

Enterprise COBOL for z/OS



カスタマイズ・ガイド

バージョン 6.1

Enterprise COBOL for z/OS



カスタマイズ・ガイド

バージョン 6.1

お願い

本書および本書で紹介する製品をご使用になる前に、93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 Enterprise COBOL for z/OS バージョン 6 リリース 1 (プログラム番号 5655-EC6) および新しい版で
- | 明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。製品のレベルに応じ
- | た正しい版を使用していることを確認してください。

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： SC27-8712-00
Enterprise COBOL for z/OS
Customization Guide
Version 6.1
First edition

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 IBM Corporation 1996, 2016.

目次

図	v
表	vii
前書き	ix
本書について	ix
構文図の読み方	ix
マクロ計画ワークシートの使用	x
変更の要約	xi
第 1 章 Enterprise COBOL のカスタマイズ	1
インストール後の変更: なぜカスタマイズが必要か	1
コンパイラー・オプションのデフォルト値を変更する計画	2
コンパイラー・オプションの固定	2
コンパイラー・オプションおよびコンパイラー・フェーズの変更	3
コンパイラー・フェーズを共有ストレージに置く計画	6
コンパイラー・フェーズを共有ストレージに置く目的	6
コンパイラー・フェーズおよびそのデフォルト	8
追加の予約語テーブルを作成する計画	13
追加の予約語テーブルを作成する目的	13
ネストされたプログラムの使用の制御	13
Enterprise COBOL と一緒に提供される予約語テーブル	13
製品登録を使用して Enterprise COBOL を使用可能/使用不可にする	15
第 2 章 Enterprise COBOL コンパイラー・オプション	17
COBOL コンパイラー・オプションの指定	17
矛盾するコンパイラー・オプション	17
標準準拠のためのコンパイラー・オプション	18
コンパイラー・オプションの構文および説明	18
ADATA	19
ADEXIT	19
ADV	20
AFP	21
ALOWCBL	21
ARCH	22
ARITH	24
AWO	24
BLOCK0	25
BUF	25
CICS	26
CODEPAGE	26
COMPILE	27
COPYRIGHT	28

CURRENCY	28
DATA	30
DBCS	31
DBCSXREF	31
DECK	32
DIAGTRUNC	32
DISPSIGN	33
DLL	34
DYNAM	34
EXPORTALL	35
FASTSRT	36
FLAG	37
FLAGSTD	37
HGPR	40
INEXIT	40
INTDATE	41
LANGUAGE	42
LIBEXIT	42
LINECNT	43
LIST	43
LITCHAR	44
MAP	45
MAXPCF	45
MDECK	47
MSGEXIT	47
NAME	48
NSYMBOL	48
NUM	49
NUMCLS	49
NUMPROC	50
OBJECT	51
OFFSET	51
OPTIMIZE	52
OUTDD	53
PGMNAME	53
PRTEXIT	54
QUALIFY	55
RENT	56
RMODE	57
RULES	58
SEQ	59
SERVICE	59
SOURCE	60
SPACE	60
SQL	60
SQLCCSID	61
SQLIMS	62
SSRANGE	63
STGOPT	64
SUPPRESS	64
TERM	65

TEST	65	制御ステートメントのオペランドのコーディング	85
THREAD	66	制御ステートメントのオペランドのコーディング	86
TRUNC	68	規則	86
VBREF	69	ABBR ステートメント	86
VLR	70	INFO ステートメント	86
I VSAMOPENFS	71	RSTR ステートメント	87
WORD	71	非 SMP/E JCL の変更および実行	87
XMLPARSE	72	予約語テーブルを作成する JCL の実行	88
XREFOPT	73	共有ストレージへの Enterprise COBOL モジュール	
ZONECHECK	74	の配置	88
ZONEDATA	75	インストール先でのカタログ式プロシージャの調	
ZWB	76	整	89
第 3 章 Enterprise COBOL のカスタマイズ	79	I 付録. Enterprise COBOL for z/OS のアクセスビリティ機能	91
ユーザー変更の要約	79	特記事項	93
コンパイラー・オプションのデフォルトの変更	80	プログラミング・インターフェース情報	95
コンパイラー・オプション・デフォルト・モジュールの変更	81	商標	96
固定として指定されたオプションのオーバーライド	82	リソース・リスト	97
予約語の変更	82	Enterprise COBOL for z/OS	97
予約語テーブルの作成または変更	84	関連資料	97
制御ステートメントのコーディング	84	索引	99
制御ステートメントのコーディング規則	85		



1. IGYCOPT コンパイラー・オプション / フェーズ・マクロの構文形式 4
2. 予約語制御ステートメントの構文形式 84

表

1. コンパイラー・オプション用の IGYCDOPT ワ ークシート	4	5. LANGUAGE コンパイラー・オプションの値	42
2. コンパイラー・フェーズ用の IGYCDOPT プロ グラム・ワークシート	12	6. RENT および RMODE が常駐モードに与える 影響	56
3. 矛盾するコンパイラー・オプション	17	7. RMODE および RENTINORENT が常駐モード に与える影響	57
4. DISPSIGN=COMPAT オプションまたは DISPSIGN=SEP オプションが指定された DISPLAY 出力:	34	8. Enterprise COBOL 用のユーザー変更ジョブの 要約	80

前書き

本書について

本書は、IBM® Enterprise COBOL for z/OS® をそれぞれの設置場所でカスタマイズする責任のあるシステム・プログラマーを対象に書かれています。本書は、z/OS のもとで Enterprise COBOL のカスタマイズの計画と実行を行う際に必要となる情報について説明しています。また、組織にとっての Enterprise COBOL の価値を査定するために役立つ情報も含まれています。

本書で使用している「オペレーティング・システム」という総称用語は、z/OS を指します。

本書を使用し、正常にカスタマイズを行うためには、Enterprise COBOL およびシステム稼働環境を理解する必要があります。

構文図の読み方

このセクションでは、本書中の構文図の読み方を説明します。

- 構文図は、左から右、上から下へと線をたどってください。下の表は、構文図の線の始まりと終わりにある記号の意味を示しています。

記号	意味
>>-	構文図の始まりを示しています。
->	構文図が次の線に続くことを示しています。
>-	構文図が前の線から続いていることを示しています。
-<<	構文図の終わりを示しています。

完全なステートメント以外の構文単位の図は、>- 記号で始まり、-> 記号で終わっています。

- 必須項目は、横線 (幹線) 上に示されています。

▶▶—STATEMENT—required item—▶▶

- 任意指定項目は、幹線の下に示されています。

▶▶—STATEMENT—
└ optional item ┘—▶▶

- 2 つ以上の項目から選択できる場合は、それらの項目が縦に重ねられています。

いずれか 1 つの項目を選択しなければならない場合は、重ねられた項目のうちの 1 つが幹線上に示されています。デフォルトがあれば、それが幹線上に示されます。ユーザーが別の選択項目を指定しない場合は、IGYCOPT マクロはこのデフォルトを選択します。デフォルトは、プログラムが実行されているシステムによって異なる場合があります。



いずれか 1 つの項目の選択が任意である場合は、重ねられた項目全体が幹線の下に示されています。



- 幹線から分岐して左へ戻る矢印は、反復可能な項目を示しています。



重ねられた項目の上に反復矢印がある場合は、重ねられた項目から 2 つ以上の項目を選択するか、または 1 つの項目を繰り返すことができます。

- キーワードは大文字で示されています (例えば、**PRINT**)。キーワードは、示されているとおりに入力しなければなりません。変数はイタリック体で示されています (例えば、*item*)。変数は、ユーザーが指定する名前または値です。
- 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は、それらを構文の一部として入力しなければなりません。
- パラメーターを区切るには、1 つ以上のブランクまたはコンマを使用してください。

構文図におけるアスタリスク (*) の意味と詳しい説明については、18 ページの『コンパイラー・オプションの構文および説明』を参照してください。

マクロ計画ワークシートの使用

本書に記載されている計画ワークシート (4 ページの『コンパイラー・オプション用の IGYCDOPT ワークシート』および 11 ページの『コンパイラー・フェーズ用の IGYCDOPT ワークシート』) を、Enterprise COBOL のカスタマイズを準備するために使用できます。これらのワークシートに記入することにより、IBM 提供のデフォルトから変更する必要がある値を容易に判別することができます。記入後のワークシートを、IBM 提供のデフォルト値をカスタマイズする際の基礎として使用することもできます。

各ワークシートのヘッディングは、互いに少し異なります。コンパイラー・オプション用ワークシートの各欄のヘッディングの説明については、以下の定義リストを参照してください。

コンパイラー・オプション

特定のインストール・マクロに含まれるオプション。この欄には、マクロに指定するとおりの形でオプションが示されています。

固定の場合 * を記入

アプリケーション・プログラマーがオーバーライドできないオプション。固定するオプションについてのみ、アスタリスク (*) を記入してください。

選択を記入

各オプションに関連付けられる値。用意されているスペースに、各オプションに割り当てたい値を記入してください。適切な値を選択できるよう支援するため、「構文の説明」欄に参照情報が示されています。

IBM 提供のデフォルト

オプションを変更しなかった場合に指定のインストール・マクロに渡される値。IBM 提供のデフォルトが、設定する値と同じである場合は、その特定のマクロ内のオプションを変更する必要はありません。

構文の説明

構文図およびオプションについての具体的情報が含まれているトピック。

ワークシートが完成したら、IBM 提供のデフォルトと異なるオプションを確認してください。それらの項目をインストール・マクロにコーディングする必要があります。ワークシートの各項目は、それらの項目の順序が実際のコーディングのセマンティクスと整合するような順序で並べられています。

変更の要約

このセクションでは、Enterprise COBOL バージョン 5 以降で本書に対して行われた主な変更をリストします。最新の技術上の変更には、PDF 版では左側の余白に縦棒 (|) が付けられています。

バージョン 6 リリース 1

- コンパイラー・オプションに対するいくつかの変更:
 - 新しいコンパイラー・オプション:
 - SUPPRESS (64 ページの『SUPPRESS』)
 - VSAMOPENFS (71 ページの『VSAMOPENFS』)
 - 変更されたコンパイラー・オプションは以下のとおりです。
 - SSRANGE: コンパイラーによる参照変更長の検査方法を制御する新しいサブオプション ZLEN および NOZLEN が、SSRANGE コンパイラー・オプションに追加されました (63 ページの『SSRANGE』)。
- LVLINFO インストール・オプションが除去されました。ビルド・レベル情報は、必ずリスト・ヘッダーに含まれます。

APAR PI40822 および PI40853 用の PTF がインストールされたバージョン 5 リリース 2

- コンパイラー・オプションに対するいくつかの変更:
 - 新しいコンパイラー・オプションは以下のとおりです。
 - ZONECHECK=NO | MSG | ABD (74 ページの『ZONECHECK』)
 - 変更されたコンパイラー・オプションは以下のとおりです。
 - ZONEDATA: 新しいサブオプション NOPFD が ZONEDATA コンパイラー・オプションに追加されました。ZONEDATA=NOPFD を使用すると、COBOL V4 で NUMPROC=NOPFD|PFDF を使用した場合に COBOL V4 が実行するのと同じ方法で、ゾーン 10 進数データの比較を実行するコードをコンパイラーが生成できるようになります (75 ページの『ZONEDATA』)。

バージョン 5 リリース 2

- コンパイラー・オプションに対するいくつかの変更:
 - 新しいコンパイラー・オプション:
 - COPYRIGHT (28 ページの『COPYRIGHT』)
 - QUALIFY=COMPAT | EXTEND (55 ページの『QUALIFY』)
 - RULES (58 ページの『RULES』)
 - SERVICE (59 ページの『SERVICE』)
 - SQLIMS (62 ページの『SQLIMS』)
 - VLR=COMPAT | STANDARD (70 ページの『VLR』)
 - XMLPARSE=XMLSS | COMPAT (72 ページの『XMLPARSE』)
 - ZONEDATA=PF | MIG (75 ページの『ZONEDATA』)
 - 変更されたコンパイラー・オプション:
 - ARCH: ARCH(6) は現在受け入れられません。新しい上位レベルの ARCH(11) が受け入れられ、ARCH(7) がデフォルトです (22 ページの『ARCH』)。
 - MAP: 新しいサブオプション HEX および DEC が MAP コンパイラー・オプションに追加されました。これらのサブオプションは、コンパイラー・リストの MAP 出力 (45 ページの『MAP』) に、16 進オフセットと 10 進オフセットのどちらを表示するかを制御します。
 - 以下のコンパイラー・オプションは削除されました。
 - SIZE
 - XML COMPAT サポートが復活しました。XMLPARSE(XMLSS|COMPAT) コンパイラー・オプションを指定して、解析に COBOL ライブラリーの互換モード COBOL XML パーサーを使用するか、z/OS XML System Services パーサーを使用するかを選択できます。これにより、Enterprise COBOL V5 コンパイラーへの移行が容易になります。

バージョン 5 リリース 1 モディフィケーション 1

最新サービスが適用されている COBOL V5.1.0 は、V5.1.1 のように動作し、以下の変更されたコンパイラー・オープンおよび新しいコンパイラー・オプションを備えています。

- いくつかの例外的なケースを除き、プログラムの AMODE 24 実行がサポートされています。Enterprise COBOL V5.1.1 でコンパイルされた多くのプログラムが AMODE 31 または AMODE 24 で実行されます。
- 新規コンパイラー・オプション SQLIMS では、IMS™ SQL コプロセッサが使用できます。(62 ページの『SQLIMS』)
- VLR=COMPAT | STANDARD (70 ページの『VLR』)
- XMLPARSE=XMLSS | COMPAT (72 ページの『XMLPARSE』)
- 新しいサブオプション HEX および DEC が MAP コンパイラー・オプションに追加されました。これらのサブオプションは、コンパイラー・リストの MAP 出力 (45 ページの『MAP』) に、16 進オフセットと 10 進オフセットのどちらを表示するかを制御します。

バージョン 5 リリース 1

- コンパイラー・オプションに対するいくつかの変更:
 - 新しいコンパイラー・オプション:
 - AFP=VOLATILE | NOVOLATILE (21 ページの『AFP』)
 - ARCH=*n* (22 ページの『ARCH』)
 - DISPSIGN=SEP | COMPAT (33 ページの『DISPSIGN』)
 - HGPR=PRESERVE | NOPRESERVE (40 ページの『HGPR』)
 - MAXPCF=*n* (45 ページの『MAXPCF』)
 - STGOPT=YES | NO (64 ページの『STGOPT』)
 - 変更されたコンパイラー・オプション:
 - MDECK オプションは、LIB オプションに依存しなくなりました。
 - NUMPROC コンパイラー・オプションの MIG サブオプションは、サポートされなくなりました。(50 ページの『NUMPROC』)
 - コンパイル時に組み込まれた範囲検査は、ランタイム・オプション NOCHECK を使用して実行時に無効にすることはできません。(63 ページの『SSRANGE』)
 - 16 MB 境界より上での NORENT プログラムの実行はサポートされていません。(56 ページの『RENT』)
 - TEST コンパイラー・オプションの HOOK | NOHOOK および SEPARATE | NOSEPARATE サブオプションはサポートされなくなりましたが、移行を容易にするために引き続き許容されます。TEST コンパイラー・オプションには、新たに SOURCE および NOSOURCE サブオプションが追加されました。(65 ページの『TEST』)
 - TEST=NO オプションが拡張されて、サブオプション DWARF および NODWARF が組み込まれました。(65 ページの『TEST』)
 - EXIT コンパイラー・オプションは DUMP コンパイラー・オプションと相互排他的ではなくなり、コンパイラー終了規則が更新されました。
 - OPTIMIZE オプションが変更され、いくつかのレベルの最適化が可能になりました。(52 ページの『OPTIMIZE』)
 - 以下のコンパイラー・オプションは削除されましたが、移行を容易にするために引き続き許容されます。これらのオプションのいずれかを指定すると、通知または警告の診断が表示されます。
 - DATEPROC - 2000 年サポートは提供されなくなりました。
 - LIB - ライブラリー処理は常時行われます。
 - SIZE=MAX - この選択項目は意味がなくなりました。
 - YEARWINDOW - 2000 年サポートは提供されなくなりました。
 - XMLPARSE - XML System Services パーサーは常時使用されます。
- 追加の Unicode 機能を提供するために、新しい組み込み関数が追加されました。
 - ULENGTH
 - UPOS
 - USUBSTR
 - USUPPLEMENTARY

- UVALID
- UWIDTH
- プログラムの AMODE 24 実行はサポートされなくなりました。Enterprise COBOL V5.1.0 実行可能モジュールは AMODE 31 でなければなりません。
- 再利用可能な COBOL 環境を管理するための IGZERRE インターフェースおよび ILBOSTP0 インターフェースは、Enterprise COBOL V5 でコンパイルされたプログラムが含まれるアプリケーションではサポートされません。
- 新しい特殊レジスター XML-INFORMATION は、XML イベント用に提供された XML コンテンツが完了しているか、次のイベントに継続するかを簡単に判別できるメカニズムを備えています。
- COBOL ライブラリーの互換モード COBOL XML パーサーの使用はサポートされなくなりました。V5 プログラム内の XML PARSE ステートメントには、常に z/OS XML System Services パーサーが使用されます。
- 新規の NAME 句、TYPE 句、および SUPPRESS 句が XML GENERATE ステートメントに追加されました。
- JCL カタログ式プロシージャの変更:
 - COBOL コンパイラーは現在、z/OS TSO 下またはバッチ下でのコンパイル時に、15 個のユーティリティ・データ・セット (SYSUT1 - SYSUT15) および SYSDCK データ・セットを必要とします。以下の JCL カタログ式プロシージャが変更されました。
 - IGYWC
 - IGYWCL
 - IGYWCLG
 - 以下の JCL カタログ式プロシージャは現在サポートされていません。これらはいずれも、現在サポートされていない Language Environment® Prelinker または DFSMS Loader を使用しているためです。
 - IGYWCG
 - IGYWCPG
 - IGYWCPL
 - IGYWCPLG
 - IGYWPL
- デバッグ機能強化:
 - プログラム・オブジェクトに NOLOAD デバッグ・セグメントがある場合、Enterprise COBOL V5 デバッグ・データは必ず実行可能ファイルに適合し、検索場所となるデータ・セットのリストを指定しなくてもデバッグ・データは常に使用可能であり、ロードされるプログラムのサイズが増えることはありません。
 - Enterprise COBOL V5 に付属の Debug Tool に関する他の改善については、「Enterprise COBOL 移行ガイド」の『IBM Enterprise COBOL バージョン 5 の Debug Tool の変更点』を参照してください。
- Enterprise COBOL バージョン 5 リリース 1 と旧バージョンの COBOL との相互運用性に制約事項が追加されました。詳しくは、「Enterprise COBOL 移行ガイド」の『IBM COBOL プログラムの古いレベルとのインターオペラビリティ』を参照してください。

第 1 章 Enterprise COBOL のカスタマイズの計画

Enterprise COBOL のカスタマイズを計画するときには、コンパイラー・オプションのデフォルト値を変更するかどうか、コンパイラー・フェーズを共有ストレージに置くかどうか、追加の予約語テーブルを作成するかどうかを検討する必要があります。

カスタマイズを計画する際、以下の説明が役立ちます。

- 『インストール後の変更: なぜカスタマイズが必要か』
- 2 ページの『コンパイラー・オプションのデフォルト値を変更する計画』
- 6 ページの『コンパイラー・フェーズを共有ストレージに置く計画』
- 13 ページの『追加の予約語テーブルを作成する計画』
- 15 ページの『製品登録を使用して Enterprise COBOL を使用可能/使用不可にする』

IBM デバッグ・ツール をインストールしようとしている場合は、そのモジュールを共有ストレージに置くかどうか、また、デバッグ・ツール と連携して機能するように CICS® 環境をセットアップするかどうかを決めることができます。

実際のカスタマイズ手順については、79 ページの『第 3 章 Enterprise COBOL のカスタマイズ』を参照してください。

本書には、マクロ内の IBM 提供のデフォルト値の変更を計画する際に役立つワークシートも含まれています。計画シートについては、x ページの『マクロ計画ワークシートの使用』を参照してください。

重要: Enterprise COBOL のカスタマイズを計画する際には、この製品を使用するアプリケーション・プログラマーと相談してください。これにより、カスタマイズによって適用される変更が、アプリケーション・プログラマーの要件を満たすと同時に、開発されるアプリケーションをサポートするようになります。

インストール後の変更: なぜカスタマイズが必要か

Enterprise COBOL のインストール時には、コンパイラー・オプションおよびコンパイラー・フェーズ、および予約語テーブルに関して IBM 提供のデフォルトを受け取ります。アプリケーション・プログラマーのニーズにより合うように、Enterprise COBOL をカスタマイズしたい場合があります。

Enterprise COBOL をインストールした後、以下を行うことができます。

- コンパイラー・オプションのデフォルト値を変更する: 2 ページの『コンパイラー・オプションのデフォルト値を変更する計画』を参照してください。
- コンパイラー・オプションを固定する: 2 ページの『コンパイラー・オプションの固定』を参照してください。
- コンパイラー・フェーズの常駐値を変更する: 6 ページの『コンパイラー・フェーズを共有ストレージに置く計画』を参照してください。
- 追加の予約語テーブルを作成する: 13 ページの『追加の予約語テーブルを作成する計画』を参照してください。

コンパイラー・オプションのデフォルト値を変更する計画

コンパイラー・オプションのデフォルト値およびコンパイラー・フェーズの常駐値は、IGYCDOPT プログラムで設定されます。

IGYCDOPT は、インストール時に AMODE 31 および RMODE ANY でリンク・エディットされます。

IGYCDOPT プログラムで設定されるコンパイラー・オプションのデフォルト値およびコンパイラー・フェーズの常駐値については、4 ページの表 1 および 12 ページの表 2 をそれぞれ参照してください。

IGYCDOPT のような COBOL カスタマイズ・パーツをアセンブルするときには、システム MACLIB にアクセスする必要があります。通常、MACLIB は SYS1.MACLIB にあります。COBOL MACLIB IGY.V6R1M0.SIGYMAC へのアクセスも必要です。

IGYCDOPT プログラムには 2 つの目的があります。1 つはコンパイラー・オプションのデフォルトを選択して固定することであり、もう 1 つはどのコンパイラー・フェーズが共有ストレージにあるかを指定することです。Enterprise COBOL のインストール時の IBM 提供のコンパイラー・オプション値をそのまま受け入れることも、お客様のシステムを担当するプログラマーの要件により適合するように値を変更することもできます。また、アプリケーション・プログラマーがこれらのオプションをオーバーライドできるようにするかどうかを選択することもできます。

注：高位修飾子 IGY.V6R1M0 は、Enterprise COBOL がインストールされたときに変更されている場合があります。

共有システム・ストレージに置かれるコンパイラー・フェーズを指定すれば、コンパイラーは作業域用領域内のストレージを使用することができます。コンパイラー・フェーズのデフォルト値を変更する目的の詳細については、6 ページの『コンパイラー・フェーズを共有ストレージに置く目的』を参照してください。

コンパイラー・オプションの固定

以下のセクションでは、コンパイラー・オプションを固定する理由、コンパイラー・オプションを固定する方法、および固定されたオプションを迂回する方法を説明します。

Enterprise COBOL では、お客様に固有のプログラミング標準をセットアップできます。例えば、多くのお客様では RENT を優先コンパイラー・オプションとして設定し、強制的に使用する必要がある場合があります。

Enterprise COBOL では、IGYCDOPT プログラムを使用してオプションを固定し、コンパイル時にそのオプションが変更またはオーバーライドされないように指定します。これにより、コンパイル時に、固定されたオプションをオーバーライドしようとする、ゼロ以外のコンパイラー戻りコードで診断メッセージが出されます。

一貫して使用するために特定のオプションを固定した場合、特別な条件のために、固定されたオプションを迂回する必要があることがあります。このような変更を行うには、別のパラメーターを指定して、IGYCDOPT プログラムの一時コピーをア

サンプルします。プログラマーはコンパイル時に、必要な IGYCDOPT モジュールを含む JOBLIB または STEPLIB を使用して、固定されたオプションを迂回することができます。

