処置:

SYSIN の行 &1 の末尾にある継続文字を削除するか、または行 &1 を続ける SYSIN に新しい行 &2 を追加してください。

メッセージ戻りコード

12

BOZ4120S	Cannot have more than one IN= or OUT= specifier.
-----------------	---

説明

このメッセージは、複数の IN 指定子または OUT 指定子が BOPT ディレクティブで検出される場合に表示されます。

システムの処置:

オブティマイザーは、次の BOPT ディレクティブの処理を続行します。

ユーザーの処置:

BOPT ディレクティブから IN 指定子や OUT 指定子を削除して、それぞれの指定子がディレクティブごとに 1 つのみとなるようにします。

メッセージ戻りコード

12

BOZ4121S	Invalid log specification. '&1' is not a directory.
-----------------	--

説明

LOG オプションの有効な指定子はディレクトリーのみです。 &1 はディレクトリーではありません。

システムの処置:

LOG オプションは、無視されます。

ユーザーの処置:

サポート対象フォーマットになるように LOG の指定を訂正してください。サポートされる仕様については、『LOG』を参照してください。

メッセージ戻りコード

12

BOZ4124I	The HANDLERS option is now deprecated and no longer necessary; the previously default HANDLERS=Y behaviour now always applies.
-----------------	---

説明

現在、オブティマイザー・オプション HANDLERS は非推奨となり、サポートされなくなり、必須でもありません。

システムの処置:

オブティマイザーは常に、指定された HANDLERS オプションとは関係なく、従来のデフォルト・オプション HANDLERS=Y が有効になっているかのように動作します。

ユーザーの処置:

ユーザーによる処置は必要ありません。

メッセージ戻りコード

0

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510
東京都中央区日本橋箱崎町 19 番 21 号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス 渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。

本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、[“Copyright and trademark information”](#) の Web をご覧ください。

資料名リスト

IBM Automatic Binary Optimizer for z/OS 資料

IBM Automatic Optimizer for z/OS の APAR および PTF に関する最も完成度の高い最新の情報については、[フィックス・リストおよび新機能のページ](#)を参照してください。

以下の資料が [IBM Automatic Binary Optimizer for z/OS ライブラリー](#) に用意されています。

- ユーザーズ・ガイド (SC43-5133-00)
- *Program Directory* (GI13-4513-04)

関連資料

z/OS 資料

[「z/OS Internet Library」](#) に、以下の資料が用意されています。

- 「z/OS MVS 初期設定およびチューニング解説書」(SA88-7085) には、parmlib メンバー IEFOPZxx に関する情報があります。
- 「z/OS MVS プログラム管理: 拡張機能」(SA88-7064) には、バインダー API の戻りコードおよび理由コードに関する情報が記載されています。
- 「z/OS MVS システム 管理機能 (SMF)」(SA88-7082) には、IEFOPZ 構成を取り込む SMF レコード 90 サブタイプ 38 に関する情報があります。
- 「z/OS MVS システム・メッセージ 第 8 巻 (IEF - IGD)」(SA88-5573) には、メッセージに関する情報があります。

Enterprise COBOL for z/OS の資料

以下の資料が [Enterprise COBOL for z/OS ライブラリー](#) にあります。

- カスタマイズ・ガイド (SC43-3029): z/OS の下で Enterprise COBOL を計画したりカスタマイズしたりするときに役立つ情報が記載されています。
- 言語解説書 (SC43-3031): IBM COBOL コンパイラー用にプログラムを作成するために必要となる COBOL 言語およびリファレンスに関する情報が記載されています。
- プログラミング・ガイド (SC43-3030): プログラムやクラスを作成、コンパイル、およびデバッグするときに役立つ情報や例が記載されています。
- 移行ガイド (GC43-3028): 最新バージョンの IBM Enterprise COBOL に移行するときに役立つ情報が記載されています。
- *Performance Tuning Guide* (SC27-9202): IBM Enterprise COBOL for z/OS を使用するときの重要なパフォーマンス利点やチューニング考慮事項が明らかにされています。
- *Messages and Codes* (SC27-4648): コンパイラーやプリプロセッサのメッセージを理解するときに役立ちます。

Application Delivery Foundation for z Systems 資料

次の資料が [IBM Knowledge Center](#) にあります。

- *IBM Application Performance Analyzer for z/OS* ユーザーズ・ガイド (SC43-3664): システム制約を明らかにしたりアプリケーション・パフォーマンスを向上させたりするときに役立つ情報が記載されています。

- *IBM Developer for z Systems* 資料 (オンライン版のみ): 開発者の生産性を向上させるために設計された統合開発環境 (IDE) に関する情報が記載されています。
- *IBM Fault Analyzer for z/OS ユーザーズ・ガイド*および*リファレンス (SA88-7132)*: アプリケーション障害やシステム障害の分析/修正に関する情報が記載されています。



SC43-5133-00