例えば、OPT=1 オプションを固定する (つまり、COBOL コンパイラーで常に最適化されたオブジェクト・コードを生成する) ことを選択し、あるアプリケーションをこの要件から除外する必要がある場合は、このオプションからアスタリスク・パラメーターを除去後に IGYCDOPT プログラムを再アセンブルしなければなりません。次に、アセンブルした IGYCDOPT モジュールを一時ライブラリーに入れ、コンパイル時に JOBLIB または STEPLIB としてアクセスされるようにします。

サンプル・インストール・ジョブ

Enterprise COBOL には 2 つのサンプル・インストール・ジョブがあり、それらを修正して使用すれば、コンパイラー・オプションのデフォルト値を変更することができます。一方のサンプル・ジョブは、IBM 提供のコンパイラー・オプションのデフォルト値を変更する例です。もう一方のサンプル・ジョブは、固定されているコンパイラー・オプションをオーバーライドする例です。

IGYWDOPT

このサンプル・インストール・ジョブを使用すれば、SMP/E を使用して IBM 提供のデフォルトを変更することができます。

IGYWUOPT

IGYCDOPT プログラムによって固定されたコンパイラー・オプションをオーバーライドする必要性が生じた場合、このサンプル・インストール・ジョブを使用すれば、SMP/E の外側にモジュールを作成し、そのモジュール内で異なるデフォルトを指定することができます。

1

これらのジョブは COBOL サンプル・データ・セット IGY.V6R1M0.SIGYSAMP に入っています。

コンパイラー・オプションおよびコンパイラー・フェーズの変更

コンパイラー・オプションおよびコンパイラー・フェーズの値を変更する場合は、IGYCOPT の構文形式を使用してください。

IBM 提供のデフォルト値は、計画ワークシートに示されているほか、各構文図のすぐ後にも示されています。構文図には、ix ページの『構文図の読み方』で説明しているようにデフォルトも示しています。

コンパイラー・オプション、コンパイラー・フェーズ、およびこれらのデフォルトについて、以下に説明します。これらのオプション、フェーズ、およびデフォルト値をよく検討して、アプリケーションに最も適する値を決定してください。

IGYCOPT の形式

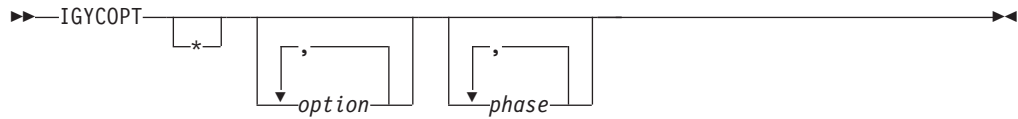


図1. IGYCOPT コンパイラー・オプション / フェーズ・マクロの構文形式

コンパイラー・オプション用の IGYCDOPT ワークシート

IGYCDOPT ワークシートは、IGYCDOPT プログラムのコンパイラー・オプション部分を計画し、コーディングする際に役立ちます。

「固定の場合 * を記入」欄と「選択を記入」欄に記入して、このワークシートを完成させてください。

IGYCDOPT ワークシートには、コンパイラー・フェーズのための部分もあります。ワークシートのその部分は、コンパイラー・フェーズの説明の後に記載されています (11 ページの『コンパイラー・フェーズ用の IGYCDOPT ワークシート』を参照)。

注:

- コンパイラー・オプションのデフォルト値を変更するときにアスタリスク [*] をコーディングすると、このオプションは固定され、アプリケーション・プログラマーがこのオプションをオーバーライドできなくなります。
- ALLOWCBL、DBCSXREF、および NUMCLS オプションは、コンパイル時にオーバーライドすることはできません。したがって、ワークシートでは、これらのオプションについては、「固定の場合 * を記入」欄はブランクになります。
- ADEXIT、INEXIT、LIBEXIT、MSGEXIT、および PRTEXIT の IBM 提供のデフォルト値はヌルです。したがって、これらのオプションについては、「IBM 提供のデフォルト」欄はブランクになっています。
- DUMP コンパイラー・オプションは、IGYCDOPT プログラムでは設定できません。コンパイル時に変更しない限り、DUMP は必ず NODUMP に設定されます。
- OPTFILE コンパイラー・オプションは、IGYCDOPT プログラムでは設定できません。

表1. コンパイラー・オプション用の IGYCDOPT ワークシート

コンパイラー オプション	固定の 場合 * 固定	選択を記入	IBM 提供の デフォルト	構文 説明
ADATA=	___	_____	<u>NO</u>	19 ページの『ADATA』
ADEXIT=	___	_____	_____	19 ページの『ADEXIT』
ADV=	___	_____	<u>YES</u>	20 ページの『ADV』
AFP=	___	_____	<u>VOLATILE</u>	21 ページの『AFP』
ALLOWCBL=	___	_____	<u>YES</u>	21 ページの『ALLOWCBL』
ARCH=	___	_____	<u>7</u>	22 ページの『ARCH』

表1. コンパイラー・オプション用の IGYCDOPT ワークシート (続き)

コンパイラー オプション	固定の 場合 * 固定	選択を記入	IBM 提供の デフォルト	構文 説明
ARITH=	_____	_____	<u>COMPAT</u>	24 ページの『ARITH』
AWO=	_____	_____	<u>NO</u>	24 ページの『AWO』
BLOCK0=	_____	_____	<u>NO</u>	25 ページの『BLOCK0』
BUF=	_____	_____	<u>4K</u>	25 ページの『BUF』
CICS=	_____	_____	<u>NO</u>	26 ページの『CICS』
CODEPAGE=	_____	_____	<u>1140</u>	26 ページの『CODEPAGE』
COMPILE=	_____	_____	<u>NOC(S)</u>	27 ページの『COMPILE』
COPYRIGHT=	_____	_____	<u>NO</u>	28 ページの『COPYRIGHT』
CURRENCY=	_____	_____	<u>NO</u>	28 ページの『CURRENCY』
DATA=	_____	_____	<u>31</u>	30 ページの『DATA』
DBCS=	_____	_____	<u>Yes</u>	31 ページの『DBCS』
DBCSXREF	_____	_____	<u>NO</u>	31 ページの『DBCSXREF』
DECK=	_____	_____	<u>NO</u>	32 ページの『DECK』
DIAGTRUNC=	_____	_____	<u>NO</u>	32 ページの『DIAGTRUNC』
DISPSIGN=	_____	_____	<u>COMPAT</u>	33 ページの『DISPSIGN』
DLL=	_____	_____	<u>NO</u>	34 ページの『DLL』
DYNAM=	_____	_____	<u>NO</u>	34 ページの『DYNAM』
EXPORTALL=	_____	_____	<u>NO</u>	35 ページの『EXPORTALL』
FASTSRT=	_____	_____	<u>NO</u>	36 ページの『FASTSRT』
FLAG=	_____	_____	<u>(L)</u>	37 ページの『FLAG』
FLAGSTD=	_____	_____	<u>NO</u>	37 ページの『FLAGSTD』
HGPR=	_____	_____	<u>PRESERVE</u>	40 ページの『HGPR』
INEXIT=	_____	_____		40 ページの『INEXIT』
INTDATE=	_____	_____	<u>ANSI</u>	41 ページの『INTDATE』
LANGUAGE=	_____	_____	<u>EN</u>	42 ページの『LANGUAGE』
LIBEXIT=	_____	_____		42 ページの『LIBEXIT』
LINECNT=	_____	_____	<u>60</u>	43 ページの『LINECNT』
LIST=	_____	_____	<u>NO</u>	43 ページの『LIST』
LITCHAR=	_____	_____	<u>QUOTE</u>	44 ページの『LITCHAR』
MAP=	_____	_____	<u>NO</u>	45 ページの『MAP』
MAXPCF=	_____	_____	<u>60000</u>	45 ページの『MAXPCF』
MDECK=	_____	_____	<u>NO</u>	47 ページの『MDECK』
MSGEXIT=	_____	_____		47 ページの『MSGEXIT』
NAME=	_____	_____	<u>NO</u>	48 ページの『NAME』
NSYMBOL=	_____	_____	<u>NATIONAL</u>	48 ページの『NSYMBOL』
NUM=	_____	_____	<u>NO</u>	49 ページの『NUM』
NUMCLS=	_____	_____	<u>PRIM</u>	49 ページの『NUMCLS』
NUMPROC=	_____	_____	<u>NOPFD</u>	50 ページの『NUMPROC』
OBJECT=	_____	_____	<u>YES</u>	51 ページの『OBJECT』
OFFSET=	_____	_____	<u>NO</u>	51 ページの『OFFSET』
OPTIMIZE=	_____	_____	<u>0</u>	52 ページの『OPTIMIZE』
OUTDD=	_____	_____	<u>SYSDOUT</u>	53 ページの『OUTDD』
PGMNAME=	_____	_____	<u>COMPAT</u>	53 ページの『PGMNAME』
PRTEXIT=	_____	_____		54 ページの『PRTEXIT』
QUALIFY=	_____	_____	<u>COMPAT</u>	55 ページの『QUALIFY』
RENT=	_____	_____	<u>YES</u>	56 ページの『RENT』
RMODE=	_____	_____	<u>AUTO</u>	57 ページの『RMODE』
RULES=	_____	_____	<u>NO</u>	58 ページの『RULES』
SEQ=	_____	_____	<u>YES</u>	59 ページの『SEQ』
SERVICE=	_____	_____	<u>NO</u>	59 ページの『SERVICE』
SOURCE=	_____	_____	<u>YES</u>	60 ページの『SOURCE』

表1. コンパイラー・オプション用の IGYCDOPT ワークシート (続き)

コンパイラー オプション	固定の 場合 *	選択を記入	IBM 提供の デフォルト	構文 説明
SPACE=	_____	_____	<u>1</u>	60 ページの『SPACE』
SQL=	_____	_____	<u>NO</u>	60 ページの『SQL』
SQLCCSID=	_____	_____	<u>YES</u>	61 ページの『SQLCCSID』
SQLIMS=	_____	_____	<u>NO</u>	62 ページの『SQLIMS』
SSRANGE=	_____	_____	<u>NO</u>	63 ページの『SSRANGE』
STGOPT=	_____	_____	<u>NO</u>	64 ページの『STGOPT』
SUPPRESS=	_____	_____	<u>YES</u>	64 ページの『SUPPRESS』
TERM=	_____	_____	<u>NO</u>	65 ページの『TERM』
TEST=	_____	_____	<u>(NO, NODWARF)</u>	65 ページの『TEST』
THREAD=	_____	_____	<u>NO</u>	66 ページの『THREAD』
TRUNC=	_____	_____	<u>STD</u>	68 ページの『TRUNC』
VBREF=	_____	_____	<u>NO</u>	69 ページの『VBREF』
VLR=	_____	_____	<u>COMPAT</u>	70 ページの『VLR』
VSAMOPENFS=	_____	_____	<u>COMPAT</u>	71 ページの 『VSAMOPENFS』
WORD=	_____	_____	<u>NO</u>	71 ページの『WORD』
XMLPARSE=	_____	_____	<u>XMLSS</u>	72 ページの『XMLPARSE』
XREFOPT=	_____	_____	<u>FULL</u>	73 ページの『XREFOPT』
ZONECHECK=	_____	_____	<u>NO</u>	74 ページの 『ZONECHECK』
ZONEDATA=	_____	_____	<u>PFD</u>	75 ページの『ZONEDATA』
ZWB=	_____	_____	<u>YES</u>	76 ページの『ZWB』

コンパイラー・フェーズを共有ストレージに置く計画

以下のセクションでは、共有ストレージを定義し、各種コンパイラー・フェーズについて説明し、コンパイラー・フェーズを共有ストレージに置く理由を説明します。

Enterprise COBOL プログラムが実行される時、またはプログラム・オブジェクトが共有される時にプログラム・オブジェクトの検索を最小限に押さえるために、いくつかのプログラム・オブジェクトをリンク・バック域に常駐させることをお勧めします。さらに、一部または全部のコンパイラー・フェーズを常駐させることもできます。

共有ストレージ という用語は一般に、リンク・バック域 (LPA)、拡張リンク・バック域 (ELPA)、または修正リンク・バック域 (MLPA) を説明するために使用されます。本書では、特に断りのない限り、リンク・バック域 という用語はこれら 3 つすべてを指します。

コンパイラー・フェーズを共有ストレージに置く目的

このセクションでは、共有ストレージを定義し、共有ストレージに置くことができるコンパイラー・フェーズと、コンパイラー・フェーズを共有ストレージに置く理由について説明します。

共有ストレージは、各仮想アドレス・スペースで同じである、ストレージの領域です。これはすべてのユーザーにとって同じスペースであるため、そこに保管されている情報をユーザー領域にロードする必要はありません。情報を共有することによって、より多くのスペースがコンパイラ作業域のために使用可能になります。

ダンプ実行モジュール (IGYCRDPR および IGYCRDSC) および予約語ユーティリティー (IGY8RWTU) 以外のすべてのコンパイラ・モジュールは、z/OS マシンの共有ストレージに置くことができます。IGYCRCTL と IGYCSIMD を除くすべてのコンパイラ・フェーズには RMODE ANY および AMODE 31 があります。IGYCRCTL および IGYCSIMD は RMODE 24 であるため、これらを、LPA または MLPA に置くことができますが、ELPA に置くことはできません。

注: Enterprise COBOL、COBOL for MVS™ & VM、COBOL for OS/390® & VM、および VS COBOL II のコンパイラには、同じ名前のモジュールがあります。そのため、オペレーティング・システムの初期化のときに LPA に配置できるのは、1つのフェーズ・セットのみです。例外として、LPALST を使用して Enterprise COBOL V5 および V6 モジュールを共有ストレージに配置することはできません。コンパイラ・モジュールが PDSE データ・セットにあるため、IPL の実行時にモジュール名を LPALST 連結に組み込むことができません。IPL の終了時に動的にモジュールを LPA に追加することはできません。

IGYCDOPT プログラムは、各コンパイラ・フェーズ (フェーズ IGYCBE、IGYECWI、IGYMSGK、および IGYMSGE を除く) がどこにロードされるか (ユーザー領域の内側 (IN) か外側 (OUT) か) を指示します。コンパイラ・フェーズを MLPA に置くと、コンパイラは、ユーザーのプログラムのために、より多くのストレージを使用できます。

あるフェーズをユーザー領域に置かないことを指定した場合は、そのフェーズを実際に共有ストレージに置く必要があります。コンパイラは、この情報を用いて、システムがコンパイラ・フェーズをユーザー領域にロードするためのストレージの量を決定します。

フェーズを共有ストレージに置く方法については、97 ページの『関連資料』のリスト中の「初期設定およびチューニング」の解説書を参照してください。

以下のフェーズを共有ストレージ域に置くことをお勧めします。

IGYCRCTL

コンパイルの初めから終わりまでユーザー領域に常駐するため。

IGYCSIMD

コンパイルの初めから終わりまでユーザー領域に常駐するため。

IGYCBE

最大のコンパイラ・フェーズであるため。

IGYECWI

2 番目に大きいコンパイラ・フェーズであるため。

IGYCPGEN

3 番目に大きいコンパイラ・フェーズであるため。

IGYCSKAN

4 番目に大きいコンパイラ・フェーズであるため。

どのコンパイラー・フェーズを共有ストレージに置くかは、同時に使用する頻度と、そのサイズに基づいて、判断することができます。システムでコンパイラーをほとんど使用しないなら、フェーズを共有ストレージに置く利点はありません。一方、コンパイルを頻繁に行い、かつ十分な MLPA ストレージが使用可能な場合、コンパイラー全体を常駐させることは効果的といえます。使用可能な共有ストレージが十分でない場合は、コンパイル時にユーザー領域に必ずある 2 つのフェーズ IGYCRCTL および IGYCSIMD を優先させる必要があります。さらに、最大コンパイラー・フェーズである IGYCBE も優先させる必要があります。

共有ストレージにコンパイラー・フェーズを置く別の利点として、コンパイル時の初期化論理で、共有ストレージに常駐していないフェーズで最大のものを十分収容できる記憶ブロックを、ユーザー領域に割り振ることができます。どのようなユーザー領域のサイズに対してもスペース割り振りを最小限にすれば、コンパイル・プロセス用により多くのスペースを使用でき（つまり、ユーザー領域でより大きなプログラムをコンパイルでき）、より効率的にコンパイルを行うことができます。コンパイラー・フェーズ IGYCBE および IGYECWI は、次に大きいコンパイラー・フェーズより約 10 MB 大きいコンパイラー・フェーズです。共有ストレージにコンパイラー・フェーズを置くと、最小領域サイズを使用してコンパイルする場合には、かなりの効果があります。

コンパイラー・フェーズおよびそのデフォルト

このセクションでは、コンパイラー・フェーズのロード先を指定する方法と、フェーズについて説明します。また、フェーズに対する変更を計画する際に役立つワークシートも用意されています。

ユーザー領域に関して、各コンパイラー・フェーズをどこにロードするかを示すには、IN または OUT のいずれかを指定します。

これらのデフォルトを変更する理由、または変更しない理由については、6 ページの『コンパイラー・フェーズを共有ストレージに置く目的』を参照してください。

IN コンパイラー・フェーズを、コンパイル時に使用可能なライブラリーからユーザー領域にロードすることを指定します。

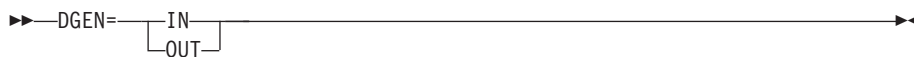
あるコンパイラー・フェーズについて IN を指定した場合でも、そのフェーズを共有システム域に置くことができます。ただし、コンパイラー制御フェーズは、コンパイラー・フェーズ用に予約された主記憶域が、IN が指定された最大のフェーズを収容するために十分な大きさであることを保証します。このオプションを指定した場合、ストレージの一部が使用されなくなります。

OUT コンパイラー・フェーズをライブラリーからユーザー領域にロードしないことを指定します。したがって、コンパイラー・フェーズは MLPA などの共有システム域に置かれている必要があります。

IGYCDGEN

データ生成フェーズです。プログラム内のデータ項目を記述する適切な WCode 命令が生成されます。

構文



IGYCDIAG

診断フェーズです。診断フェーズでは、E 形式のテキストを処理し、ソース・プログラム・エラーに関するコンパイラ診断情報を生成します。これは、IGYCDIAG およびメッセージ・モジュール IGYCxx\$D、IGYCxx\$1、IGYCxx\$2、IGYCxx\$3、IGYCxx\$4、IGYCxx\$5、および IGYCxx\$8 (xx は EN、UE、または JA です) を含んでいます。

構文



IGYCDMAP

DMAP フェーズです。DMAP フェーズでは、MAP オプションによって要求された出力のテキストを準備します。

構文



IGYCFGEN

ファイル生成フェーズです。ファイル生成フェーズでは、プログラムで定義されている FD および SD 用の制御ブロックを生成します。

構文



IGYCINIT

初期化フェーズです。初期化フェーズでは、処理フェーズの実行を準備するためにハウスキーピングを行います。

構文



IGYCLIBR

COPY フェーズです。COPY フェーズでは、ライブラリー・ソース・テキストを処理し、COPY ステートメント、BASIS ステートメント、および REPLACE ステートメントの構文チェックを行います。

構文



IGYCLSTR

ソース・リスト・フェーズです。ソース・リスト・フェーズでは、相互参照情報と診断情報が組み込まれているソース・リストを出力します。

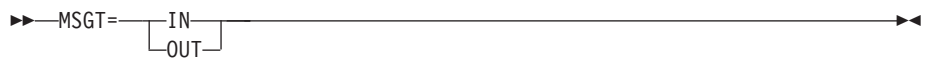
構文



IGYCMSGT

ヘッダー・テキスト・テーブルおよび診断メッセージ・レベル・テーブルを示します。これは、モジュール IGYCxx\$R、IGYCLVL0、IGYCLVL1、IGYCLVL2、IGYCLVL3、および IGYCLVL8 (xx は EN、UE、または JA です) を含んでいます。

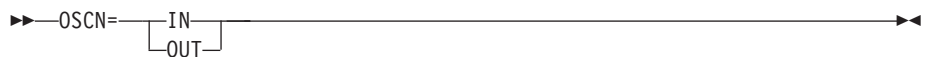
構文



IGYCOSCN

オプション・スキャン・フェーズです。オプション・スキャン・フェーズでは、デフォルト・オプションを判別し、EXEC PARM オプションを処理してから、PROCESS (CBL) ステートメントを処理します。

構文



IGYCPGEN

プロシージャ生成フェーズです。プロシージャ生成フェーズでは、すべてのプロシージャ・ソース動詞にコードを提供します。

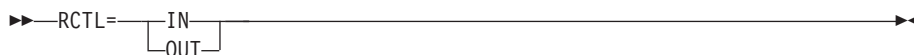
構文



IGYCRCTL

常駐制御フェーズです。常駐制御フェーズでは、コンパイラーの共通ストレージおよび作業用ストレージのサイズを設定し、プログラムの共通ストレージの初期化を実行します。

構文



IGYCRWT

標準の予約語テーブルです。

構文



IGYCSCAN

スキャン・フェーズです。スキャン・フェーズでは、ソース・プログラムの構文およびセマンティックを分析し、ソースを中間テキストに変換します。

構文



IGYCSIMD

Enterprise COBOL コンパイラー用のシステム・インターフェース・フェーズです。他のすべてのコンパイラー・フェーズが、このフェーズを呼び出して、システム依存機能を実行します。

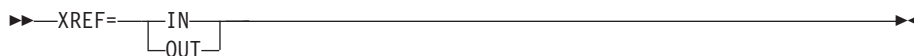
構文



IGYCXREF

XREF フェーズです。XREF フェーズでは、ユーザー名およびプロシージャ名を EBCDIC 照合シーケンスでソートします。

構文



コンパイラー・フェーズ用の IGYCDOPT ワークシート

IGYCDOPT ワークシートは、IGYCDOPT プログラムのフェーズ部分を計画し、コーディングする際に役立ちます。

以下のワークシートでは、各フェーズに割り当てる値を丸で囲んでください。各フェーズに割り当てることができる値については、8 ページの『コンパイラー・フェーズおよびそのデフォルト』を参照してください。

注: フェーズ・デフォルトはすべて、最初は IN に設定されています。

表2. コンパイラ・フェーズ用の IGYCDOPT プログラム・ワークシート

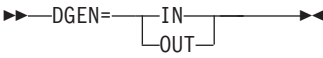
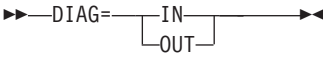
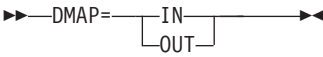

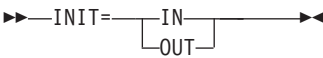
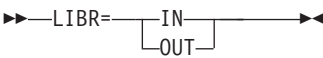
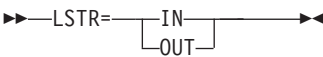
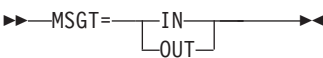
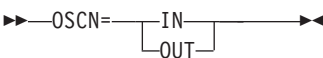
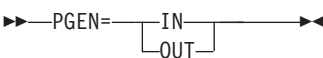
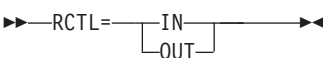
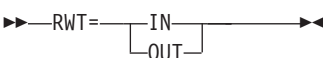
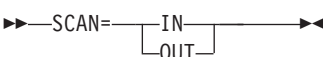
フェーズ	選択 (丸で囲む)	構文の説明
DGEN=	<u>IN</u> / OUT	▶▶-DGEN=  ▶▶
DIAG=	<u>IN</u> / OUT	▶▶-DIAG=  ▶▶
DMAP=	<u>IN</u> / OUT	▶▶-DMAP=  ▶▶
FGEN=	<u>IN</u> / OUT	▶▶-FGEN=  ▶▶
INIT=	<u>IN</u> / OUT	▶▶-INIT=  ▶▶
LIBR=	<u>IN</u> / OUT	▶▶-LIBR=  ▶▶
LSTR=	<u>IN</u> / OUT	▶▶-LSTR=  ▶▶
MSGT=	<u>IN</u> / OUT	▶▶-MSGT=  ▶▶
OSCN=	<u>IN</u> / OUT	▶▶-OSCN=  ▶▶
PGEN=	<u>IN</u> / OUT	▶▶-PGEN=  ▶▶
RCTL=	<u>IN</u> / OUT	▶▶-RCTL=  ▶▶
RWT=	<u>IN</u> / OUT	▶▶-RWT=  ▶▶
SCAN=	<u>IN</u> / OUT	▶▶-SCAN=  ▶▶

表 2. コンパイラー・フェーズ用の IGYCDOPT プログラム・ワークシート (続き)

フェーズ	選択 (丸で囲む)	構文の説明
SIMD=	<u>IN</u> / OUT	
XREF=	<u>IN</u> / OUT	

追加の予約語テーブルを作成する計画

以下のセクションでは、追加の予約語テーブルを作成する理由と、予約語テーブルを変更することで、ネストされたプログラムの使用を制限する方法について説明します。また、Enterprise COBOL で提供される予約語テーブルを掲載しています。

インストール後に、追加の予約語テーブルを作成することができます。コンパイル時に、WORD コンパイラー・オプションの値で、どの予約語テーブルを使用するかを決定します。

追加の予約語テーブルを作成する目的

このセクションでは、追加の予約語テーブルを作成する利点について説明します。

以下の目的のために、追加の予約語テーブルを作成することができます。

- 予約語を別の言語 (フランス語またはドイツ語など) に変換する。
- アプリケーション・プログラマーが Enterprise COBOL の特定の命令 (GO TO など) を使用するのを防ぐ。
- ネストされたプログラムの使用を制御する。
- CICS でサポートされていない語 (READ、WRITE など) にフラグを立てる。

ネストされたプログラムの使用の制御

他の COBOL 言語機能を制限せずに、ネストされたプログラムの使用を制限するには、予約語テーブルを変更します。

これは、INFO および RSTR 制御ステートメントを使用して行います。これらの変更を加える方法については、84 ページの『予約語テーブルの作成または変更』を参照してください。

Enterprise COBOL と一緒に提供される予約語テーブル

Enterprise COBOL では、インストール・メディアで予約語テーブルが提供されません。

提供される予約語テーブルは以下のとおりです。

- デフォルト予約語テーブル
- CICS 予約語テーブル

デフォルト予約語テーブル (IGYCRWT)

機能全体に対して提供されるデフォルト予約語テーブルについては、「Enterprise COBOL 言語解説書」を参照してください。

CICS 予約語テーブル (IGYCCICS)

Enterprise COBOL では、CICS アプリケーション・プログラム用の代替予約語テーブルが提供されるため、CICS ではサポートされていない COBOL 語にはコンパイラーによってフラグが立てられます。

CICS 予約語テーブルは、以下の COBOL 語が制限付き (RSTR) としてマークされている以外は、デフォルト予約語テーブルと同じです。

- CLOSE
- DELETE
- FD
- FILE
- FILE-CONTROL
- INPUT-OUTPUT
- I-O-CONTROL
- MERGE
- OPEN
- READ
- RERUN
- REWRITE
- SD
- SORT
- START
- WRITE

SORT ユーザー: Enterprise COBOL は、CICS のもとの SORT ステートメントのためのインターフェースをサポートしています。CICS のもとの SORT ステートメントを使用したい場合は、その使用前に CICS 予約語テーブルを修正する必要があります。上記の下線が引かれた語は、SORT 機能のために必要なため、制限付きとしてマークされた語のリストから削除する必要があります。

テーブルの使用:

CICS 予約語テーブルを使用するには、WORD(CICS) コンパイラー・オプションを指定してください。

CICS 予約語テーブルをデフォルトとして使用するには、WORD コンパイラー・オプションのデフォルト値を WORD=CICS に設定してください。

テーブルの場所:

CICS 予約語テーブルの作成に使用されるデータは、IGY.V6R1M0.SIGYSAMP 内のメンバー IGY8CICS に入っています。

| 注: 高位修飾子 IGY.V6R1M0 は、Enterprise COBOL がインストールされたときに
| 変更されている場合があります。

製品登録を使用して Enterprise COBOL を使用可能/使用不可にする

| Enterprise COBOL V6 はデフォルトの動作では各 z/OS システムで実行されます
| が、SYSx.PARMLIB の IFAPRDxx メンバーを使用すれば、選択した z/OS システ
| ムで COBOL V6 が実行されないようにすることができます。

| COBOL V6 を使用不可にしたい場合は、以下のコードをアクティブな IFAPRDxx
| メンバーに追加します。

```
PRODUCT OWNER('IBM CORP')  
      NAME('ENTERPRISE COBOL')  
      ID(5655-EC6)  
      VERSION(06) RELEASE(*) MOD(*)  
      FEATURENAME(*)  
      STATE(DISABLED)
```

この場合、コンパイラーは RC=16 で停止し、オペレーター宛メッセージを出力し
ます。

| COBOL V6 を明示的に使用可能にしたい場合は、以下のコードをアクティブな
| IFAPRDxx メンバーに追加します。

```
PRODUCT OWNER('IBM CORP')  
      NAME('ENTERPRISE COBOL')  
      ID(5655-EC6)  
      VERSION(06) RELEASE(*) MOD(*)  
      FEATURENAME(*)  
      STATE(ENABLED)
```

| ただし、COBOL V6 はデフォルトで使用可能になっているため、上記のステートメ
| ントをアクティブな IFAPRDxx メンバーに追加して COBOL V6 を明示的に使用可
| 能にする必要はありません。

第 2 章 Enterprise COBOL コンパイラー・オプション

ここでは、デフォルト値を変更できるコンパイラー・オプションについて説明します。

いくつかの説明に付記されている注は、コンパイル時の他のオプションとの相互影響など、これらのオプションについての追加情報です。

これらの情報は、ご使用のシステムに適切なデフォルト値を決定するために役立ちます。

コンパイラー・オプションについて詳しくは、「Enterprise COBOL プログラミング・ガイド」の『コンパイラー・オプション』を参照してください。

重要:

Enterprise COBOL のカスタマイズを計画する際には、この製品を使用するアプリケーション・プログラマーと相談してください。そうすることにより、アプリケーション・プログラマーのニーズを満たし、開発されるアプリケーションをサポートするような変更を行うことができます。

COBOL コンパイラー・オプションの指定

IGYCOPT マクロにコンパイラー・オプションを指定するときは、オプション名とその値の両方を大文字で指定する必要があります。

オプション名を大文字で指定しなければ、オプション名とその値の両方が無視され、デフォルト値が使用されます。エラー・メッセージは出されません。オプション値だけが大きい場合は、無効なオプション値が指定されていることを示すエラー・メッセージが出されます。

矛盾するコンパイラー・オプション

特定のコンパイラー・オプション値を指定すると、他のコンパイラー・オプションとの矛盾が生じる場合があります。このトピックでは、コンパイラー・オプション間に生じる可能性がある矛盾について説明します。

表 3. 矛盾するコンパイラー・オプション

コンパイラー・オプション	矛盾するオプション
CICS=YES	RENT=NO DYNAM=YES
DBCS=NO	NSYMBOL=NATIONAL
DBCSXREF=(NO 以外)	XREFOPT=NO
DLL=NO	EXPORTALL=YES

表3. 矛盾するコンパイラー・オプション (続き)

コンパイラー・オプション	矛盾するオプション
DLL=YES	DYNAM=YES RENT=NO
DYNAM=YES	CICS=YES DLL=YES EXPORTALL=YES
EXPORTALL=YES	DLL=NO DYNAM=YES RENT=NO
FLAGSTD=(NO 以外)	WORD=xxxx
LIST=YES	OFFSET=YES
NSYMBOL=NATIONAL	DBCS=NO
OBJECT=NO	TEST=(NO 以外)
OFFSET=YES	LIST=YES
RENT=NO	CICS=YES DLL=YES EXPORTALL=YES THREAD=YES
THREAD=YES	RENT=NO
WORD=xxxx	FLAGSTD=(NO 以外)
XREFOPT=NO	DBCSXREF=(NO 以外)

標準準拠のためのコンパイラー・オプション

85 COBOL 標準に準拠するには、いくつかのコンパイラー・オプションが必要です。

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『85 COBOL 標準に準拠するオプション設定』を参照してください。

コンパイラー・オプションの構文および説明

以下のトピックにある構文図は、変更可能な各コンパイラー・オプションを説明しています。各構文図の下にあるテキストは、特定のパラメーターを選択した場合の結果を説明しています。

注:

- DUMP オプションはこのリストに含まれていません。コンパイル時に DUMP を変更しない限り、常に NODUMP に設定されます。このオプションは、通常は使用しません。IBM 担当員から要求されたときにのみ使用してください。
- OPTFILE オプションはこのリストに含まれていません。これはコンパイラー呼び出し PARM オプションとして、または COBOL ソース・プログラム内の PROCESS または CBL ステートメントでのみ指定できます。
- コンパイラー・オプションのデフォルト値を変更するときにアスタリスク (*) をコーディングすると、このオプションは固定され、アプリケーション・プログラマーがこのオプションをオーバーライドできなくなります。

ADATA

ADATA は、コンパイル時に関連データ・ファイルを作成するかどうかに作用します。

構文



デフォルト

ADATA=NO

YES

適切なレコードを使用して関連データ・ファイルを作成します。

NO 関連データ・ファイルを作成しません。

注:

- ADATA オプションの指定は、呼び出し時にオプション・リストを介してか、JCL の PARM フィールド上か、コマンド・オプションとしてか、またはインストール・システム・デフォルトとしてか、のいずれかの場合のみ可能です。
- 日本語オプションを選択すると、関連データ・ファイル内のレコードに DBCS 文字が書き込まれることがあります。
- NOCOMPILE(WIEIS) を指定した場合、コンパイルが途中で停止し、特定の関連データ・レコードが消失することがあります。
- INEXIT オプションを使用すると、コンパイル・ソース・モジュールは SYSADATA (関連データ・ファイル) 情報内で識別されません。

ADEXIT

ADEXIT は、SYSADATA ファイルに書き込まれるレコードごとに呼び出すモジュールを指定します。

構文



デフォルト

出口は指定されません。EXIT コンパイラー・オプションの NOADEXIT サブオプションを指定するのと同様です。ADEXIT=* が *name* パラメーターなしでコーディングされた場合、NOADEXIT をオーバーライドすることはできません。

name

EXIT コンパイラー・オプションで使用するモジュールを識別します。このユーザー出口のサブオプションが指定されている場合、コンパイラーは指定されたモジュールをロードし、SYSADATA ファイルに書き込まれるレコードごとにそのモジュールを呼び出します。

EXIT コンパイラー・オプションについては、「Enterprise COBOL プログラミング・ガイド」の『EXIT コンパイラー・オプション』を参照してください。

ADV

ADV は、WRITE ... ADVANCING ステートメントに作用し、プリンター制御文字のレコード長に 1 バイト追加するかどうかを決定します。

構文



デフォルト

ADV=YES

YES

印刷制御文字用の 1 バイトをレコード長に追加します。このオプションは、ソース・ファイル内で WRITE ... ADVANCING を使用するプログラマーに有用です。レコードの先頭文字をプログラマーが明示的に予約する必要はありません。

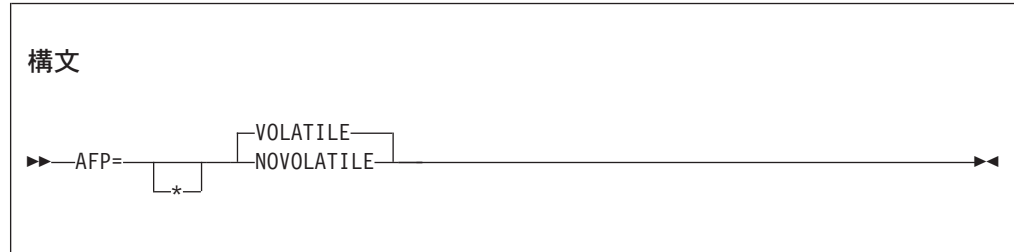
NO レコード長を WRITE ... ADVANCING 用に調整しません。コンパイラーは、指定されたレコード域の先頭文字を用いて印刷制御文字を置きます。アプリケーション・プログラマーは、レコード記述にこの追加のバイトが見込まれていることを確認する必要があります。

注:

- ADV=YES を指定した場合、物理装置上のレコード長は、ソース・プログラムにおけるレコード記述長よりも 1 バイト大きくなります。
- 出力ファイルのレコード長がソース・コードで定義されていない場合は、COBOL は、DCB パラメーターが適切に設定されていることを確認します。
- ADV=YES が指定され、かつ出力ファイルのレコード長がソース・コードで定義された場合は、プログラマーは、レコード記述長をソース・プログラムのレコード記述よりも 1 バイト大きいものとして指定する必要があります。さらに、プログラマーは、1 バイト大きいレコード・サイズの正確な倍数でブロック・サイズを指定する必要があります。
- ファイル記述 (FD) に LINAGE 節が指定されると、コンパイラーは、ADV=YES が指定されているものとしてそのファイルを処理します。

AFP

AFP オプションは、z/Architecture[®] プロセッサに備わっている追加浮動小数点 (AFP) レジスタのコンパイラによる使用を制御します。



デフォルト: AFP=VOLATILE

Enterprise COBOL コンパイラは、z/Architecture プロセッサに備わっている 16 個の浮動小数点レジスタ (FPR) をすべて使用するコードを生成します。これらの FPR には以下があります。

- オリジナル FPR。 0、2、4、および 6 の番号が付けられています。
- AFP レジスタ。 1、3、5、7、および 8 から 15 の番号が付けられています。

注: ご使用のコードが CICS Transaction Server V4.1 より前のバージョンで実行される場合は、AFP=VOLATILE を指定する必要があります。

AFP=VOLATILE

AFP=VOLATILE を指定した場合、8 から 15 までの AFP レジスタは揮発性のレジスタであるとみなされます。つまり、呼び出されたサブプログラムがこれらのレジスタを変更する可能性があります。このため、COBOL コンパイラはこれらのレジスタの値を保護するために、追加コードを生成します。

AFP=NOVOLATILE

AFP=NOVOLATILE を指定した場合、8 から 15 までの AFP レジスタは不揮発性のレジスタであるとみなされます。つまり、これらのレジスタは、呼び出された各サブプログラムによって変更されないもの、または保持されるものとみなされます。このため、コンパイラは浮動小数点演算が含まれるプログラムに対して、より効率的なコード・シーケンスを生成できます。これは通常の z/OS アーキテクチャーの規則です。

ALLOWCBL

ALLOWCBL は、COBOL プログラムで PROCESS (または CBL) ステートメントを使用できるかどうか作用します。

構文



デフォルト

ALLOWCBL=YES

YES

COBOL プログラムでの PROCESS (または CBL) ステートメントの使用を許可します。

NO プログラムで PROCESS (または CBL) ステートメントが使用された場合は、エラーと診断します。

注:

- ALLOWCBL は、コンパイル時にオーバーライドすることはできません。ALLOWCBL を PROCESS (または CBL) ステートメントに組み込むことはできないからです。
- PROCESS (または CBL) ステートメントは、ソース・プログラム内でコンパイラー・オプション・パラメーターを指定するために使用されます。インストール要件により、ソース・プログラムでコンパイラー・オプションを指定することを許可しない場合は、ALLOWCBL=NO を指定してください。

ARCH

ARCH オプションは、実行可能プログラム命令の生成対象となるマシン・アーキテクチャーを指定します。



より高い ARCH レベルを指定すると、コンパイラーは、新しいより高速な命令を使用するコードを生成します。そのようなコードからなるアプリケーションは、ARCH オプションで指定されたアーキテクチャー・レベルより低いアーキテクチャー・レベルのプロセッサ上で実行されると異常終了する可能性があります。アプリケーションを実行する最もレベルの低いマシン・アーキテクチャーに合致する ARCH レベルを使用してください。

現在サポートされているアーキテクチャー・レベルおよびモデルのグループは以下のとおりです。

デフォルト

ARCH=7

- 7** z/Architecture モードの 2094-xxx (IBM System z9[®] EC) モデルおよび 2096-xxx (IBMSystem z9 BC) 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 ARCH(7) 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。

- 拡張即時 (Extended-immediate) 機能
- 10 進数浮動小数点機能。これらの命令は、10 進数データが数値演算で使用されている場合に生成される可能性があります。

8 z/Architecture モードの 2097-xxx (IBM System z10[®] EC) モデルおよび 2098-xxx (IBM System z10 BC) 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 ARCH(8) マシンおよびその後継マシンには、汎用命令拡張機能でサポートされる命令が追加されています。

9 z/Architecture モードの 2817-xxx (IBM zEnterprise[®] 196) モデルおよび 2818-xxx (IBM zEnterprise 114) モデルで使用可能な命令を使用するコードを生成します。

具体的には、このような ARCH=9 マシンとその後継マシンは、以下の機能でサポートされる命令を追加します。

- 上位ワード機能
- インターロック・アクセス機能
- 条件付きロード/ストア機能
- 独立オペランド機能
- ポピュレーション・カウント機能

10 z/Architecture モードの 2827-xxx (IBM zEnterprise EC12) モデルおよび 2828-xxx (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 ARCH(10) 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。

- 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。算術計算を実行するためにゾーン 10 進数データ項目をバック 10 進数データ項目に変換する代わりに、コンパイラーはゾーン 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-xxx (IBM z13[™]) モデルおよび 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 ARCH(11) 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。

- バック 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

- いくつかの INSPECT REPLACING ステートメントおよび INSPECT TALLYING ステートメントのための、新しいベクトル拡張機能 (SIMD) 命令の利用

注: より高い ARCH レベルには、低い ARCH レベルの機能が含まれています。例えば、ARCH=11 には、下位 ARCH レベルの機能がすべて含まれます。

これらの機能について詳しくは、「z/Architecture 解説書」を参照してください。

ARITH

ARITH は、整数にコーディングできる最大桁数、および固定小数点の中間結果で使用される桁数に作用します。

構文



デフォルト

ARITH=COMPAT

COMPAT

10 進データの最大精度として 18 桁を指定します。

EXTEND

10 進データの最大精度として 31 桁を指定します。

AWO

AWO は、可変長ブロック化フォーマットの物理順次ファイルに対して APPLY-WRITE-ONLY 節をアクティブにするかどうか作用します。

構文



デフォルト

AWO=NO

YES

APPLY-WRITE-ONLY 節がプログラムで指定されているかどうかに関係なく、プログラム内の可変長ブロック形式のすべての物理順次ファイルについて、APPLY-WRITE-ONLY 節をアクティブにします。

パフォーマンスの考慮: AWO=YES を使用すると、通常は、入出力処理時に実行時ファイルのためにデータ管理サービスを呼び出す回数が減ります。

NO APPLY-WRITE-ONLY 節がプログラムで指定されていない場合には、プログラム内の可変長ブロック形式のすべての物理順次ファイルについて、APPLY-WRITE-ONLY 節をアクティブにしません。

BLOCK0

BLOCK0 は、QSAM ファイルのデフォルト・ブロック化指定を非ブロック化からブロック化に変更するかどうかには作用します。

構文



デフォルト

BLOCK0=NO

YES

ファイル記述項目に BLOCK CONTAINS も RECORDING MODE U も指定されていない QSAM ファイルに対するデフォルトのブロック化仕様を変更します。BLOCK0=YES を指定すると、そのようなファイルに対して BLOCK CONTAINS 0 節がアクティブになり、その結果として、実行時に、システムが決定するブロック・サイズになります。

パフォーマンスの考慮: BLOCK0=YES を指定すると、処理速度が速くなり、QSAM 出力ファイルのストレージ要件を最小化することができます。ただし、以下の推奨事項を参照してください。

NO どのようなファイルについても、デフォルトで BLOCK CONTAINS 0 節をアクティブにしません。

推奨: 既存プログラム内のファイル記述に BLOCK CONTAINS 0 節を追加すると、INPUT としてオープンされるファイルに対して望ましくない影響が起こるなど、プログラムの動作が変わる可能性があります。この理由から、インストールのデフォルトとして BLOCK0=YES を設定しないことをお勧めします。

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『BLOCK0』を参照してください。

BUF

BUF は、コンパイル時に使用する動的ストレージの量を指定します。

構文



デフォルト

BUF=4K

integer

各コンパイラ作業ファイル・バッファに割り振られる動的ストレージの量をバイト単位で指定します。最小値は 256 バイトです。

パフォーマンスの考慮: 大容量のバッファを使用すると、コンパイラーの性能が向上します。

integerK

バッファに割り振られる動的ストレージの量を 1K (1024) バイト単位で指定します。

注:

- BUF は、使用する装置のトラック容量を超えてはならず、また、データ管理サービスで許可される最大量を超えてもなりません。

CICS

CICS は、CICS ステートメントを含む COBOL ソース・プログラムが、組み込みの CICS 変換プログラムによって処理されるかどうか作用します。

構文



デフォルト

CICS=NO

YES

COBOL ソース・プログラムに CICS ステートメントが含まれているが CICS 変換プログラムによってプリプロセスされていない場合は、YES オプションを指定する必要があります。

NO NO オプションを指定すると、ソース・プログラムで検出された CICS ステートメントはすべて診断され廃棄されます。

注:

- CICS コンパイラー・オプションには CICS サブオプションを含めることができます。CICS サブオプションの区切り文字として、引用符またはアポストロフィを使用することができます。CICS サブオプションを COBOL インストール時のデフォルトとして指定することはできません。
- CICS コンパイラー・オプションは、どのコンパイラー・オプション・ソースでも指定できます。すなわち、インストール時のデフォルト、コンパイラー呼び出し、PROCESS ステートメント、または CBL ステートメントのいずれかに指定することができます。

CODEPAGE

CODEPAGE は、文字エンコードに依存する、コンパイル時および実行時の COBOL 操作を処理するための、EBCDIC コード・ページのコード化文字セット ID (CCSID) に作用します。

構文

▶▶ CODEPAGE= *ccsid* ▶▶

デフォルト

CODEPAGE=1140

ccsid

EBCDIC コード・ページを示す有効な文字セット ID (CCSID) 整数を指定します。

デフォルトの CCSID 1140 は CCSID 37 (EBCDIC Latin-1, USA) と等価ですが、ユーロ記号を含みます。

推奨: COBOL および DB2® を使用するシステムでの必要のない変換および関連したパフォーマンスのオーバーヘッドを防ぐため、DB2 サブシステム・パラメーターとアプリケーション・プログラミングのデフォルト (DSNHDECP で指定) と同じ CODEPAGE コンパイラー・オプション設定を使用してください。

注: TEST オプションを指定する場合は、CODEPAGE オプションを、COBOL ソース・プログラムで使用する CCSID に設定する必要があります。特に、DBCS リテラルまたは DBCS ユーザー定義語で日本語文字を使用するプログラムは、CODEPAGE オプションを日本語コード・ページ CCSID に設定してコンパイルする必要があります。

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『CODEPAGE』を参照してください。

COMPILE

COMPILE は、指定した重大度の診断メッセージが生成された場合にコンパイルを続行するかどうかを決定します。

構文

▶▶ COMPILE= YES NOC() ▶▶

デフォルト

COMPILE=NOC(S)

NOC

構文検査のみを行うことを示します。

NOC(W)

NOC(E)

NOC(S)

エラー・メッセージ・レベルを指定します。W は警告、E はエラー、S は重大

です。指定したレベルまたはより重大なレベルのエラーが発生すると、コンパイラが停止し、それ以降はコンパイルのバランスを取る構文検査のみが行われま
す。

YES

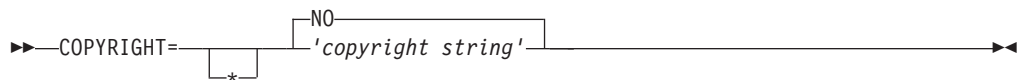
診断およびオブジェクト・コードを含む、完全なコンパイルを行うことを示しま
す。

NOCOMPILE を指定した場合、コンパイルが途中で停止し、関連データ・ファイル
に影響を与え、特定のメッセージが消失することがあります。

COPYRIGHT

COPYRIGHT は、オブジェクト・モジュールが生成された場合に、オブジェクト・モ
ジュール内にストリングを配置します。オブジェクトがプログラム・オブジェクト
にリンクされている場合、ストリングはこのプログラム・オブジェクトとともにメ
モリーにロードされます。

構文



デフォルト

COPYRIGHT=NO

copyright string の長さは 64 文字に制限されています。

CURRENCY

CURRENCY は、代替通貨記号を使用するかどうかを決定します。デフォルトはド
ル記号 (\$) です。

構文



デフォルト

CURRENCY=NO

リテラル

プログラムで使用するデフォルト通貨記号を表します。

リテラルは、以下の項目以外の 1 バイト EBCDIC 文字を表す非数値リテラル
でなければなりません。

- 数値 0 から 9
- 大文字の英字: A B C D P R S V X Z
- 小文字の英字 a から z

- スペース
- 特殊文字: * + - / , . ; () = "
- COBOL プログラムが DBCS 項目を PICTURE 記号 G で定義している場合は、大文字の英字 G。記号 G が PICTURE 節で通貨記号と見なされるため、PICTURE 節は DBCS 項目については無効となります。
- COBOL プログラムが DBCS 項目を PICTURE 記号 N で定義している場合は、大文字の英字 N。記号 N が PICTURE 節で通貨記号と見なされるため、PICTURE 節は DBCS 項目については無効となります。
- COBOL プログラムが外部浮動小数点項目を定義している場合は、大文字の英字 E。記号 E が PICTURE 節で通貨記号と見なされるため、PICTURE 節は外部浮動小数点項目については無効となります。

リテラル (16 進数リテラルを含む) の構文規則は、次のとおりです。

- リテラル区切り文字には、APOST オプションと QUOTE オプションのどちらが有効であるかに関係なく、引用符またはアポストロフィのいずれかを使用できます。
- アポストロフィ (') を通貨記号にする場合、組み込みアポストロフィは二重にする必要があります。つまり、リテラル内で 1 つのアポストロフィを表すために、2 つのアポストロフィをコーディングする必要があります。以下に例を示します。

''' または ''''

- 16 進数リテラルの指定形式は、次のとおりです。

X'H1H2' または X"H1H2"

H1H2 は、通貨記号リテラルに関する上記の規則に準拠した 1 バイトの EBCDIC 文字を表す、有効な 16 進値です。16 進数リテラル内の英字は、大文字でなければなりません。

注: 16 進値 X'20' および X'21' は使用できません。

- NO** CURRENCY オプションによって代替のデフォルト通貨記号を指定しないことを示します。また、コンパイル時に CURRENCY オプションを指定しない限り、ドル記号をプログラムのデフォルト通貨記号として使用することを示します。

値 NO を指定すると、COBOL ソース・プログラムで CURRENCY SIGN 節を省く場合と同じ結果になります。

注:

- COBOL プログラムの PICTURE 節で使用する通貨記号を選択する場合に、CURRENCY SIGN 節 (COBOL ソース・プログラムで指定する) の代わりに CURRENCY オプションを使用することができます。
- CURRENCY オプションと CURRENCY SIGN 節の両方がプログラム内で使用される場合、CURRENCY オプションが固定 (*) であっても、CURRENCY SIGN 節に指定されている記号が使用されると、その記号が PICTURE 節内の通貨記号です。

DATA

DATA オプションは、動的データ域のストレージおよび他の動的ランタイム・ストレージを 16 MB 境界の上下どちらから取得するか作用します。

構文



デフォルト

DATA=31

- 24 ユーザー・データ域が、LOC=BELOW オプションを指定した GETMAIN によって獲得されたストレージの、16 MB より下の仮想アドレスで割り振られます。

データ・パラメーターを 24 ビット・モードのプログラムに渡すプログラムを RENT オプションでコンパイルする場合、DATA=24 を指定してください。これには、以下の場合が含まれます。

- COBOL プログラムが WORKING-STORAGE 内の項目を AMODE 24 プログラムに渡す場合。
- COBOL プログラムが、参照によって、その呼び出し元から受け取ったデータ項目を AMODE 24 プログラムに渡す場合。受け取られるデータが 16 MB 境界より下にある場合も、DATA=24 を指定する必要があります。

指定しなければ、呼び出し先プログラムでデータをアドレス指定することができません。

DATA は、LOCAL-STORAGE データの位置に作用しません。代わりに、STACK ランタイム・オプションがその位置をプログラムの AMODE とともに制御します。

- 31 ユーザー・データ域 (WORKING-STORAGE や FD レコード域など) が、制限のないストレージから割り振られるか、または LOC=ANY オプションを指定した GETMAIN によって獲得されたスペースで割り振られます。このオプションを指定した場合、ストレージは 16 MB 境界より上の仮想アドレスで獲得されることもありますし、16MB 境界より下の仮想アドレスで獲得されることもあります。オペレーティング・システムは、通常は、16 MB より上の仮想アドレスのスペースが使用可能であれば、そこで要求を満たします。

注:

- プログラムを RENT オプションでコンパイルすると、WORKING-STORAGE およびパラメーター・リスト用のスペースの獲得方法を DATA オプションが制御します。
- DATA オプションは、NORENT オプションでコンパイルされるプログラムには影響を与えません。

DBCS

DBCS は、コンパイラーが非数値リテラル内の X'0E' および X'0F' を認識し、それらを DBCS データを区切るためのシフトアウトおよびシフトイン制御文字として扱うかどうか作用します。

構文



デフォルト

DBCS=YES

YES

非数値リテラル内の X'0E' および X'0F' を認識し、それらを DBCS データを区切るためのシフトアウトおよびシフトイン制御文字として扱います。

NO 非数値リテラル内の X'0E' および X'0F' をシフトアウトおよびシフトイン制御文字として認識しません。

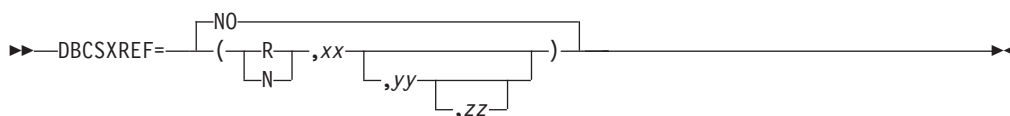
注:

- 非数値リテラル内に DBCS データが存在すると、コンパイラーはそのリテラルの特定の使用を許可しないことがあります。例えば、DBCS 文字はプログラム名または DDNAMES としては使用できません。
- DBCS=NO を指定すると、NSYMBOL(NATIONAL) と矛盾します。

DBCSXREF

DBCSXREF は、配列プログラムを使用して DBCS 名の相互参照を行うことを指示します。

構文



デフォルト

DBCSXREF=NO

- R** DBCS 日本語配列プログラム (DBCSOS) をユーザー領域にロードすることを指定します。
- N** DBCS 日本語配列プログラム (DBCSOS) を共有システム域 (MLPA など) にロードすることを指定します。
- xx** これは、DBCS 相互参照を生成するための関連配列プログラムのプログラム・オブジェクトを指定します。 8 文字の長さにする必要があります。

- yy** 配列タイプを指定します。2 文字の長さにする必要があります。このパラメーターを省略すると、指定した配列プログラムが定義するデフォルトの配列タイプが使用されます。
- zz** 指定した配列タイプが使用するエンコード・テーブルを指定します。8 文字の長さにする必要があります。このパラメーターを省略すると、特定の配列タイプに関連したデフォルトのエンコード・テーブルが使用されます。
- NO** DBCS 名の相互参照に配列プログラムを使用しないことを指定します。XREF フェーズを指定すると、プログラム内の物理的な順序に基づいた DBCS 名相互参照リストが提供されます。

注:

- DBCSXREF=NO 以外を指定するためには、DBCS 日本語配列プログラム (DBCSOS) がインストールされていなければなりません。
- R が指定されていて、場合は、ユーザー領域がコンパイラーと配列プログラムの両方を入れるために十分な大きさになるようにします。
- XREFOPT=NO と配列プログラムを指定した DBCSXREF の両方を指定すると、カスタマイズ・マクロのアセンブル時にゼロ以外の戻りコードが返されます。
- アセンブル処理は、妥当性検査で以下の状態が診断されると終了します。
 - パラメーター長が無効である
 - 「R」および「N」以外の文字が指定された
 - コマの後ろのパラメーターが欠落している
 - zz が指定されているときに yy が欠落している

DECK

DECK は、SYSPUNCH DD ステートメントで定義されたファイル内に 80 桁のオブジェクト・コード・レコードを作成するかどうかを決定します。

構文



デフォルト

DECK=NO

YES

生成されたオブジェクト・コードを、SYSPUNCH が定義するファイルに置きます。

NO SYSPUNCH にオブジェクト・コードを送りません。

DIAGTRUNC

DIAGTRUNC は、受け取り側が数値である MOVE ステートメントで、受け取りデータ項目の整数桁数が送り出しデータ項目またはリテラルの整数桁数より少ない場合に、コンパイラーが重大度 4 (警告) の診断メッセージを出すかどうか作用します。

構文



デフォルト

DIAGTRUNC=NO

YES

受け取り側が数値である MOVE ステートメントの場合、受け取りデータ項目の整数桁数が送り出しデータ項目またはリテラルの整数桁数より少ないと、コンパイラーは、重大度 4 (警告) の診断メッセージを出します。

NO コンパイラーは重大度 4 メッセージを作成しません。

注:

- 送り出し側が英数字データ名またはリテラルで、受け取り側が数値の場合の移動についても、診断メッセージが出されず (送り出しフィールドが参照変更される場合を除きます)。
- COMP-5 の受け取り側、または TRUNC(BIN) オプションが指定された場合の 2 進数の受け取り側については、診断メッセージは出されません。

DISPSIGN

DISPSIGN オプションは、符号付き数値項目の DISPLAY の出力形式設定を制御します。

構文



デフォルト: DISPSIGN=COMPAT

DISPSIGN=COMPAT

DISPSIGN=COMPAT を指定した場合、符号付き数値項目の表示値は、旧バージョンの Enterprise COBOL と互換性がある形式になります。オーバーパンチ符号が生成される場合もあります。

DISPSIGN=SEP

DISPSIGN=SEP を指定した場合、符号付き 2 進数、符号付きパック 10 進数、またはオーバーパンチ符号付きゾーン 10 進数の各項目の表示値は常に、先頭に分離符号が付いた形式になります。

以下の例は、DISPSIGN=COMPAT オプションまたは DISPSIGN=SEP オプションが指定された DISPLAY 出力を示しています。

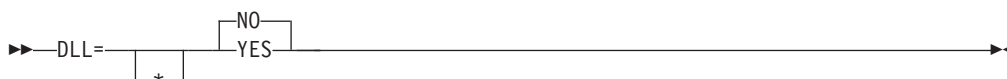
表 4. DISPSIGN=COMPAT オプションまたは DISPSIGN=SEP オプションが指定された DISPLAY 出力:

データ項目	DISPSIGN=COMPAT オプションが指定された DISPLAY 出力	DISPSIGN=SEP オプションが指定された DISPLAY 出力
符号なし 2 進数	111	111
正の 2 進数	111	+111
負の 2 進数	11J	-111
符号なしパック 10 進数	222	222
正のパック 10 進数	222	+222
負のパック 10 進数	22K	-222
符号なしゾーン 10 進数	333	333
末尾が正のゾーン 10 進数	33C	+333
末尾が負のゾーン 10 進数	33L	-333
先頭が正のゾーン 10 進数	C33	+333
先頭が負のゾーン 10 進数	L33	-333

DLL

DLL は、コンパイラによって生成されたオブジェクト・モジュールがダイナミック・リンク・ライブラリー (DLL) サポートで使用可能かどうかには作用します。

構文



デフォルト

DLL=NO

YES

ダイナミック・リンク・ライブラリー (DLL) サポートで使用可能なオブジェクト・モジュールを生成します。DLL 使用可能性が必要となるのは、プログラムが DLL の一部である場合、プログラムが DLL を参照する場合、またはプログラムがオブジェクト指向の COBOL 構文 (例えば、INVOKE ステートメント、クラス定義) を含んでいる場合です。

DLL オプションを指定する場合は、NODYNAM オプションおよび RENT オプションも使用する必要があります。

NO DLL 使用を可能にしないオブジェクト・モジュールを生成します。

関連参照

CALLINTERFACE (*Enterprise COBOL 言語解説書*)

DYNAM

DYNAM は、CALL *literal* ステートメントによって呼び出されるサブプログラムを、コンパイラが動的にロードするかどうかには作用します。

構文



デフォルト

DYNAM=NO

YES

CALL *literal* ステートメントによって呼び出されるサブプログラムを動的にロードします。

パフォーマンスの考慮: DYNAM=YES を使用すると、サブプログラムを変更した場合にアプリケーションを再リンク・エディットしないため、サブプログラムの保守が容易になります。ただし、CALL *literal* ステートメントのある個々のアプリケーションは、パスが長くなるためにパフォーマンスが多少低下する可能性があります。

NO 呼び出し側プログラムにおいて、CALL *literal* ステートメントによって呼び出されるサブプログラムのテキスト・ファイルが単一のプログラム・オブジェクトに組み込まれます。

注:

- DYNAM オプションは、コンパイル時に CALL *identifier* ステートメントには影響を与えません。CALL *identifier* ステートメントは常に、コンパイルされて動的呼び出しになります。
- CICS のもとで実行されるアプリケーションについては、DYNAM=YES を指定してはなりません。

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『DYNAM』を参照してください。

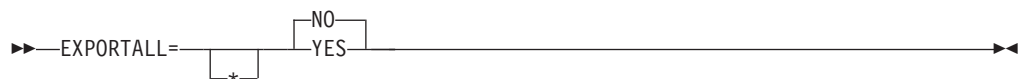
関連参照

CALLINTERFACE (Enterprise COBOL 言語解説書)

EXPORTALL

EXPORTALL は、オブジェクト・デックをリンク・エディットして DLL を作成するとき、コンパイラーが特定の記号を自動的にエクスポートするかどうか作用します。

構文



デフォルト

EXPORTALL=NO

YES

オブジェクト・デックをリンク・エディットして DLL を作成するとき、プログラム名および代替エンタリー・ポイント名を自動的にエクスポートします。

EXPORTALL を指定する場合は、DLL、RENT、および NODYNAM オプションも使用する必要があります。

NO どの記号もエクスポートしません。

FASTSRT

FASTSRT は、DFSORT または同等の製品によってソートおよびマージの入出力を実行するかどうか、または Enterprise COBOL によってそれらを実行するかどうかを決定します。形式 1 SORT ステートメントを使用したファイルのソートにのみ適用されます。

構文



デフォルト

FASTSRT=NO

YES

USING または GIVING オプションの使用時に、IBM DFSORT ライセンス・プログラムまたは同等の製品で入出力を実行することを指定します。

パフォーマンスの考慮: FASTSRT=YES を使用すると、各レコードを処理した後で Enterprise COBOL に戻る際のオーバーヘッドが軽減されます (CPU 時間の使用に関して)。しかし、このオプションを使用する場合には、従わなければならない制約事項があります。(この制約事項について詳しくは、「Enterprise COBOL プログラミング・ガイド」の『FASTSRT を使用してのソートのパフォーマンスの向上』を参照してください。)

NO Enterprise COBOL がソートおよびマージについて入出力を実行することを指定します。

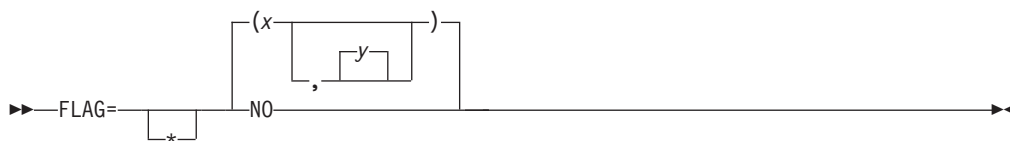
注:

- コンパイル時に FASTSRT が有効である場合、コンパイラーは、以下の 2 つを除くすべての制約事項について FASTSRT インターフェースを使用できるかどうかを検査します。
 - ソート作業ファイル用に直接アクセス装置以外の装置を使用する必要があること
 - 入力ファイルまたは出力ファイルの DD ステートメントの DCB パラメーターが、ファイルのファイル記述 (FD) と一致する必要があること
- FASTSRT が使用できない場合は、コンパイラーは、USING または GIVING オプションの使用時に診断メッセージを生成し、ソート・プログラムが入出力を実行できないようにします。このため、デフォルトとして YES を指定すると、有利な場合があります。

FLAG

FLAG は、指定した重大度レベル以上の診断メッセージをコンパイラーが生成するかどうかに作用します。

構文



デフォルト

FLAG=(I,I)

注: この構文で使用する 2 番目の重大度レベルは、1 番目の重大度レベル以上でなければなりません。

x IIWIEISIU

指定した重大度レベル以上のエラーにフラグを立て、ソース・リストの末尾に書き込むことを指定します。

ID	タイプ	戻りコード
I	通知	0
W	警告	4
E	エラー	8
S	重大エラー	12
U	回復不能エラー	16

y IIWIEISIU

任意指定の 2 番目の重大度レベルは、ソース・リストの終わりに書き込まれるだけでなくソース・リスト内に挿入される構文メッセージのレベルを指定します。

NO エラー・メッセージにフラグを立てないことを示します。

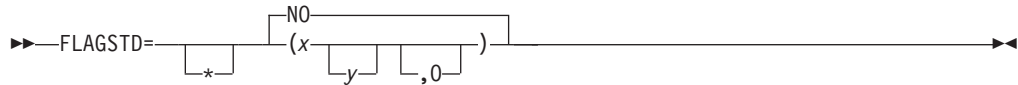
注:

- ソース・リストにメッセージを挿入する場合は、コンパイル時に **SOURCE** を指定する必要があります。こうすると、該当するソース・ステートメントの後にメッセージが置かれるので、生産性が向上します。
- **FLAG(WIEIS)** を指定すると、関連データ (**SYSADATA**) ファイル内のイベント・レコードから全クラスのメッセージが消失する場合があります。詳しくは、「Enterprise COBOL プログラミング・ガイド」の『FLAG』を参照してください。

FLAGSTD

FLAGSTD は、準拠していると見なされる 85 COBOL 標準言語エレメントのサブセット、および使用される言語エレメントに関する情報メッセージのフラグ付けに作用します。

構文



デフォルト

FLAGSTD=NO

- x** M、I、または H です。FIPS COBOL サブセットまたは規格についてのフラグを立てることを指定します。
 - M** = 標準 COBOL の ANS 最小サブセット
 - I** = ANS 最小サブセットの一部ではない別の中間サブセット言語エレメントから構成される、ANS 中間サブセット
 - H** = ANS 中間サブセットの一部ではない別の高サブセット言語エレメントから構成される、ANS 高サブセット
- y** D、N、または S の 1 つまたは 2 つの組み合わせです。フラグ設定のレベルをより詳しく定義します。
 - D** ANS デバッグ・モジュール・レベル 1 を指定します
 - N** ANS 分割モジュール・レベル 1 を指定します
 - S** ANS 分割モジュール・レベル 2 を指定します (S は N のスーパーセットです)
- 0** 上記のセットの中で生じる差し替え済みエレメントにフラグを立てることを指定します。
- NO** FIPS フラグ設定を行わないことを指定します

注:

- 以下のエレメントは、85 COBOL 標準に対する非標準 IBM 拡張機能に準拠しないものとしてフラグが立てられます。
 - COBOL 自動日付処理機能で使用される言語構文
 - オブジェクト指向および C/C++ との相互運用性の向上のための言語構文
 - PGMNAME=LONGMIXED コンパイラー・オプションの使用
- FIPS フラグ設定を指定した場合、ソース・プログラム・リスト内の通知メッセージは以下を示します。
 - 言語エレメントが廃止であるか、非標準の標準であるか、または非標準の非標準であるか (廃止かつ非標準である言語エレメントは、単に廃止としてフラグが立てられます)
 - 非標準または廃止である構文を含んでいる節、ステートメント、またはヘッダー
 - ソース・プログラム行、およびその行での開始桁の提示
 - 言語エレメントが属しているレベルまたは任意指定のモジュール
- E レベル以上のエラーとして診断されるエラーが発生すると、FIPS フラグ設定は抑制されます。
- FLAGSTD とその他のコンパイラー・オプションの相互作用:

- 以下のコンパイラー・オプションがプログラムで明示的または暗黙的に指定されている場合は、FLAGSTD=(NO 以外) を指定すると、コンパイラーの FIPS メッセージが出されます。
 - ADV=NO
 - BLOCK0=YES
 - CICS=YES
 - DLL=YES
 - DYNAM=NO
 - EXPORTALL=YES
 - FASTSRT=YES
 - LITCHAR=APOST
 - NAME=NO
 - NUMPROC=PF
 - PGMNAME=LONGMIXED
 - QUALIFY=EXTEND
 - THREAD=YES
 - TRUNC=OPT または BIN
 - VLR=COMPAT
 - WORD=(NO 以外、または RWT 以外)
 - ZONEDATA=MIG または NOPFD
 - ZWB=NO
- 以下のオプションを FLAGSTD=(NO 以外) とともに指定すると、カスタマイズ・マクロのアセンブル試行時にゼロ以外の戻りコードが返されます。
 - ADV=NO
 - DBCS=YES
 - DYNAM=NO
 - LITCHAR=APOST
 - NUM=YES
 - NUMPROC=PF
 - QUALIFY=EXTEND
 - SEQ=YES
 - TRUNC=OPT または BIN
 - VLR=COMPAT
 - WORD=(NO 以外、または RWT 以外)
 - ZONEDATA=MIG または NOPFD
 - ZWB=NO
- FLAGSTD は、関連データ・ファイル内に、FIPS 規格合致メッセージのイベント・レコードを作成することがあります。エラー・メッセージは、ソース・レコード番号の順番になるとは限りません。

FLAGSTD メッセージを診断メッセージに変換するか、抑制することができます。詳細については、47 ページの『MSGEXIT』を参照してください。

HGPR

HGPR オプションは、z/Architecture プロセッサに備わっている 64 ビット・レジスターをコンパイラーが使用する方法を制御します。



デフォルト: HGPR=PRESERVE

Enterprise COBOL コンパイラーは、64 ビット幅の z/Architecture 汎用レジスター (GPR) を使用します。HGPR は「High-halves of 64-bit GPRs (64 ビット GPR の上位半分)」の略語であり、ネイティブ 64 ビット命令を使用することを意味します。

HGPR=PRESERVE

HGPR=PRESERVE を指定した場合、コンパイラーは、プログラムが使用する 64 ビット GPR の高位半分を、関数のプロローグで保存してエピローグで復元することによって保持します。PRESERVE サブオプションは、プログラムの呼び出し元が Enterprise COBOL、Enterprise PL/I、z/OS XL C/C++ のどのコンパイラー生成コードでもない場合にのみ必要です。

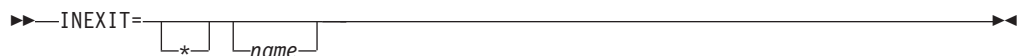
HGPR=NOPRESERVE

HGPR=NOPRESERVE を指定した場合、コンパイラーは、プログラムが使用する 64 ビット GPR の高位半分を保持する処理を省略するため、パフォーマンスが向上します。

INEXIT

INEXIT は、SYSIN データ・セットを読み取る代わりに、ソース・ステートメントを取得するために呼び出すモジュールを指定します。

構文



デフォルト

出口は指定されません。EXIT コンパイラー・オプションの NOINEXIT サブオプションを指定するのと同様です。INEXIT=* が name パラメーターなしでコーディングされた場合、NOINEXIT をオーバーライドすることはできません。

name

EXIT コンパイラー・オプションで使用するモジュールを識別します。このユーザー出口のサブオプションが指定されている場合、コンパイラーは **SYSIN** データ・セットを読み取らずに、指定されたモジュールをロードし、それを呼び出して、ソース・ステートメントを取得します。このオプションを指定した場合は、**SYSIN** データ・セットはオープンされません。

EXIT コンパイラー・オプションについて詳しくは、「Enterprise COBOL プログラミング・ガイド」の『EXIT コンパイラー・オプション』を参照してください。

INTDATE

INTDATE は、日付組み込み関数で使用される開始日に作用します。

構文



デフォルト

INTDATE=ANSI

ANSI

日付組み込み関数で使用される整数日付形式の日付に、ANSI COBOL 規格の開始日付を使用します。Day 1 = Jan 1, 1601。

INTDATE(ANSI) を指定すると、日付組み込み関数は COBOL/370 リリース 1 の場合と同じ結果を返します。

LILIAN

日付組み込み関数で使用される整数日付形式の日付に、言語環境リリアン開始日付を使用します。Day 1 = Oct 15, 1582。

INTDATE(LILIAN) を指定すると、日付組み込み関数は、言語環境プログラムの日付呼び出し可能サービスと互換性のある結果を返します。これらの結果は、COBOL/370 リリース 1 での結果とは異なります。

注:

- INTDATE(LILIAN) が有効である場合は、組み込み関数または呼び出し可能サービスのいずれかを使用して ANSI 整数を意味のある日付に変更することができないので、CEECBLDY を使用することはできません。INTDATE(LILIAN) が有効である場合、呼び出しのターゲットとして CEECBLDY を使用して CALL literal ステートメントをコーディングすると、コンパイラーはこれを診断し、この呼び出しターゲットを CEEDAYS に変換します。
- インストール・オプションを INTDATE(LILIAN) に設定した場合は、組み込み関数を使用するすべての COBOL/370 リリース 1 プログラムを再コンパイルすることにより、すべてのコードがリリアン整数日付規格を使用するようにしてください。整数の日付を保管し、プログラム間で渡したり、PL/I から COBOL および C プログラムへ問題なく渡すことができるので、このメソッドは最も安全な方式です。

LANGUAGE

LANGUAGE は、コンパイラ出力メッセージに使用される言語に作用します。

構文

▶—LANGUAGE= —XX—▶

デフォルト

LANGUAGE=EN

XX コンパイラ出力メッセージ用の言語を指定します。このパラメーターの値は、以下のリストから選択してください。

表 5. LANGUAGE コンパイラ・オプションの値

値	言語
EN または ENGLISH	英語 (大 / 小文字混合)
JA、JP、または JAPANESE	日本語
UE または UENGLISH	英語 (大文字)

注:

- LANGUAGE オプション名は、少なくとも最初の 2 文字を入力する必要があります。最初の 2 文字の後の文字も使用してかまいませんが、言語名の判別に使用されるのは最初の 2 文字だけです。
- このコンパイラ・オプションは、ランタイム・メッセージが表示されるときの言語には影響を与えません。ランタイム・オプションおよびメッセージについて詳しくは、「z/OS 言語環境プログラム プログラミング・ガイド」を参照してください。
- プリンターによっては、大文字のみを使用し、大 / 小文字混合 (LANGUAGE=ENGLISH) の出力を受け入れません。
- 日本語オプションを指定するためには、日本語機能がインストールされていなければなりません。
- 英語オプション (英大/小文字混合) を指定するためには、英語機能がインストールされていなければなりません。
- ご使用のシステムに上記以外の言語が提供されており、それをご使用のシステムのデフォルトとして選択する場合は、その言語名の少なくとも最初の 2 文字を指定する必要があります。この 2 文字は英数字でなければなりません。
- ADATA オプションの指定と一緒に日本語を選択すると、関連データ・ファイル内のエラー識別レコードに DBCS 文字が書き込まれる結果になることがあります。

LIBEXIT

LIBEXIT は、SYSLIB または library-name データ・セットを読み取る代わりに、COPY ステートメントを取得するために呼び出すモジュールを指定します。

構文

▶▶—LIBEXIT= * name

デフォルト

出口は指定されません。EXIT コンパイラー・オプションの NOLIBEXIT サブオプションを指定するのと同様です。LIBEXIT=* が name パラメーターなしでコーディングされた場合、NOLIBEXIT をオーバーライドすることはできません。

name

EXIT コンパイラー・オプションで使用するモジュールを識別します。このユーザー出口のサブオプションが指定されている場合、コンパイラーは SYSLIB または library-name データ・セットを読み取らずに、指定されたモジュールをロードし、それを呼び出して COPY ステートメントを取得します。このオプションを指定した場合は、SYSLIB および library-name データ・セットはオープンされません。

EXIT コンパイラー・オプションについて詳しくは、「Enterprise COBOL プログラミング・ガイド」の『EXIT コンパイラー・オプション』を参照してください。

LINECNT

LINECNT は、コンパイラー・ソース・リストの各ページに印刷される行数に作用します。

構文

▶▶—LINECNT= * 60 integer

デフォルト

LINECNT=60

integer

コンパイラー・ソース・コード・リストの各ページに印刷される行数を指定します。そのうちの 3 行は、ヘッディングの生成に使用されます。例えば、LINECNT=60 を指定した場合は、57 行のソース・コードが出力リストの各ページに印刷され、3 行がヘッディング用に使用されます。

LINECNT インストール・オプションは、LINECOUNT コンパイラー・オプションと同様です。

LIST

LIST は、ソース・リストにアセンブラー言語展開を作成するかどうか作用します。

構文



デフォルト

LIST=NO

YES

以下を含むリストを作成します。

- ソース・コードのアセンブラー言語展開
- 定数域
- プログラム・プロローグ域 (PPA1、PPA2、PPA3、PPA4)
- タイム・スタンプ、コンパイラー・バージョン、およびビルド・レベル情報
- コンパイラー・オプションおよびプログラム情報
- ベース・ロケーター・テーブル
- 外部シンボル辞書
- 静的マップ
- 自動マップ

NO このリストを抑止します。

LIST と OFFSET コンパイラー・オプションは、相互に排他的です。OFFSET=YES と LIST=YES を共に指定すると、カスタマイズ・マクロのアセンブル時に、ゼロ以外の戻りコードおよびエラー・メッセージが出されます。

LITCHAR

LITCHAR は、QUOTE 形象定数が引用符またはアポストロフィのどちらを表すかに作用します。

構文



デフォルト

LITCHAR=QUOTE

APOST

1 つ以上のアポストロフィ (') 文字を表すために形象定数 [ALL] QUOTE または [ALL] QUOTES が必要な場合は、APOST を使用します。

QUOTE

1 つ以上の引用符 (") 文字を表すために形象定数 [ALL] QUOTE または [ALL] QUOTES が必要な場合は、QUOTE を使用します。QUOTE は 85 COBOL 標準に準拠します。

注:

- APOST または QUOTE オプションが有効であるかどうかに関係なく、引用符またはアポストロフィのいずれかをリテラル区切り文字として使用できます。
- リテラルの左区切り文字として使用する区切り文字を、その同じリテラルの右区切り文字として使用する必要があります。

MAP

MAP オプションは、DATA DIVISION 項目に関するマップ情報と、暗黙的に宣言された項目をリストに表示するかどうかに作用します。このオプションはまた、リストのマップ出力に 16 進数または 10 進数のどちらのオフセットを表示するかを制御します。

構文



デフォルト

MAP=NO

HEX または DEC

DATA DIVISION で宣言されている項目をマップします。マップ出力には以下が含まれます。

- DATA DIVISION マップ
- グローバル・テーブル
- リテラル・プール
- ネストされたプログラム構造マップ、およびプログラム属性
- プログラムの WORKING-STORAGE と LOCAL-STORAGE のサイズ、およびプログラムが NORENT オプションを使用してコンパイルされた場合には、オブジェクト・コード内でのその位置

MAP=HEX を指定した場合、グループ内のデータ項目オフセットは 16 進表記になります。

MAP=DEC を指定した場合、グループ内のデータ項目オフセットは 10 進表記になります。

NO マッピングを実行しません。

MAXPCF

MAXPCF オプションは、最大プログラム複雑度係数値を指定するために使用します。プログラム複雑度係数 (PCF) はコンパイラによって計算され、その値はリスト・ファイルに収容されます。プログラムの PCF が最大値を超えると、コンパイラは自動的に最適化レベルを下げて、コンパイルを高速化し、ストレージ使用量を削減します。そのため、一組のプログラムをコンパイルする際に、プログラムごとに OPTIMIZE オプション値を指定する必要はありません。

構文

▶▶ MAXPCF= $\left[\begin{array}{c} \text{ } \\ \text{ } \\ * \\ \text{ } \end{array} \right] n$ ▶▶

デフォルト: MAXPCF=60000

n は 0 から 999999 までの整数でなければなりません。

複雑度係数を計算する際には、プログラムについて以下を考慮してください。

- CICS、SQL または SQLIMS オプションから生成されたステートメント、および COPY や REPLACE ステートメントの拡張を含む、PROCEDURE DIVISION 内の COBOL ステートメント数
- value 節のある WORKING-STORAGE データ項目または LOCAL-STORAGE データ項目の初期化操作
- DATA DIVISION 内の可変長グループまたはサブグループのサイズを実行時に計算する操作

巨大で複雑なプログラムの場合は、コンパイラーが最適化しようとするプログラムの複雑度に対してしきい値を MAXPCF オプションにより設定できます。MAXPCF 値を下げると最適化のレベルが下がるため、コンパイラーに必要なメモリーおよびコンパイル時間は少なくなります。コンパイル時間が長くなるのを承知の上でプログラムの最適化を試みるのであれば、MAXPCF 値を上げてください。

MAXPCF=0 を指定した場合は、プログラムの複雑度に対して制限は課されず、MAXPCF オプションは何の効果もありません。

MAXPCF= n を指定し、 n がゼロでない場合に、プログラム複雑度係数が n を超えると、OPTIMIZE(1) または OPTIMIZE(2) の指定はすべて OPTIMIZE(0) にリセットされ、警告メッセージが生成されます。

COBOL ソース・ファイルに一連のソース・プログラムが含まれている場合 (バッチ・コンパイルの場合) は、MAXPCF 制限がプログラム単位で適用されます。

注:

- OPT=1 オプションまたは OPT=2 オプションがインストール時にオーバーライド不可の固定オプションとして設定されている場合、MAXPCF= n (n はゼロ以外) は矛盾するオプションとなります。この場合は、OPTIMIZEオプションが優先され、MAXPCF=0 オプションが強制的に適用されます。
- MAXPCF の値を n (n はデフォルトより大きい値) に上げることによって、または MAXPCF(0) を指定することによって、デフォルトしきい値より大きいプログラムを最適化しようとする、コンパイラーでは、メモリー不足のために、成否にかかわらずコンパイルに過剰な時間がかかることがあります。

関連参照

52 ページの『OPTIMIZE』

MDECK

MDECK は、ライブラリー処理からの出力をファイルに書き込むかどうかに関与します。

構文



デフォルト

MDECK=NO

COMPILE

ライブラリーが処理され、MDECK 出力ファイルが生成された後、コンパイルは正常に続きます。

NOCOMPILE

ライブラリー処理が完了し、拡張ソース・プログラム・ファイルが書き込まれた後、コンパイルは終了します。

NO MDECK 出力ファイルは作成されません。

MSGEXIT

MSGEXIT は、コンパイラー・メッセージのカスタマイズを可能にするために呼び出すモジュールを指定します。

構文



デフォルト

出口は指定されません。EXIT コンパイラー・オプションの NOMSGEXIT サブオプションを指定するのと同様です。MSGEXIT=* が name パラメーターなしでコーディングされた場合、NOMSGEXIT をオーバーライドすることはできません。

name

EXIT コンパイラー・オプションで使用するモジュールを識別します。このユーザー出口のサブオプションが指定されている場合、コンパイラーは指定されたモジュールをロードし、それを呼び出してコンパイラー・メッセージのカスタマイズを可能にします。メッセージの重大度を変更でき、メッセージを抑制することができます。また、FLAGSTD コンパイラー・オプションを指定した場合に発生する FIPS メッセージを診断メッセージに変換できます。

EXIT コンパイラー・オプションについては、「Enterprise COBOL プログラミング・ガイド」の『EXIT コンパイラー・オプション』を参照してください。

NAME

NAME は、プログラム管理バインダー NAME ステートメントを各オブジェクト・モジュールに追加するかどうか、および ENTRY ステートメントごとに ALIAS ステートメントを作成するかどうかに作用します。

構文



デフォルト

NAME=NO

ALIAS

これは、プログラム内の ENTRY ステートメントごとにプログラム管理バインダー ALIAS ステートメントを作成します。PROGRAM-ID に対応する NAME ステートメントの前に ALIAS ステートメントが挿入されます。

NOALIAS

これは、バッチ・コンパイルで作成された各オブジェクト・モジュールにプログラム管理バインダー NAME ステートメント (NAME *modname*(R)) を付加します。モジュール名 (*modname*) は、PROGRAM-ID から、外部モジュール名の形成に関する規則に従って導き出されます。

NO プログラム管理バインダー NAME ステートメントは付加されません。

NAME オプションを使用すると、単一のバッチ・コンパイルで、プログラム・ライブラリー内に複数のモジュールを作成することができ、動的呼び出しでの使用に役立ちます。

NSYMBOL

NSYMBOL は、PICTURE 節で使用される N 記号の解釈を制御し、国別処理または DBCS 処理のどちらを前提とするかを指示します。

構文



デフォルト

NSYMBOL=NATIONAL

DBCS

データ項目を PICTURE 記号 N のみで構成された PICTURE 節で定義し、USAGE 節を使用しない場合は、DBCS を使用します。このようなデータ項目は USAGE DISPLAY-1 節を指定した場合と同様に処理されます。形式が N'...' または N'. . .' のリテラルは、DBCS リテラルとして扱われます。

NATIONAL

データ項目を PICTURE 記号 N のみで構成された PICTURE 節で定義し、USAGE 節を使用しない場合は、NATIONAL を使用します。このようなデータ項目は USAGE NATIONAL 節を指定した場合と同様に処理されます。形式が N". . ." または N'. . .' のリテラルは、国別リテラルとして扱われます。

注:

- NSYMBOL(DBCS) オプションは、IBM COBOL の前のリリースと互換性があります。NSYMBOL(NATIONAL) オプションは、N 記号を 2002 COBOL 標準に従って処理します。
- NSYMBOL(NATIONAL) を指定すると、DBCS オプションが強制的に実行されず。

NUM

NUM は、エラー・メッセージおよびプロシージャ・マップでソース・プログラム行番号を使用するかどうかに作用します。

構文



デフォルト

NUM=NO

YES

エラー・メッセージおよびプロシージャ・マップで、コンパイラーが生成した行番号ではなく、ソース・プログラムの行番号を使用します。

NO エラー・メッセージおよびプロシージャ・マップで、コンパイラーが生成した行番号を使用します。

COBOL プログラマーは、COPY ステートメントを使用し、NUM=YES が有効である場合には、ソース・プログラムの行番号と COPY メンバーの行番号を調整する必要があります。

NUMCLS

NUMCLS は、NUMPROC とともに、コンパイラーが数値クラス・テストで有効として扱う数値符号に作用します。

NUMCLS は、以下のすべての条件を使用して定義されているデータ項目に対して数値クラス・テストで有効と認識される符号表記を指定します。

- 符号付きである (PICTURE 節で「S」によって指定)
- DISPLAY または COMPUTATIONAL-3 (パック 10 進) を使用する
- SIGN 節に SEPARATE 句を指定しない

構文



デフォルト

NUMCLS=PRIM

ALT

ALT を指定した処理では、16 進数 A から F が有効であるとして受諾されます。

PRIM

PRIM を指定した処理では、16 進数 C、D、および F が有効であるとして受諾されます。

注:

- NUMPROC オプションと NUMCLS オプションの指定内容が、数値クラス・テストに影響を与えます。
- NUMCLS オプションは、NUMPROC=NOPFD の場合にのみ有効です。
NUMPROC=PFD は、有効な符号の構成に、より厳密な規則を指定します。

NUMPROC

NUMPROC は、内部 10 進数データおよびゾーン 10 進数データにおける符号の扱いや処理に作用します。

構文



デフォルト

NUMPROC=NOPFD

NOPFD

入力上の符号を修復します。修復が実行された後は、符号は NUMPROC=PFD に関する基準に適合します。

PFD

特に ゼロ以外の OPTIMIZE レベル (OPT=1 または OPT=2) が指定された場合に、生成コードが最適化されます。明示的な符号の修復は実行されません。NUMPROC=PFD には、正しい結果を出すための厳密な基準があることに注意してください。NUMPROC=PFD を使用する場合は、以下の事項に従ってください。

- 符号なしの数値項目の符号桁は、X'F' にする必要があります。
- 符号付きの数値項目の符号桁は、正またはゼロの場合は X'C'、負の場合は X'D' にする必要があります。
- 別個に符号が付けられる数値項目の符号桁は、正またはゼロの場合は「+」、負の場合は「-」に必要があります。

Enterprise COBOL における基本 MOVE ステートメントおよび算術ステートメントは、常にこれらの望ましい符号を伴う結果を生成しますが、グループ MOVE および再定義は、非標準の結果を生成することがあります。数値クラス・テストを検査のために使用することができます。NUMPROC=PFDF を指定すると、符号が望ましい符号基準に適合しない場合は、数値項目の数値クラス・テストが失敗します。

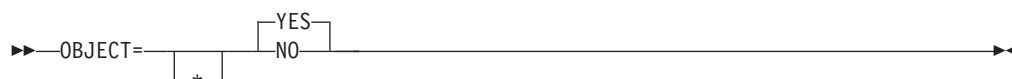
パフォーマンスの考慮: NUMPROC=PFDF を使用すると、数値の比較に関してかなり効率的なコードが生成されます。COMP-3 および DISPLAY 数値データ項目を参照するほとんどの場合、NUMPROC=NOPDF を使用すると、符号の「修正」処理のために余分なコードが生成されます。この余分なコードが原因で、他のいくつかのタイプの最適化が禁止される場合もあります。このオプションを設定する前に、アプリケーション・プログラマーと相談して、アプリケーション・プログラムの出力に与える影響を判断してください。

NUMPROC と NUMCLS の両オプションは、数値クラス・テストに影響を与えます。NUMPROC=NOPDF を指定した場合は、数値クラス・テストの結果は、NUMCLS の設定によって制御されます。NUMPROC=PFDF の場合、データ項目を望ましい符号基準と適合させて、数値と見なされるようにする必要があります。

OBJECT

OBJECT は、生成オブジェクト・コードをファイルに書き込むかどうか作用します。

構文



デフォルト

OBJECT=YES

YES

生成されたオブジェクト・コードが、バインダーへの入力として使用されるファイル (SYSLIN DD ステートメントによって定義されたもの) に配置されます。

NO オブジェクト・コードを SYSLIN に出力しません。

OBJECT=NO オプションは、TEST に指定された NO 以外のすべての値と矛盾しません。

OFFSET

OFFSET は、圧縮された PROCEDURE DIVISION リストを生成するかどうか作用します。

構文



デフォルト

OFFSET=NO

YES

圧縮された PROCEDURE DIVISION リストを生成します。リストのプロシージャ部分には、行番号、動詞の参照、および動詞ごとに生成された最初の命令の位置が含まれます。

次の項目も出力リストに書き込まれます。

- 定数域
- プログラム・プロローグ域 (PPA1、PPA2、PPA3、PPA4)
- タイム・スタンプおよびコンパイラー・バージョン情報
- コンパイラー・オプションおよびプログラム情報
- ベース・ロケーター・テーブル
- 外部シンボル辞書
- 静的マップ
- 自動マップ

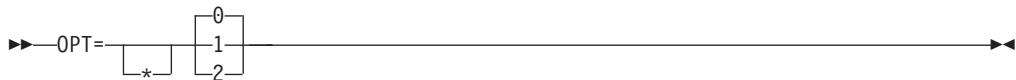
NO リストを圧縮せず、上記のものを生成しません。

LIST と OFFSET コンパイラー・オプションは、相互に排他的です。OFFSET=YES と LIST=YES をともに指定すると、カスタマイズ・マクロのアセンブル試行時にゼロ以外の戻りコードが返されます。競合解決について詳しくは、17 ページの『矛盾するコンパイラー・オプション』を参照してください。

OPTIMIZE

OPTIMIZE は、オブジェクト・コードに対して行われる最適化のレベルに作用し、その結果パフォーマンスが向上します。

構文



デフォルト

OPT=0

- 0** 制限付きの最適化が指定されます。結果として、コンパイル時間が最も短くなります。TEST オプションを指定すると、全デバッグ機能が使用可能になります。
- 1** アプリケーション実行時パフォーマンスの向上につながる最適化が指定されます。このレベルの最適化には、基本インライン化、強度の削減、複雑な演算から同等の単純な演算への単純化、一部の到達不能コードの除去、ブロック再配置な

どがあります。また、OPT=1 の場合は、共通する副次式の除去や値の伝搬など、いくつかのブロック内最適化が行われます。TEST オプションが指定されている場合は、多くのデバッグ機能が使用可能になります。

- さらなる最適化が指定されます。この最適化には、より積極的な単純化と命令スケジューリングが含まれます。また、グローバル値の伝搬およびループ不変コードの動作など、いくつかのブロック間最適化も含まれます。TEST オプションが指定されている場合は、いくつかのデバッグ機能が使用可能になります。

パフォーマンスの考慮: 通常、OPT=1 または OPT=2 を使用すると、より効率的なランタイム・コードが生成されます。

注:

- OPTIMIZE コンパイラー・オプションは、Java™ インターオペラビリティ用のオブジェクト指向構文を使用するプログラムに対して完全にサポートされています。
- S レベル・エラーまたは U レベル・エラーが発生した場合、あるいはプログラム複雑度係数が指定の MAXPCF 整数を超えた場合は、最適化は 0 に設定されます。

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『OPTIMIZE』を参照してください。

OUTDD

OUTDD は、DISPLAY 出力の送信先となる ddname を指定します。

構文



デフォルト

OUTDD=SYSOUT

ddname

実行時の DISPLAY 出力用に使用されるファイルの ddname を指定します。

実行時に、ddname として SYSOUT を必要とする別の製品と対立することが予測される場合は、このオプションのデフォルトを変更してください。

OUTDD が MSGFILE ランタイム・オプションとどのように相互作用するかについては、「z/OS 言語環境プログラム プログラミング・リファレンス」で MSGFILE の説明を参照してください。

PGMNAME

PGMNAME は、プログラム名および入り口点名の処理を制御します。

構文



デフォルト

PGMNAME=COMPAT

COMPAT

プログラム名は、COBOL/370 リリース 1 および VS COBOL II と互換性のある方法で処理されます。

LONGMIXED

プログラム名は、切り捨てられたり、変換されたり、または大文字に変換されることなく、現状のまま処理されます。

LONGUPPER

プログラム名はコンパイラーによって大文字に変換されるか、あるいは、切り捨てられたり変換されることなく、現状のまま処理されます。

PGMNAME オプションは、以下のコンテキストで使用された名前の処理を制御します。

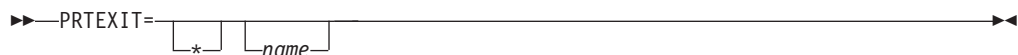
- PROGRAM-ID パラグラフで定義されたプログラム名
- ENTRY ステートメントのプログラム入り口点の名前
- 以下におけるプログラム名参照:
 - ネストされたプログラム、静的にリンクされたプログラム、または DLL を参照する、CALL ステートメント
 - 静的にリンクされたプログラムまたは DLL を参照する SET *procedure-pointer* または *function-pointer* ステートメント
 - ネストされたプログラムを参照する CANCEL ステートメント

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『PGMNAME』を参照してください。

PRTEXIT

PRTEXIT は、SYSPRINT データ・セットに書き込む出力の代わりに、呼び出すモジュールを指定します。

構文



デフォルト

出口は指定されません。EXIT コンパイラー・オプションの NOPRTEXIT サブオプションを指定するのと同様です。PRTEXIT=* が name パラメーターなしでコーディングされた場合、NOPRTEXIT をオーバーライドすることはできません。

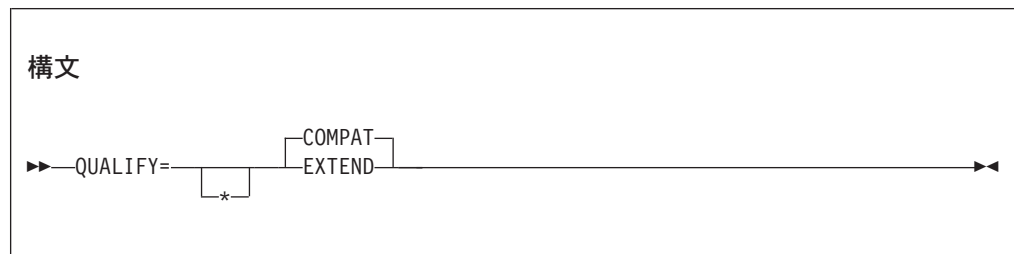
name

EXIT コンパイラー・オプションで使用するモジュールを識別します。このユーザー出口のサブオプションが指定されている場合、コンパイラーは **SYSPRINT** データ・セットに書き込む代わりに、指定されたモジュールをロードし、それを呼び出します。このオプションを指定した場合は、**SYSPRINT** データ・セットはオープンされません。

EXIT コンパイラー・オプションについて詳しくは、「Enterprise COBOL プログラミング・ガイド」の『EXIT コンパイラー・オプション』を参照してください。

QUALIFY

QUALIFY は、修飾の規則に作用し、COBOL 標準規則の下で参照できない一部のデータ項目を参照できるように修飾規則を拡張するかどうかを制御します。



デフォルト

QUALIFY=COMPAT

COMPAT

QUALIFY=COMPAT が有効になっている場合、動作は以前の COBOL コンパイラーと同じです。完全に一致する当該修飾子のセットを持つデータ項目が 1 つしか存在しない場合でも、参照は固有でなければなりません。

EXTEND

QUALIFY=EXTEND が有効になっている場合は、COBOL 標準規則では固有でない一部の参照を固有にできるように、修飾規則が拡張されます。名前のグループを含む階層内のレベルがすべて修飾されている場合、その修飾子のセットは修飾子の完全セットと呼ばれます。特定の修飾子の完全セットを持つデータ項目が 1 つしかない場合、参照はそのデータ項目に解決されます。これは、同じ修飾子のセットが、修飾子の不完全セットとして別の参照に一致する場合も同様です。

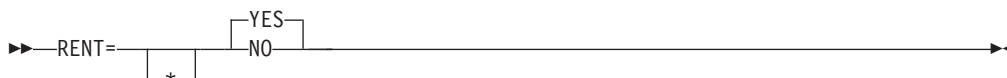
例

```
01 A.  
  02 B.  
    03 C PIC X.  
  02 C PIC X.  
  .  
  .  
Move space to C of A      *-> Refers to 02 level C (unique only with QUALIFY(EXTEND))  
Move space to C of B of A *-> Refers to 03 level C (unique by COBOL standard rules)  
Move space to C of B     *-> Refers to 03 level C (unique by COBOL standard rules)
```

RENT

RENT は、生成されるオブジェクト・コードが再入可能かどうかには作用します。

構文



デフォルト

RENT=YES

YES

生成されるオブジェクト・コードを再入可能にすることを指示します。
RENT=YES を使用すると、プログラムを 16 MB 境界より上で実行するために、共有ストレージに置くことができます。しかし、このオプションを使用すると、コンパイラーはアプリケーション・プログラムを再入可能にするための追加のコードを生成することになります。

NO 生成されるオブジェクト・コードを再入可能にしないことを指示します。

注:

- プログラムが 16 MB 境界より上の仮想記憶アドレスで実行される場合は、そのプログラムを、RENT を指定してコンパイルします。
- 16 MB より上で再入不可プログラムを実行することはサポートされていません。NORENT を指定してコンパイルされたプログラムは RMODE 24 でなければなりません。
- CICS の下で実行されるプログラムの場合、RENT コンパイラー・オプションが必要です。
- プログラムに割り当てられる RMODE は、RENTNORENT コンパイラー・オプションおよび RMODE コンパイラー・オプションによって決まります。下の表に、有効な組み合わせを示します。

表 6. RENT および RMODE が常駐モードに与える影響

RENTNORENT 設定	RMODE 設定	割り当てられる 常駐モード
RENT	AUTO	RMODE ANY
RENT	ANY	RMODE ANY
RENT	24	RMODE 24
NORENT	AUTO	RMODE 24
NORENT	ANY	コンパイラー・オプションの 競合
NORENT	24	RMODE 24

- THREAD コンパイラー・オプションを指定する場合、RENT コンパイラー・オプションも指定する必要があります。THREAD と NORENT を同じレベルの優先順位で指定した場合、RENT オプションが強制的に実行されます。

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『RENT』を参照してください。

RMODE

RMODE は、生成されるオブジェクト・プログラムの常駐モードに作用します。

構文



デフォルト

RMODE=AUTO

24 NORENT または RENT のいずれかが指定されているかには関係なく、プログラムが RMODE 24 となることを指定します。

ANY

これは、RENT が指定されている場合はプログラムで RMODE ANY が使用されるように指定し、NORENT が指定されている場合はプログラムがエラーを受け取るように指定します。

AUTO

プログラムが、NORENT が指定されている場合は RMODE 24 になり、RENT が指定されている場合は RMODE ANY になることを指定します。

注:

- AMODE 24 で実行中のプログラムにデータを渡す Enterprise COBOL NORENT プログラムは、RMODE (24) オプションを指定してコンパイルするか、RMODE 24 を指定してリンク・エディットする必要があります。NORENT プログラムのデータ域が 16 MB 境界の上下どちらに置かれるかは、プログラムの RMODE によって異なります。これは DATA(24) が指定されている場合でも同様です。DATA(24) は、RENT オプションでコンパイルしたプログラムにのみ適用されません。
- Enterprise COBOL でコンパイルしたプログラムは、必ず AMODE ANY です。プログラムに割り当てられる RMODE は、RMODE および RENTINORENT コンパイラ・オプションによって決まります。下の表に、有効な組み合わせを示します。

表7. RMODE および RENTINORENT が常駐モードに与える影響

RMODE 設定	RENTINORENT 設定	割り当てられる常駐モード
AUTO	RENT	RMODE ANY
AUTO	NORENT	RMODE 24
ANY	RENT	RMODE ANY
ANY	NORENT	適用されない。コンパイラ・エラーが発行される。

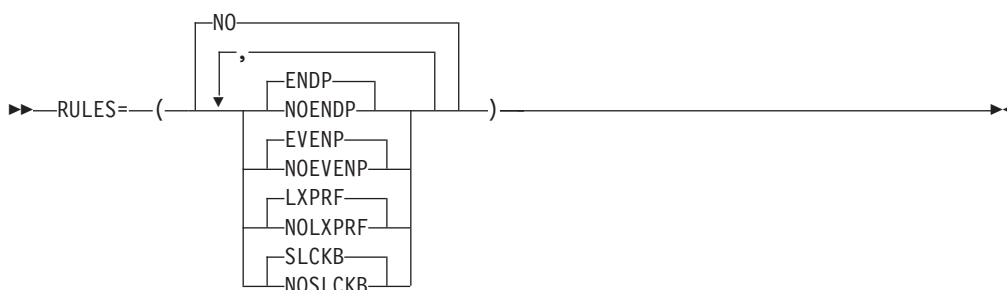
表7. RMODE および RENTINORENT が常駐モードに与える影響 (続き)

RMODE 設定	RENTINORENT 設定	割り当てられる 常駐モード
24	RENT	RMODE 24
24	NORENT	RMODE 24

RULES

RULES オプションを使用すると、コンパイル時に特定タイプのソース・コードにフラグを立てることにより、プログラムを改善するためにプログラムに関する情報をコンパイラから要求できます。

構文



デフォルト

RULES = (NO)

RULES=(ENDP,EVENP,LXPRF,SLCKB) と同じ効果を持ちます。

RULES の以下のサブオプションを指定できます (デフォルトを除く)。

RULES=(NOENDP)

条件ステートメントの有効範囲が明示範囲終了符号 END-* ではなくピリオドで終了している場合に、コンパイラから警告メッセージを出します。

RULES=(NOEVENP)

桁数が偶数であるすべての USAGE PACKED-DECIMAL (COMP-3) データ項目に対して、コンパイラから警告メッセージを出します。

RULES=(NOLXPRF)

効果のない COBOL 機能の使用に対して、コンパイラから警告メッセージを出します。このような機能には、算術ステートメントでの USAGE DISPLAY 数値データ項目、MOVE ステートメントでの大量スペース埋め込み、効果のないコンパイラ・オプションなどがあります。

RULES=(NOSLCKB)

コンパイラが遊びバイト (レコード内の遊びバイトまたはレコード間の遊びバイト) を追加することになるすべての SYNCHONIZED データ項目に対して、コンパイラから警告メッセージを出します。遊びバイトの追加を発生させるデータ項目はそれぞれ、コンパイラ診断を受け取ります。

注:

- RULES のすべてのサブオプションを指定する必要はありません。サブオプションを指定しない場合は、デフォルトが有効になります。
- RULES は、インストール・デフォルトの 1 つ以上のサブオプションとともに指定する必要があります。
- 複数のサブオプションを指定する場合は、サブオプションをコンマまたはスペースで区切ります。例えば、RULES=(NOLXPRF,SLCKB) または RULES=(NOLXPRF SLCKB) のように指定します。

SEQ

SEQ は、ソース・ステートメントがシーケンス番号の昇順になっていることをコンパイラーが検査するかどうかに作用します。

構文



デフォルト

SEQ=YES

YES

ソース・ステートメントが行番号の英数字順 (昇順) になっているかどうかを検査します。

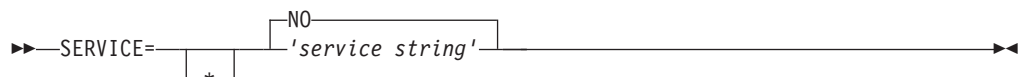
NO 順序検査を行いません。

コンパイル時に SEQ と NUM の両方が有効である場合は、順序は、英数字ではなく数字の照合順序に従って検査されます。

SERVICE

SERVICE は、オブジェクト・モジュールが生成された場合に、オブジェクト・モジュール内にストリングを配置します。オブジェクト・モジュールがプログラム・オブジェクトにリンクされている場合、ストリングはこのプログラム・オブジェクトとともにメモリーにロードされます。Language Environment ダンプにトレースバックが含まれている場合は、このストリングがそのトレースバックに組み込まれます。

構文



デフォルト

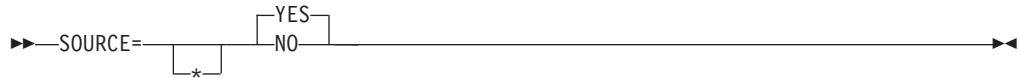
SERVICE=NO

service string の長さは 64 文字に制限されています。

SOURCE

SOURCE は、コンパイラ・リストにソース・ステートメントを含めるかどうか
に作用します。

構文



デフォルト

SOURCE=YES

YES

コンパイラ生成の出力に、ソース・ステートメントのリストを入れると指定
します。このリストには、COPY によって組み込まれたステートメントも含ま
れます。

NO 出力にソース・ステートメントを入れません。

メッセージをソース・リストに出力したい場合は、コンパイル時に SOURCE コン
パイラ・オプションを有効にしておく必要があります。

SPACE

SPACE は、ソース・リストで 1 行送り、2 行送り、または 3 行送りのいずれを使
用するか作用します。

構文



デフォルト

SPACE=1

- 1 ソース・ステートメント・リストを 1 行送りにすることを指定します。
- 2 ソース・ステートメント・リストを 2 行送りにすることを指定します。
- 3 ソース・ステートメント・リストを 3 行送りにすることを指定します。

SQL

SQL は、DB2 コプロセッサを使用可能にするかどうか、および DB2 オプション
を指定可能にするかどうか作用します。

構文



デフォルト

SQL=NO

YES

DB2 コプロセッサを使用可能にし、DB2 オプションを指定する場合に使用します。COBOL ソース・プログラムに SQL ステートメントが含まれており、そのプログラムが DB2 プリコンパイラによって処理されていない場合には、SQL オプションを指定する必要があります。

NO ソース・プログラムにある SQL ステートメントを識別して廃棄することを指定します。

SQL=NO は、COBOL ソース・プログラムに SQL ステートメントが含まれていない場合、または COBOL コンパイラを呼び出す前に別の SQL プリコンパイラを使用して SQL ステートメントを処理する場合に使用してください。

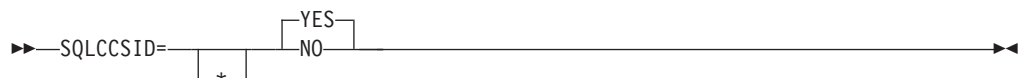
注:

- SQL オプションは、どのコンパイラ・オプション・ソースでも指定できます。すなわち、コンパイラ呼び出し、PROCESS または CBL ステートメント、またはインストール・デフォルトのどれにでも指定できます。
- DB2 オプションのストリングを区切るには、引用符またはアポストロフィを使用してください。
- DB2 オプションは、SQL オプションのカスタマイズの一環として指定することはできません。(DB2 オプションは、SQL コンパイラ・オプションが呼び出しオプションとして指定される場合、あるいは CBL ステートメントまたは PROCESS ステートメントで指定される場合にのみサポートされます)。ただし、DB2 製品のインストール・デフォルトをカスタマイズするときには、デフォルトの DB2 オプションを指定することができます。

SQLCCSID

SQLCCSID は、SQL コンパイラ・オプションが有効な場合に、CODEPAGE コンパイラ・オプションが SQL ステートメントの処理に影響を与えるかどうかを制御します。

構文



デフォルト

SQLCCSID=YES

YES

統合された DB2 コプロセッサ (SQL コンパイラ・オプション) が使用され

た場合に、CODEPAGE コンパイラー・オプション設定がソース・プログラム内の SQL ステートメントの処理に影響を及ぼすことを示します。

- NO** 統合された DB2 コプロセッサが使用された場合に、CODEPAGE コンパイラー・オプション設定がソース・プログラム内の SQL ステートメントの処理に影響を及ぼさないことを示します。COBOL ステートメントのみが CODEPAGE オプションの CCSID の影響を受けます。

注:

- SQLCCSID オプションは、統合された DB2 コプロセッサを使用する場合にのみ効力を持ちます (SQL コンパイラー・オプション)。
- DB2 プリコンパイラーの動作との最高度の互換性を必要とするアプリケーションの場合、NOSQLCCSID オプションを推奨します。
- 詳しくは、「Enterprise COBOL プログラミング・ガイド」の『SQLCCSID』を参照してください。

SQLIMS

SQLIMS は、IMS SQL コプロセッサを使用可能にするかどうか、および情報管理システム (IMS) サブオプションを指定できるかどうかに作用します。

構文



デフォルト

SQLIMS=NO

YES

これは、IMS SQL コプロセッサを使用可能にして情報管理システム (IMS) サブオプションを指定する場合に使用します。COBOL ソース・プログラムに SQLIMS ステートメントが含まれている場合は、SQLIMS オプションを指定する必要があります。

- NO** これは、ソース・プログラムにある SQLIMS ステートメントを診断して廃棄する場合に指定します。

COBOL ソース・プログラムに SQLIMS ステートメントが含まれていない場合は、SQLIMS=NO を使用します。

注:

- SQLIMS オプションは、任意のコンパイラー・オプション・ソース (コンパイラー呼び出し、PROCESS/CBL ステートメント、またはインストール・デフォルト) で指定できます。
- IMS サブオプションのストリングを区切るには、二重引用符または単一引用符を使用します。
- 長いサブオプション・ストリングを複数のサブオプション・ストリングに分割して、複数の CBL ステートメントにすることができます。

SSRANGE

SSRANGE は、範囲外のストレージ参照がないかを検査するためのコードを生成するかどうかに作用します。

構文



デフォルト

SSRANGE=NO

ZLEN|NOZLEN

割り当てられた区域外のストレージを参照しないようにするために、添え字、参照変更、可変長グループ範囲、および指標を実行時に検査するコードを生成します。生成されたコードは、関数の引数として指定された、ALL 添え字付けを指定したテーブルに、少なくとも 1 回の出現が含まれているかどうかを検査します。

生成されたコードはまた、OCCURS DEPENDING ON オブジェクトの正しくない設定の結果として、定義された最大長を可変長項目が超えていないかどうかを検査します。無制限グループまたはその従属項目の場合、検査は参照変更式に対してのみ行われます。無制限グループに従属するテーブルに対する添字付きまたは索引付きの参照は、検査されません。

ZLEN サブオプションおよび NOZLEN サブオプションは、コンパイラーによる参照変更長の検査方法を制御します。

- ZLEN が有効になっている場合、コンパイラーは、参照変更長がゼロ以上であることを保証するコードを生成します (つまり、ゼロ長の参照変更を指定すると、実行時に SSRANGE エラーを受け取りません。)
- NOZLEN が有効になっている場合、コンパイラーは、参照変更長が 1 以上であることを保証するコードを生成します (つまり、ゼロ長の参照変更を指定すると、実行時に SSRANGE エラーを受け取ります。) これは、旧バージョンの COBOL での SSRANGE の動作方法と互換性があります。

パフォーマンスの考慮: コンパイル時に SSRANGE=ZLEN または SSRANGE=NOZLEN を指定すると、オブジェクト・コードのサイズが増大し、範囲検査を行うための実行時オーバーヘッドが増加します。

NO 添え字または指標を実行時に検査するためのコードを生成しません。

注:

- SSRANGE=ZLEN または SSRANGE=NOZLEN を指定すると、コンパイラーによって範囲検査が生成され、実行時に必ず検査が実行されます。言語環境プログラムのランタイム・オプション CHECK(OFF) を指定しても、コンパイルされた範囲検査を無効にすることはできません。
- 範囲検査コードは、予期しないエラーを再コンパイルなしで解決する場合に役立つために使用できます。

STGOPT

STGOPT オプションは、ストレージ最適化を制御します。



デフォルト: STGOPT=NO

STGOPT=YES を指定した場合、コンパイラーは以下のデータ項目の一部または全部を破棄する可能性があり、そのデータ項目に対してストレージを割り振りません。

- 未参照 LOCAL-STORAGE および非外部 WORKING-STORAGE のレベル 77 およびレベル 01 の基本データ項目
- 非外部レベル 01 グループ項目 (どの従属項目も参照されない場合)
- 未参照特殊レジスター

コンパイラーは、これらの破棄されたデータ項目をその VALUE 節の値に初期化するコードを生成しません。

また、STGOPT=YES の場合は、パフォーマンスを最適化するためにメモリーにおいて LOCAL-STORAGE SECTION 内のデータ項目を再配列できます。

SUPPRESS

NOSUPPRESS オプションは、コピーブック情報をリストに表示できるように、プログラム内のすべての COPY ステートメントの SUPPRESS 句を無視する場合に使用します。コピーブック情報はデバッガーやツールなどで使用でき、その場合それらのソース・コードを変更する必要はありません。

構文



デフォルト

SUPPRESS=YES

YES

COPY ステートメントの SUPPRESS 句を有効にします。

NO COPY ステートメントの SUPPRESS 句を無視します。

TERM

TERM は、進行メッセージと診断メッセージを SYSTEMM デバイスに送るかどうかに作用します。

構文



デフォルト

TERM=NO

YES

進行メッセージと診断メッセージを SYSTEMM ファイル (特に指定しない限り、ユーザーの端末にデフォルト設定されます) に送ることを指定します。

NO メッセージを SYSTEMM ファイルに送らないことを指定します。

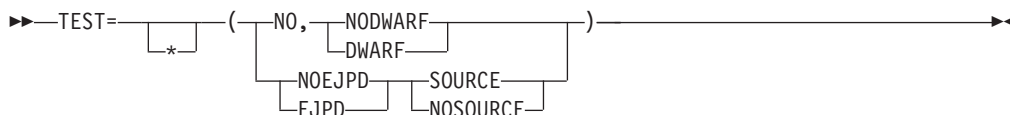
ソース・プログラム内に TERM を指定する場合は、アプリケーション・プログラムごとに SYSTEMM DD ステートメントを指定する必要があります。

TERM は TERMINAL コンパイラー・オプションに対応します。

TEST

TEST は、オブジェクト・コードに作成されるデバッグ情報の量に作用します。これにより、使用可能なデバッグ・サポートのレベルが決定されます。

構文



オプション・デフォルト

TEST=(NO, NODWARF)

TEST=(NO, DWARF)

TEST=(NO, DWARF) が有効になっている場合は、基本 DWARF 診断情報がアプリケーション・モジュールに組み込まれます。このオプションにより、アプリケーション障害分析ツール (CEEDUMP や IBM Fault Analyzer など) が使用可能になります。NOTEST(DWARF) の場合、デバッグ情報は、TEST を使用して入手できる DWARF 情報のサブセットです。NOTEST(DWARF) が有効になっている場合に生成される DWARF 診断情報は、IBM デバッグ・ツールでは使用できません。

コンパイラーによって生成されたデバッグ情報は、業界標準の DWARF 形式です。DWARF について詳しくは、「Common Debug Architecture library」の『About Common Debug Architecture』を参照してください。

TEST=(NO, NODWARF)

TEST=(NO,NODWARF) が有効になっている場合、DWARF 診断情報はアプリケーション・モジュールに組み込まれません。

TEST=(NO 以外)

デバッグ・ツールを使用してシンボリック・デバッグできるオブジェクト・コードが生成されます。TEST オプションが指定されている場合は、DWARF デバッグ情報がアプリケーション・モジュールに組み込まれます。

TEST=(EJPD, ...)

TEST=(EJPD,...) および OPT=(1|2) を指定した場合は、以下のようになります。

- デバッグ・ツール・コマンド GOTO および JUMPTO が使用可能になります。
- プログラムの最適化が削減されます。最適化はステートメント内で実行されます。最適化がステートメントの境界を越えて実行されることはほとんどありません。

TEST=(NOEJPD, ...)

TEST=(NOEJPD,...) および OPT=(1|2) を指定した場合は、以下のようになります。

- JUMPTO および GOTO は有効化されません。ただし、SET WARNING OFF デバッグ・ツール・コマンドを使用する場合は、JUMPTO および GOTO を使用できます。このシナリオでは、JUMPTO と GOTO の結果は予測不能なものとなります。
- 通常程度のプログラムの最適化が実行されます。

TEST=(..., SOURCE)

TEST=(..., SOURCE) を指定した場合は、コンパイラーによって生成された DWARF デバッグ情報に拡張ソース・コードが組み込まれるため、IBM Debug Tool ではコンパイラー・リストが不要になります。

TEST=(..., NOSOURCE)

TEST=(..., NOSOURCE) を指定した場合は、生成された DWARF デバッグ情報に拡張ソース・コードが組み込まれないため、IBM Debug Tool を使用するためにコンパイラー・リストにアクセスする必要があります。

注: TEST オプションを指定する場合は、CODEPAGE オプションを、COBOL ソース・プログラムで使用する CCSID に設定する必要があります。特に、DBCS リテラルまたは DBCS ユーザー定義語で日本語文字を使用するプログラムの場合、CODEPAGE オプションを日本語コード・ページ CCSID に設定してコンパイルしてください。詳細については、26 ページの『CODEPAGE』を参照してください。

THREAD

THREAD は、プログラムをマルチスレッド・アプリケーションで使用可能にするかどうかを制御します。

構文



デフォルト

THREAD=NO

YES

複数の POSIX スレッドまたは PL/I タスクを持つ言語環境プログラム・エンクレーブでプログラムを実行できるように指示するには、YES を使用します。

NO 複数の POSIX スレッドまたは PL/I タスクを持つ言語環境プログラム・エンクレーブでプログラムを実行できないように指示するには、NO を使用します。

パフォーマンスの考慮: THREAD コンパイラー・オプションが指定されている場合、自動的に生成される直列化ロジックが原因で実行時のパフォーマンスが低下する可能性があります。

注:

- THREAD コンパイラー・オプションが指定されている場合、プログラムはスレッド化されたアプリケーション内で使用可能になります。ただし、スレッド化されていないアプリケーションでも THREAD を使用できます。例えば、実行時にアプリケーションに複数の POSIX スレッドまたは PL/I タスクが含まれていない場合、THREAD オプションを指定してコンパイルされたプログラムを CICS 環境で実行できます。
- THREAD コンパイラー・オプションを指定する場合、RENT コンパイラー・オプションも指定する必要があります。同じレベルの優先順位で THREAD と NORENT が指定されている場合、強制的に RENT が使用されます。
- スレッド化されたアプリケーション内で COBOL プログラムを実行するには、実行単位内のすべての COBOL プログラムが THREAD コンパイラー・オプションを指定してコンパイルされている必要があります。
- THREAD コンパイラー・オプションが指定されている場合、以下の言語エレメントはサポートされません。以下の言語エレメントのいずれかを指定すると、エラーとして診断されます。
 - ALTER ステートメント
 - DEBUG-ITEM 特殊レジスター
 - プロシージャ名を含まない GO TO ステートメント
 - PROGRAM-ID パラグラフ内の INITIAL 句
 - ネストされたプログラム
 - RERUN
 - 分割モジュール
 - 形式 1 SORT ステートメントまたは MERGE ステートメント
 - STOP *literal* ステートメント
 - USE FOR DEBUGGING ステートメント

- THREAD コンパイラー・オプションを指定してプログラムをコンパイルする場合、呼び出しごとに以下の特殊レジスターが割り振られます。
 - ADDRESS OF
 - RETURN-CODE
 - SORT-CONTROL
 - SORT-CORE-SIZE
 - SORT-FILE-SIZE
 - SORT-MESSAGE
 - SORT-MODE-SIZE
 - SORT-RETURN
 - TALLY
 - XML-CODE
 - XML-EVENT
 - XML-INFORMATION
 - XML-NAMESPACE
 - XML-NAMESPACE-PREFIX
 - XML-NNAMESPACE
 - XML-NNAMESPACE-PREFIX
 - XML-NTEXT
 - XML-TEXT

TRUNC

TRUNC は、MOVE および算術演算時に 2 進データを切り捨てる方法に作用します。

構文



デフォルト

TRUNC=STD

BIN

インストール・システムのデフォルトとして使用してはなりません。以下を指定します。

- 出力 2 進数フィールドは、PICTURE 節の境界ではなく、ハーフワード、フルワード、およびダブルワード境界でのみ切り捨てられます。
- 送信される 2 進数フィールドは、ハーフワード、フルワード、またはダブルワードとして扱われ、値は PICTURE 節で暗黙指定される値に限定されずとは想定されません。
- DISPLAY は、PICTURE 記述に合わせた切り捨てを行わずに、2 進数フィールドの完全な内容を変換し、出力します。

OPT

コンパイラーは、データが PICTURE および USAGE の指定に従うものと見なします。コンパイラーは、結果を、ストレージ内のフィールドのサイズ (ハーフワードまたはフルワード) に基づいて処理します。

TRUNC=OPT をお勧めしますが、これを指定するのは、2 進域に移動されるデータが、2 進項目の PICTURE 節で定義された精度よりも大きい精度の値を持たない場合に限ってください。それ以外の場合に指定すると、高位桁が切り捨てられることがあります。

STD

85 COBOL 標準に準拠します。

MOVE および算術演算時に算術フィールドを切り捨てる方法を制御します。

TRUNC オプションは、MOVE ステートメントおよび演算式における 2 進数 (COMP) 受け取りフィールドにのみ適用されます。TRUNC=STD が有効であれば、演算式の (または MOVE ステートメントの送り出しフィールドの) 最終的な中間結果は、2 進数受け取りフィールドの PICTURE 節内の桁数に切り捨てられます。

パフォーマンスの考慮: TRUNC=OPT を使用すると、余分なコードが生成されず、通常はパフォーマンスが向上します。一方、TRUNC=BIN または TRUNC=STD を使用した場合は、BINARY データ項目が変更されると必ず余分なコードが生成されます。TRUNC=BIN を指定すると、通常、これらのオプションの中では比較的処理が遅くなります。

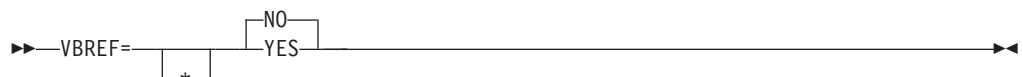
推奨:

- 他のプロダクトによって設定される 2 進値を使用するプログラムの場合、推奨されるオプションは TRUNC=BIN です。他のプロダクト (IMS、DB2、C/C++、FORTRAN、および PL/I など) が、COBOL の 2 進数データ項目に、それらのデータ項目の PICTURE 節に従わない値を入れることがあります。
- データが 2 進データ項目用の PICTURE 節に矛盾しない場合は、CICS プログラムで TRUNC=OPT を使用することができます。
- このオプションの設定は、プログラム実行時ロジックに影響を与えます。つまり、同じ COBOL ソース・プログラムでも、このオプションの設定によって結果が異なることがあります。COBOL ソース・プログラムを正しく実行するために特定の設定を前提としているかどうかを検証してください。

VBREF

VBREF は、行番号に対する動詞の相互参照と動詞使用の要約を作成するかどうかに作用します。

構文



デフォルト

VBREF=NO

YES

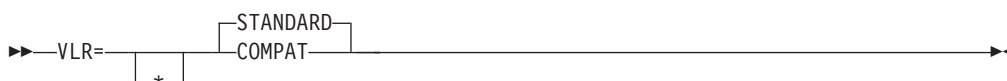
ソース・プログラム内のすべての動詞タイプと、それらが検出された行番号との相互参照を生成します。VBREF=YES を指定すると、各動詞がプログラムで使用された回数の要約も生成されます。

NO 相互参照リストまたは動詞要約リストを生成しません。

VLR

VLR オプションは、可変長レコードの READ ステートメントから返されたレコード長がレコード記述と矛盾する場合に、返されるファイル状況に作用します。これにより、ご使用のプログラムにレコード長の矛盾を引き起こす READ ステートメントがある場合に、旧バージョンから Enterprise COBOL V5 または V6 への移行が容易になります。

構文



デフォルト

VLR=STANDARD

READ ステートメントの実行後:

- 読み取られたレコード内の文字位置数がファイルのレコード記述項目で指定された最小サイズより小さい場合、レコード域内の読み取られた最後の有効文字の右側の部分は未定義になります。
- 読み取られたレコード内の文字位置数がファイルのレコード記述項目で指定された最大サイズより大きい場合、最大サイズの右側にあるレコード部分は切り捨てられます。

いずれの場合でも、READ ステートメントは正常に実行され、ファイル状況は VLR コンパイラー・オプション設定に応じて、00 (レコード長の矛盾状態を隠します) または 04 (レコード長の矛盾が発生したことを示します) に設定されます。

COMPAT

VLR=COMPAT を指定した場合、READ ステートメントでレコード長の矛盾が検出されると、状況値 00 を受け取ります。

注: この設定により、不正な長さの読み取り状態で引き起こされる入出力問題を隠すことができます。VLR=COMPAT オプションの使用には注意が必要です。また、READ ステートメントが正しいかどうかを確認してください。

STANDARD

VLR=STANDARD を指定した場合、READ ステートメントでレコード長の矛盾が検出されると、状況値 04 を受け取ります。

FS=04 をテストするためのコードを追加して、レコード内の未定義データをアクセスしないように、また切り捨てられたレコード部分を参照する試行に関する保護例外を受け取らないようにすることができます。

VSAMOPENFS

VSAMOPENFS オプションは、ファイル整合性検査の確認を必要とする、正常に実行された VSAM OPEN ステートメントから報告されるユーザー・ファイル状況に作用します。

構文



デフォルト

VSAMOPENFS=COMPAT

COMPAT

VSAMOPENFS=COMPAT を指定した場合、VSAM OPEN ステートメントが正常に検査されると、ステートメントはファイル状況 97 を返します。これは、V6 以前の COBOL の実行時動作と互換性があります。

SUCC

VSAMOPENFS=SUCC を指定した場合、VSAM OPEN ステートメントが正常に検査されると、ステートメントはファイル状況 00 を返します。これにより、ユーザーは、他の正常な操作で通常行うように、返されたファイル状況の最初の桁で 0 の有無を確認するだけで済みます。

WORD

WORD は、コンパイル時に使用する代替予約語テーブルを指示します。

構文



デフォルト

WORD=NO

CICS

CICS 固有の予約語テーブル (IGYCCICS) が、代替予約語テーブルとして提供されます。詳細については、14 ページの『CICS 予約語テーブル (IGYCCICS)』を参照してください。

xxxx

コンパイル時に使用される代替のデフォルト予約語テーブルを指定します。**xxxx** は、使用される予約語テーブルの名前の最後の文字 (1 から 4 文字の長さ) を表しています。最初の 4 文字は IGYC です。最後の 4 文字は、以下にリストする文字ストリングのいずれかにすることはできず、ドル記号文字 (\$) を入れることもできません。

- CBE
- DGEN

- DIAG
- DMAP
- DOPT
- ECWI
- FGEN
- INIT
- LIBO
- LIBR
- LSTR
- LVL0
- LVL1
- LVL2
- LVL3
- LVL8
- OSCN
- PGEN
- RCTL
- RDPR
- RDSC
- RWT
- SAW
- SCAN
- SIMD
- XREF

NO 代替予約語テーブルをデフォルトとして使用しないことを指定します。

注:

- **WORD** の指定は、入力予約語の解釈に影響を与えます。システム名 (UPSI、SYSPUNCH など) および組み込み関数名を予約語の別名として使用してはなりません。予約語テーブル **ABBR** 制御ステートメントを使用して関数名を別名として指定すると、その関数名はコンパイラーによって予約語として認識および診断されるので、組み込み関数は実行されません。
- **WORD=XXXX** オプションのデフォルト値を変更すると、**FLAGSTD** に指定された **NO** 以外のすべての値と矛盾します。

XMLPARSE

XMLPARSE は、XML 入力の処理に使用するパーサーと、その結果使用可能な XML 構文解析機能を指示します。

構文



デフォルト

XMLPARSE=XMLSS

COMPAT

XML PARSE ステートメントは、COBOL ライブラリーの組み込みコンポーネントである XML パーサーを使用して処理されます。XML PARSE ステートメントの結果および操作上の動作は、Enterprise COBOL バージョン 3 で取得されるものと互換性があります。また、XMLPARSE(COMPAT) が使用された場合はバージョン 4 と互換性があります。

XMLSS

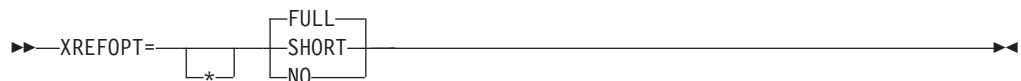
XML PARSE ステートメントは z/OS XML システム・サービス・パーサーを使用して処理されます。以下の XML 構文解析機能は、このサブオプションが有効である場合にのみ使用できます。

- ネーム・スペースの処理の拡張
- XML PARSE ステートメントの ENCODING、RETURNING NATIONAL、および VALIDATING 句
- UTF-8 でエンコードされた XML 文書の直接的な構文解析のサポート
- 非常に容量が大きい XML 文書の構文解析 (一度に 1 つの XML バッファ) をサポート
- XML 構文解析の System z® Application Assist Processors (zAAPs) へのオフロード

XREFOPT

XREFOPT は、XREF コンパイラー・オプションのデフォルト値を設定します。これは、リストに作成される相互参照情報の量に作用します。

構文



デフォルト

XREFOPT=FULL

FULL

プログラム内で使用されている名前のソート済み相互参照を生成し、それらの名前が定義されている行を示します。COPY/BASIS 相互参照も生成します。SOURCE=YES も指定されている場合、組み込み相互参照情報がソースと同じ行に印刷されます。

SHORT

明示的に参照されている変数のみのソート済みリストを生成し、COPY/BASIS 相互参照を生成します。

NO 相互参照リストを抑制します。

注:

- XREFOPT=NO は、DBCSXREF の NO 以外の値と矛盾します。
- デフォルトを XREFOPT=NO に変更しないことをお勧めします。 XREFOPT=NO の場合、COPY/BASIS 相互参照が場合によっては不完全であったり欠落することがあります。

詳しくは、「Enterprise COBOL プログラミング・ガイド」の『XREF』を参照してください。

ZONECHECK

ZONECHECK オプションは、送信データ項目として使用するゾーン 10 進データ項目に対して IF NUMERIC クラス・テストを生成するようにコンパイラーに指示する場合に使用します。

構文



デフォルト

ZONECHECK=NO

MSG

MSG サブオプションは、送信側としてゾーン 10 進数データ項目を使用するたびに IF NUMERIC テストを実行するよう要求し、データが無効 (例: NOT NUMERIC) の場合は行番号、データ項目名、データ項目内容、およびプログラム名を示すラントタイム警告メッセージが出されます。

ABD

ABD サブオプションは、送信側としてゾーン 10 進数データ項目を使用するたびに IF NUMERIC テストが実行されるよう要求し、データが無効 (例: NOT NUMERIC) の場合は強制終了メッセージが出され、異常終了となります。

ZONECHECK=MSG と ZONECHECK=ABD のどちらを使用した場合も、コンパイラーは COBOL ステートメント内で送信側として参照される各ゾーン 10 進数データ項目に対して、暗黙的な数値クラス・テストを生成します。以下のサンプル・ステートメント内のデータ項目 B のように、受信側が送信側と受信側の両方を兼ねていない限り、受信側はチェックされません。

```
ADD A TO B
DIVIDE A INTO B
COMPUTE B = A + B
INITIALIZE B REPLACING ALPHANUMERIC BY B
```

このチェックは、各ステートメント内でデータ項目が使用される前に実行されま
す。

- データ項目が NOT NUMERIC の場合は、警告メッセージ (ZONECHECK=MSG の場合)
または強制終了メッセージ (ZONECHECK=ABD の場合) が出されます。
- データ項目が NUMERIC の場合、ステートメントの外部動作は、低速になる以外は
ZONECHECK=NO と同じです。

パフォーマンスの考慮事項: ZONECHECK=MSG と ZONECHECK=ABD は、プログラム内の
COBOL ステートメントで使用されるゾーン 10 進数データ項目の数によっては、
ZONECHECK=NO よりもはるかに低速になります。

ZONEDATA

ZONEDATA オプションは、有効なゾーン・ビットを持つデータが USAGE DISPLAY 数
値データ項目 (ゾーン 10 進数) に含まれているかどうかをコンパイラーに指示し、
含まれていない場合は、さらにコンパイラーの動作を指示します。

構文



デフォルト

ZONEDATA=PFD

PFD

ZONEDATA=PFD が有効な場合、コンパイラーは、ゾーン 10 進数データ項目内の
ゾーン・ビットが有効であると想定し、数値比較が可能な最も効率の良いコード
を生成します。例えば、コンパイラーは数値変換を避けるためにストリング比較
を生成する場合があります。

MIG

ZONEDATA=MIG が有効な場合、コンパイラーは、ゾーン 10 進数データ項目内の
各桁のゾーン・ビットを無視する数値比較を行う命令を生成します。例えば、ゾ
ーン 10 進数値は比較前に、パック命令によってパック 10 進数に変換されま
す。さらに、ゾーン 10 進数データ項目に無効なゾーン・ビットが含まれている
場合、コンパイラーは、COBOL V4 とは異なる結果を生む可能性があることが
分かっている最適化を実行しません。

NOPFD

ZONEDATA=NOPFD が有効な場合、コンパイラーは、COBOL V4 で
NUMPROC=NOPFD|PFD を使用した場合に COBOL V4 が実行するのと同じ方法
で、ゾーン 10 進数データの数値比較をするための命令を生成します。

- COBOL V4 がゾーン・ビットを考慮した場合は、ゾーン 10 進数データ項目
内の各桁のゾーン・ビットが数値比較に考慮されます。ゾーン 10 進数値は
ゾーン 10 進数のままで、文字列比較が使用されます。
- COBOL V4 がゾーン・ビットを無視した場合は、ゾーン 10 進数データ項目
内の各桁のゾーン・ビットが数値比較でも無視されます。ゾーン 10 進数値
は比較前に、PACK 命令によってパック 10 進数に変換されます。

さらに、ゾーン 10 進数データ項目に無効なゾーン・ビットが含まれている場合、コンパイラーは、COBOL V4 とは異なる結果を生む可能性があることが分かっている最適化を実行しません。

注: 符号ゾーンは、NUMPROC コンパイラー・オプション設定に準じた有効な符号でなければなりません。また、下位バイトには、SIGN IS LEADING または SIGN IS SEPARATE のいずれかによる符号なしおよび符号付きの有効なゾーン (x'F') が必要です。

ZONEDATA=NOPFD は、ゾーン 10 進数データ項目に有効なデータが含まれていない可能性があることをコンパイラーに通知します。実際に含まれていない場合は、NUMPROC=NOPFD オプションまたは NUMPROC=PFDF オプションが指定された COBOL V4 コンパイラーでコンパイルを行う場合と同じ動作をしなければなりません。具体的な動作は、COBOL V5 または V6 の NUMPROC サブオプションによって異なります。

- COBOL V5 または V6 で COBOL V4 の NUMPROC=NOPFD 動作を行うには、COBOL V5 または V6 で ZONEDATA=NOPFD および NUMPROC=NOPFD を使用します。
- COBOL V5 または V6 で COBOL V4 の NUMPROC=PFDF 動作を行うには、COBOL V5 または V6 で ZONEDATA=NOPFD および NUMPROC=PFDF を使用します。

どの値を指定した場合でも、COBOL V5 または V6 へのマイグレーションを容易にするには、以下のようにします。

- ゾーン・ビットが有効な場合は、ZONEDATA=PFDF と、COBOL V5 または V6 の使用時に COBOL V4 で使用したのと同じ NUMPROC 設定を使用します。
- 無効なゾーン・ビットがある場合:
 - NUMPROC=MIG を COBOL V4 で使用した場合、COBOL V5 または V6 では ZONEDATA=MIG および NUMPROC=NOPFD を使用します。
 - NUMPROC=NOPFD を COBOL V4 で使用した場合、COBOL V5 または V6 では ZONEDATA=NOPFD および NUMPROC=NOPFD を使用します。
 - NUMPROC=PFDF を COBOL V4 で使用した場合、COBOL V5 または V6 では ZONEDATA=NOPFD および NUMPROC=PFDF を使用します。

パフォーマンスの考慮事項: ZONEDATA=PFDF では、ZONEDATA=NOPFD|MIG よりもランタイム・パフォーマンスが向上します。ZONEDATA=NOPFD|MIG は、NUMPROC=PFDF による最適化の一部を無効にします。

ZWB

ZWB は、コンパイラーが、実行時に符号付きゾーン 10 進数フィールドを英数字フィールドと比較する前に、符号付きゾーン 10 進数フィールドから符号を除去するかどうかを決定します。

構文



デフォルト

ZWB=YES

YES

実行中に符号付き外部 10 進数 (DISPLAY) フィールドを英数字フィールドと比較するときに、符号付き外部 10 進数フィールドから符号を除去します。

NO 実行中に符号付き外部 10 進数 (DISPLAY) フィールドを英数字フィールドと比較するときに、符号付き外部 10 進数フィールドから符号を除去しません。

注:

- このオプションの設定は、プログラム実行時論理に影響を与えます。つまり、同じ COBOL ソース・プログラムでも、このオプションの設定によって結果が異なることがあります。Enterprise COBOL ソース・プログラムを正しく実行するために、特定の設定を前提としているかどうかを検証してください。
- アプリケーション・プログラマーは、SPACES の入力数値フィールドをテストするために ZWB=NO を使用します。

第 3 章 Enterprise COBOL のカスタマイズ

以下のセクションでは、Enterprise COBOL をカスタマイズするために変更可能な、提供されるユーザー・ジョブおよびモジュールについて説明します。

Enterprise COBOL に変更を加えることができるのは、この製品のインストールが完了した後に限られます。

それらの変更の 1 つは、SMP/E USERMOD を使用して行います。USERMOD を適用する前に Enterprise COBOL を配布ライブラリーの中に ACCEPT しなかった場合は、SMP/E RESTORE ステートメントを使用して USERMOD を除去することはできません。USERMOD を配布ライブラリーの中に受け入れてはなりません。USERMOD がインストール先のプログラマーの要件を満たさないことが判明した場合、USERMOD を除去することがあるからです。

USERMOD によって変更されるモジュールにサービスを適用するときは、その前に USERMOD を除去する必要があります。この場合、サービスが正常にインストールされた後、USERMOD を再適用することができます。

重要: Enterprise COBOL がインストール先のアプリケーション・プログラマーの要件を満たすようにしてください。Enterprise COBOL のカスタマイズを計画する際には、アプリケーション・プログラマーと相談してください。そうすることにより、インストール時に行う変更は、インストール先で開発されるアプリケーション・プログラムを最も適切にサポートすることになります。

Enterprise COBOL をインストールするために必要な情報はすべて、製品に付属のプログラム・ディレクトリーに含まれています。

ユーザー変更の要約

Enterprise COBOL をインストールすると、多数のサンプル変更ジョブがターゲット・データ・セット IGY.V6R1M0.SIGYSAMP に置かれます。ただし、これらのサンプル変更ジョブは、特定システム用にカスタマイズされていません。これらのジョブをカスタマイズする必要があります。

80 ページの表 8 は、これらのサンプル変更ジョブの名前を示しています。これらのジョブについては、示されている参照ページに詳しい説明があります。

IGY.V6R1M0.SIGYSAMP のメンバーを変更して実行依頼する前に、これらのメンバーを個人用データ・セットの 1 つにコピーして、変更を中止する場合のために、未変更のバックアップ・コピーがある状態にしておいてください。

可能な変更についての説明は、JCL 内のコメントに示されています。TSO を使用して、ジョブを変更し、実行依頼することができます。

表 8. Enterprise COBOL 用のユーザー変更ジョブの要約

説明	ジョブ	参照ページ
コンパイラー・オプション・デフォルト・モジュールの変更	IGYWDOPT	81 ページの『コンパイラー・オプション・デフォルト・モジュールの変更』
固定として指定されたオプションのオーバーライド	IGYWUOPT	82 ページの『固定として指定されたオプションのオーバーライド』
予約語の変更	IGYWRWD	82 ページの『予約語の変更』
Enterprise COBOL モジュールを共有ストレージに置く	サンプル・ジョブはインストールされていない	88 ページの『共有ストレージへの Enterprise COBOL モジュールの配置』

コンパイラー・オプションのデフォルトの変更

コンパイラー・オプションのデフォルトは変更することができ、固定のコンパイラー・オプションをオーバーライドすることができます。

コンパイラー・オプションのデフォルトを変更するには、以下のようになります。

1. オプション・モジュール IGYCDOPT のソースを IGY.V6R1M0.SIGYSAMP から該当するジョブにコピーして、2 行のコメントを置き換えます。
2. IGYCOPT マクロ呼び出しのパラメーターを変更して、ご使用のシステム用に選択したコンパイラー・オプションに一致するようにします。

変更した IGYCOPT マクロ呼び出しをコーディングする際には、以下の要件に注意してください。

- IGYCOPT 呼び出しの各行 (最終行以外) の桁 72 に継続文字 (ソース中の X) を置く必要があります。継続行は、桁 16 で開始する必要があります。任意のコンマの後で、コーディングを分割することができます。
- マクロの最初のオプションの前にコンマを置かないでください。
- オプションおよびサブオプションは、大文字で指定する必要があります。ストリングであるサブオプションのみは、大小混合または小文字で指定することができます。
- ストリング・サブオプションの 1 つに特殊文字 (例えば、組み込みブランク、対応しない右括弧または左括弧) を含める場合は、そのストリングを二重引用符 (") ではなくアポストロフィ (') で囲む必要があります。連続するアポストロフィまたは引用符のいずれかを使用して、ヌル・ストリングを指定することができます。

ストリングの中でアポストロフィ (') または単一アンバーサンド (&) を使用する際には、その文字を 2 つ続けて指定する必要があります。最大許容ストリングの長さを超えたかどうかを判別するときや、ストリングの有効長を設定するとき、このペアは 1 文字として数えられます。

- アポストロフィを使用するときは、どのストリングにおいても、対応しないアポストロフィの使用を避けてください。このエラーは、IGYCOPT 自体の中では検出されません。代わりに、アセンブラーは次のようなメッセージを出します。

IEV03 *** ERROR *** NO ENDING APOSTROPHE

このメッセージは、違反しているサブオプションに対する位置関係を示すものではありません。さらに、このエラーが発生すると、どのオプションも正しく解析されません。

- デフォルト値を変更するオプションのみをコーディングしてください。コーディングしないオプションについては、IGYCOPT マクロがデフォルト値を生成します。

デフォルトのコンパイラー・オプションを計画する際に役立つワークシートについては、4 ページの『コンパイラー・オプション用の IGYCDOPT ワークシート』を参照してください。オプションの説明については、17 ページの『第 2 章 Enterprise COBOL コンパイラー・オプション』を参照してください。

- マクロ命令の後に、END ステートメントを置いてください。

マクロ呼び出しのコーディング方法について詳しくは、「*High Level Assembler for z/OS Language Reference*」を参照してください。

コンパイラー・オプション・デフォルト・モジュールの変更

Enterprise COBOL のコンパイラー・オプションのデフォルトを変更するには、サンプル・ジョブ IGYWDOPT を変更します。

デフォルト値を選択するには、17 ページの『第 2 章 Enterprise COBOL コンパイラー・オプション』に記載されている情報を使用します。

いずれかのコンパイラー・フェーズ・オプションの値として OUT を指定した場合は、新しいコンパイラー・オプション・デフォルト・モジュールを使用してプログラムをコンパイルする前に、必ずこれらのフェーズを共有ストレージに置いてください。詳細については、8 ページの『コンパイラー・フェーズおよびそのデフォルト』および 88 ページの『共有ストレージへの Enterprise COBOL モジュールの配置』を参照してください。

IGYWDOPT の JCL を変更する場合の手順:

1. ご使用のシステムに適した JOB カードを追加します。
2. ご使用のシステムで必要な場合、JES ROUTE カードを追加します。
3. IGYWDOPT 内の 2 行のコメント行を、IGY.V6R1M0.SIGYSAMP にある IGYCDOPT のソースのコピーで置き換えます。
4. IGYCDOPT 内の IGYCOPT マクロ・ステートメントのパラメーターをコーディングして、ご使用のシステムでのデフォルト・コンパイラー・オプションのために選択した値を反映させます。
5. #GLOBALCSI をグローバル CSI 名に変更します。
6. SET BDY ステートメント上の #TZONE をターゲット・ゾーン名に変更します。

ジョブ IGYWDOPT を変更した後、このジョブを実行依頼します。このジョブが正しく実行されると、条件コード 0 が返されます。リストの IGYnnnn 通知メッセージを調べて、ご使用のシステムで有効になるデフォルトを確認してください。

固定として指定されたオプションのオーバーライド

固定として指定された 1 つ以上のオプションをオーバーライドすることが必要なアプリケーションがある場合があります。

そのアプリケーションのコンパイル時に STEPLIB または JOBLIB としてアクセスできる別個のデータ・セット (IGY.V6R1M0.SIGYCOMP データ・セットの前に置く) に、オプション・モジュールの一時コピーを作成することによって、その他のオプションを提供できます。

サンプル・ジョブ IGYWUOPT は、ユーザー指定のデータ・セットの中にリンク・エディットされるデフォルト・オプション・モジュールを作成します。アセンブリおよびリンク・エディットは SMP/E の制御範囲外で行われるので、標準のデフォルト・オプション・モジュールは影響を受けません。

IGYWUOPT の JCL を変更する場合の手順:

1. ご使用のシステムに適した JOB カードを追加します。
2. ご使用のシステムで必要な場合、JES ROUTE カードを追加します。
3. IGYWUOPT 内の 2 行のコメント行を、IGY.V6R1M0.SIGYSAMP にある IGYCDOPT のソースのコピーで置き換えます。
4. IGYWUOPT 内の IGYCOPT マクロ・ステートメントのパラメーターを変更して、この固定オプションのオーバーライド・コンパイラー・オプション・モジュールのために選択した値を反映させます。
5. IBM 提供ものとは異なる接頭部を Enterprise COBOL ターゲット・データ・セットに使用することを選択した場合は、SYSLIB DD ステートメント (「<<<<<」でマークされている) を検査して、データ・セット名が正しいことを確認してください。
6. SYSLMOD DD ステートメントにおける DSNAME=YOURLIB を、IGYCDOPT モジュールのバインド先となるデータ・セットの名前に変更します。選択したデータ・セットに現在入っている IGYCDOPT モジュールは、新しいバージョンで置き換えられることに注意してください。

ジョブ IGYWUOPT を変更した後、このジョブを実行依頼します。このジョブが正常に実行されると、両方のステップが条件コード 0 を返します。さらに、リストの IGYnnnn 通知メッセージを調べて、標準のデフォルト・オプション・モジュールの代わりにこのモジュールが使用されるときに有効になるデフォルトを確認してください。

予約語の変更

Enterprise COBOL で予約語として扱われている語を変更するには、予約語テーブル・ユーティリティを使用します。

Enterprise COBOL が使用する予約語は、この製品で提供されるテーブル (IGYCRWT) で保守されます。CICS 固有の予約語テーブル (IGYCCICS) が、代替予約語テーブルとして提供されています (14 ページの『CICS 予約語テーブル (IGYCCICS)』を参照)。予約語テーブル・ユーティリティ (IGY8RWTU) を使用し

て IGYCRWT または IGYCCICS を変更するか、追加の予約語テーブルを作成することによって、予約語を変更することができます。以前に作成したテーブルを変更することもできます。

予約語テーブル・ユーティリティは、予約語テーブルを作成または変更するための制御ステートメントを受け入れます。その結果、新しいテーブルには、IBM 提供のテーブルからの予約語と、ユーザーが適用したすべての変更が含まれます。

以下のタイプの変更を予約語テーブルに加えることができます。

- プログラムで使用されたときに通知メッセージで示される語を追加する。この通知メッセージを作成するには、IGYCRWT 予約語テーブルを変更し、FLAGSTD オプションを使用してコンパイルする必要があります。
- プログラムで使用されたときに重大エラー・メッセージで示される語を追加する。
- 現在は通知メッセージまたはエラー・メッセージで示される語に、メッセージを出さないことを指示する。

作成する各予約語テーブルには、固有の 1 から 4 文字の ID が必要です。使用できない文字ストリングのリストについては、71 ページの『WORD』を参照してください。

コンパイル時に、コンパイラ・オプション WORD(XXXX) の値により、使用する予約語テーブルが識別されます。XXXX は、メンバー名 IGYCXXXX 中に指定した、1 から 4 文字の固有の ID です。複数の予約語テーブルを作成できますが、コンパイル時に指定できる予約語テーブルは 1 つだけです。

注： 1 つの予約語テーブル内の項目の合計数は、サイズが 1536 バイト (1.5 KB) 以内になるようにしてください。

次に示す例によって、IBM 拡張予約語 ENTRY がプログラムで使用されると、メッセージ 0086 が出されます。

```
INFO ENTRY
```

次の例では、BOOLEAN、XD、および PARENT の使用が制限されています。これらを使用すると、エラーが発生します。

```
RSTR BOOLEAN
      XD
      PARENT
```

次の例では、GO TO および ALTER の使用が制限されています。これらを使用すると、エラーが発生します。

```
RSTR GO
      ALTER
```

次の例では、生成された予約語テーブルにより、すべての 85 COBOL 標準言語の使用が可能になります (ネストされたプログラムを除く)。

```
RSTR IDENTIFICATION(1) only allow 1 program per compilation unit
RSTR ID(1)               same for the short form
RSTR PROGRAM-ID(1)      only allow 1 program per compilation unit
RSTR GLOBAL              do not allow this phrase at all
```

予約語テーブルの作成または変更

予約語テーブルを作成するか、既存の予約語テーブルのいずれかを変更できます。

以下のように、該当するソース・ファイルのコピーを編集します。

- I • IGY.V6R1M0.SIGYSAMP 内のメンバー IGY8RWRD (IBM 提供のデフォルト予約語テーブル)
- I • IGY.V6R1M0.SIGYSAMP 内のメンバー IGY8CICS (IBM 提供の CICS 予約語テーブル)
- ユーザー・ファイル (ユーザー提供の予約語テーブル)

適切な非 SMP/E JCL を変更し、起動する必要もあります。

ファイルには 4 つのパートがあります。つまり、パート I、II、III、および IV です。ファイルおよび非 SMP/E JCL を次のように変更してください。

1. ファイルのプライベート・コピーを作成します。
2. パート I (キーワード MOD を含んでいる行までのすべての行) をスキップします。ファイルのこの部分を変更してはいけません。
3. 予約語を含む行のうち、通知メッセージを発行したくない行があれば、その行の 1 桁目にアスタリスクを付けてパート II を編集します。
4. 重大メッセージが出される必要のない予約語を含む行の桁 1 にアスタリスクを置くことにより、パート III を編集します。
5. 必要な変更を作成する追加の予約語制御ステートメントをコーディングすることにより、パート IV を編集します (『制御ステートメントのコーディング』を参照)。
6. JCL を変更し、実行します (88 ページの『予約語テーブルを作成する JCL の実行』を参照)。さらに、新しい予約語テーブルの固有の 1 から 4 文字の ID を作成し、それを JCL 内に指定する必要があります。

制御ステートメントのコーディング

予約語テーブルを作成または変更するには、予約語テーブルに作用する制御ステートメントの構文を理解する必要があります。

下の図は、予約語制御ステートメントのコーディングの形式を示しています。

```
ABBR reserved-word: user-word [comments]
      [reserved-word: user-word [comments]]
      ?
INFO COBOL-word [(0 | 1)] [comments]
      [COBOL-word [(0 | 1)] [comments]]
      ?
RSTR COBOL-word [(0 | 1)] [comments]
      [COBOL-word [(0 | 1)] [comments]]
      ?
```

図 2. 予約語制御ステートメントの構文形式

上の図に示されているように、使用できるキーワードは次のとおりです。

ABBR 既存の予約語の代替形式を指定します。

INFO プログラムで使用されたときおよび FLAGSTD コンパイラー・オプションが有効なときに通知メッセージで示される語を指定します。

RSTR プログラムで使用されたときにエラー・メッセージで示される語を指定します。

制御ステートメント・キーワード **INFO** および **RSTR** で指定した語にはすべて、その語を使用する Enterprise COBOL プログラムのソース・リストにおいてメッセージでフラグが立てられます。省略語の場合、**INFO** または **RSTR** 制御ステートメントでその省略語を指定しない限り、その省略語はソース・リストにおいてメッセージが出されません。

制御ステートメントのコーディング規則

制御ステートメントをコーディングする際には、規則に従う必要があります。

規則には以下のようなものがあります。

- 制御ステートメントを桁 1 から開始します。
- キーワードと最初のオペランドとの間に、1 つまたは複数のスペースを置きます。
- 2 番目のオペランドを指定するときは、最初のオペランドの後にコロンの (:) と 1 つまたは複数のスペースを置きます。
- 追加の指定を行うには、桁 1 から 5 にブランクを置き、その後にオペランド (複数も可) を指定することにより、制御ステートメントを続けます。
- コメントを指定するには、制御ステートメントの桁 1 にアスタリスク (*) を置きます。制御ステートメントと同じ行にコメントを置くこともできます。その場合は、コメントを始める前に、オペランドの後に少なくとも 1 つのスペースを置く必要があります。
- 複数の変更を単一の制御ステートメントに指定するには、それぞれの追加の指定を別々の行に置いてください。
- ブランク行を追加してはなりません。

制御ステートメントのオペランドのコーディング

このトピックでは、制御ステートメント内にコーディングするオペランドのタイプを示します。

reserved-word

既存の予約語。

user-word

予約語ではない、ユーザー定義の COBOL 語。

コメント

制御ステートメントと同じ行に置くか、または桁 1 にアスタリスクのある別個の行に置くコメント。

COBOL-word

システム名、予約語、またはユーザー定義語のいずれかである、最大 30 文字の語。

制御ステートメントのオペランドのコーディング規則

制御ステートメントのオペランドをコーディングする際には、規則に従います。

規則には以下のようなものがあります。

- `user-word` は、特定の予約語テーブル内の 1 つの `ABBR` ステートメントにのみ使用することができます。
- `ABBR` ステートメントに指定した `reserved-word` を、`RSTR` ステートメントまたは `INFO` ステートメントのどちらでも指定することができます。
- 特定の `reserved-word` は、`ABBR` ステートメントに一度だけ指定することができます。
- 特定の `COBOL-word` は、`RSTR` ステートメントまたは `INFO` ステートメントのどちらかに一度だけ指定することができます。

ABBR ステートメント

`ABBR` ステートメントは、指定された予約語の代替記号を定義します。この記号は、プログラムのコーディング時に使用できます。

```
ABBR reserved-word: user-word [comments]
```

注:

- `user-word` は予約語となり、このステートメントに指定した予約語の代わりに使用することができます。
- `reserved-word` は、元の定義を持つ予約語のままです。
- ソース・リストには、元のソース (それをコーディングしたときの記号) が示されます。
- コンパイラー出力 (他のリスト、一部のメッセージなど) では、予約語が使用されます。

次の例では、`REDEF` および `SEP` は、ソース・プログラムで使用できる省略語となります。ソース・プログラムのコンパイル時に、`REDEFINES` および `SEPARATE` の使用に対する適切な反応が起こります。

```
ABBR REDIFINES: REDEF  
SEPARATE: SEP
```

INFO ステートメント

`INFO` ステートメントは、コンパイラーによってフラグを立てる `COBOL` 語を指定します。また、このステートメントを使用して、ネストされたプログラムの使用を制御することもできます。

```
INFO COBOL-word[(0 | 1)] [comments]
```

1 または 0 を選択して、特定の `COBOL-word` を一度だけ使用できるか、またはまったく使用できないかを示すことができます。

0 FLAGSTD コンパイラー・オプションが有効である場合、指定した `COBOL-word` が使用されると、通知メッセージ 0086 が出されます。

1 指定した `COBOL-word` は一度だけ使用できます。2 度以上使用されると、通知メッセージ 0195 が出されます。

これらのメッセージは、通知 (I) メッセージとして処理されます。コンパイル条件は変更されません。

RSTR ステートメント

RSTR ステートメントは、プログラムで使用できない COBOL 語を指定します。また、このステートメントを使用して、ネストされたプログラムの使用を制御することもできます。

RSTR COBOL-word[(0 | 1)] [comments]

1 または 0 を選択して、特定の COBOL-word を一度だけ使用できるか、またはまったく使用できないかを示すことができます。

0 指定した COBOL-word が使用されると、メッセージ 0084 が出されません。

1 指定した COBOL-word は一度だけ使用できます。2 度以上使用されると、重大メッセージ 0194 が出されます。

以下の予約語は、オプション 1 を使用することでのみ制限できます。

IDENTIFICATION
FD
ENVIRONMENT
DATA
WORKING-STORAGE
PROCEDURE
DIVISION
SECTION
PROGRAM-ID

非 SMP/E JCL の変更および実行

予約語テーブルを作成または変更するには、IGY.V6R1M0.SIGYSAMP 内のサンプル・ジョブ IGYWRWD を使用します。

このサンプル・ジョブは、IGY.V6R1M0.SIGYSAMP 内の IGY8RWRD または IGY8CICS に基づいたメンバーを、予約語ユーティリティへの入力として使用します。これによって、プログラム・オブジェクト IGYCxxxx (xxxx はユーザー識別) が IGY.V6R1M0.SIGYCOMP 内に作成されます。

IGYWRWD の JCL を変更する場合の手順:

1. インストール・システムに合わせて JOB ステートメントを変更します。
2. 必要に応じて、JES ROUTE レコードを追加します。
3. STEP1 の STEPLIB DD ステートメント上のデータ・セット名を変更して、インストール時に使用したコンパイラ・ターゲット・データ・セット名に一致させます。
4. 変更した予約語テーブルをポイントするため、以下のステップのうちの 1 つのみを実行します。
 - //RSWDREAD DD DSN=... 内のデータ・セット名を、変更済み予約語テーブルのデータ・セット名とメンバー名に変更します。

- RSWDREAD DD を //RSWDREAD DD * で置き換え、その行の直後に変更済み予約語テーブルを挿入します。

特定の指示については、ジョブ IGYWRWD 内のコメントを参照してください。

5. STEP3 の SYSLMOD DD ステートメント内のデータ・セット名を変更して、変更済み予約語テーブルを追加する先のデータ・セットの名前に一致させます (SYSLMOD DD ステートメント内のデータ・セット名は、コンパイラー・ターゲット・データ・セットの名前でなければなりません)。さらに、SYSLMOD DD ステートメント内のデータ・セット名の後に、変更済み予約語テーブルの名前を括弧で囲んで指定する必要があります。

IGYWRWD の実行後に、予約語テーブル・ユーティリティーからゼロ以外の戻りコードを受け取った場合は、RSWDPRNT DD ステートメントに指定されている出力データ・セットに出力されたエラー・メッセージを調べて間違いを訂正し、このジョブを再実行してください。

予約語テーブルを作成する JCL の実行

予約語テーブルを作成する JCL には、STEP1、STEP2、および STEP3 があります。

この 3 つのステップでは、それぞれ以下のタスクを実行します。

1. 変更済みテーブルを使用して予約語テーブル・ユーティリティーを実行する。
2. 変更済み予約語テーブルをアSEMBルする。
3. ランタイム・プログラム・オブジェクトをオブジェクト・モジュールから作成する。

このジョブを実行すると、予約語テーブルが作成されます。この新しいテーブルは、ユーザー指定のライブラリーに入れられ、IGYC の後にユーザー指定の 1 から 4 文字の ID を加えた名前が付けられます。

共有ストレージへの Enterprise COBOL モジュールの配置

IGY.V6R1M0.SIGYCOMP 内の再入可能なモジュールはすべて、共有ストレージに入れることができます。

共有ストレージにモジュールを入れるには、以下のようになります。

- データ・セット IGY.V6R1M0.SIGYCOMP を認可します。
- (オプション) IGY.V6R1M0.SIGYCOMP を LNKLST nn 連結に組み込みます。
- システムの IPL 後にコンパイラー・モジュールを LPA に動的に追加します。

z/OS では、プログラム・オブジェクトを LPA にロードできるようにするために IGY.V6R1M0.SIGYCOMP を LNKLST nn 連結に入れる必要はありません。

LNKLST nn 連結に追加しないことを選択した場合は、以下のステップのうちの 1 つを実行して、LPA に含まれていないモジュールを、Enterprise COBOL アプリケーションをコンパイルするステップで使用できるようにする必要があります。

- 非 LPA モジュールを、LNKLST nn 連結内のデータ・セットにコピーする。
- 非 LPA モジュールを、STEPLIB または JOBLIB として使用できるデータ・セットにコピーする。

- I IGY.V6R1M0.SIGYCOMP データ・セット全体を STEPLIB または JOBLIB として使用すると、LPA が検索される前に STEPLIB または JOBLIB からモジュールがロードされるので、モジュールを LPA に置くという目的が果たされなくなります。

別のデータ・セットにコピーされたモジュールは、そのデータ・セット内で、SMP/E によって自動的にサービスされるわけではありません。更新済みモジュールを LNKSTnn データ・セットまたは STEPLIB 内で使用できるようにするためには、Enterprise COBOL にサービスを適用した後で、コピー・ジョブを再実行する必要があります。

モジュールを LPA に入れる方法については、以下の資料を参照してください。

- *z/OS MVS 初期設定およびチューニング ガイド*
- *z/OS MVS 初期設定およびチューニング 解説書*

コンパイラ・フェーズを共有ストレージに置く場合は、サンプル・ジョブ IGYWDOPT を実行してコンパイラ・オプション・デフォルトを変更するときに、対応するフェーズ・オプションを値 OUT でコーディングしてください。詳細については、80 ページの『コンパイラ・オプションのデフォルトの変更』を参照してください。

インストール先でのカタログ式プロシージャの調整

インストール先の用途に応じて、カタログ式プロシージャ IGYWC、IGYWCL、IGYWCLG を調整しなければならない場合があります。

以下の変更を検討してください。

- Enterprise COBOL または 言語環境プログラムのターゲット・データ・セットに IBM 提供のものと異なる接頭部を使用した場合、データ・セット名接頭部を変更する。
- I • LNKST 連結に IGY.V6R1M0.SIGYCOMP、CEE.SCEERUN、および CEE.SCEERUN2 を入れた場合は、STEPLIB DD ステートメントを除去する。
- 正常に実行されるためには GO ステップのデフォルトの領域サイズよりも大きな領域を必要とするプログラムが大部分を占める環境の場合、デフォルトの領域サイズを変更する。
- UNIT= パラメーターを変更する。

付録. Enterprise COBOL 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は、Enterprise COBOL for z/OS のアクセスを支援します。

アクセシビリティ機能

z/OS は、以下のような主要アクセシビリティ機能を備えています。

- 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>) および Web Content Accessibility Guidelines (WCAG) 2.0 (<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/>) 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 の Enterprise COBOL for z/OS 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/doc/kc_help.html#accessibility 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。

ユーザーはまた、IBM Rational® Developer for System z を使用して、z/OS サービスにアクセスすることもできます。

これらのインターフェースへのアクセスに関する情報については、以下の資料を参照してください。

- *z/OS TSO/E Primer* (<http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/ikj4p120>)
- *z/OS TSO/E User's Guide* (<http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/ikj4c240/APPENDIX1.3>)
- *z/OS ISPF User's Guide Volume I* (<http://publib.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/ispzug70>)
- IBM Rational Developer for System z Knowledge Center (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

Enterprise COBOL for z/OS のオンライン製品資料は、IBM Knowledge Center で入手でき、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 のユーザー・インターフェースと連動します。具体的な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) を参照してください。

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。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、

利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. 1996, 2016.

プライバシー・ポリシーに関する考慮事項:

サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品 (「ソフトウェア・オファリング」) では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie はじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項をご確認ください。

この「ソフトウェア・オファリング」は、Cookie もしくはその他のテクノロジーを使用して個人情報を収集することはありません。

この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。

このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、『IBM オンラインでのプライバシー・ステートメント』 (<http://www.ibm.com/privacy/details/jp/ja/>) の『クッキー、ウェブ・ビーコン、その他のテクノロジー』および「IBM Software Products and Software-as-a-Service Privacy Statement」 (<http://www.ibm.com/software/info/product-privacy>) を参照してください。

プログラミング・インターフェース情報

IBM Enterprise COBOL for z/OS では、お客様のインストール済み環境で IBM Enterprise COBOL for z/OS のサービスを使用するプログラムを作成するためのマクロは提供されていません。

重要: IBM Enterprise COBOL for z/OS マクロをプログラミング・インターフェースとして使用しないでください。

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 Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

リソース・リスト

Enterprise COBOL for z/OS COBOL for z/OS の資料

「Enterprise COBOL for z/OS library」には、以下の資料があります。

- | • カスタマイズ・ガイド (SC43-3029)
- | • 言語解説書 (SC43-3031)
- | • プログラミング・ガイド (SC43-3030)
- | • 移行ガイド (GC43-3028)
- | • *Program Directory* (GI13-4526)
- | • *Licensed Program Specifications* (GI13-4532)

ソフトコピー資料

次のコレクション・キットには、Enterprise COBOL およびその他の製品資料が含まれます。これらは <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss> にあります。

- *z/OS Software Products Collection*
- *z/OS and Software Products DVD Collection*

サポート

Enterprise COBOL for z/OS をご使用になっ
ていて問題が生じた場合、最新のサポート情報を提供す
る以下のサイトを参照してください。

www.ibm.com/software/awdtools/cobol/zos/support

関連資料

z/OS ライブラリー資料

「z/OS Internet Library」には、以下の資料があり
ます。

ランタイム・ライブラリー拡張機能

- 共通デバッグ・アーキテクチャー ライブラリ
ー・リファレンス
- 共通デバッグ・アーキテクチャー ユーザーズ・
ガイド
- *DWARF/ELF* エクステンション ライブラリー・
リファレンス

z/OS ISPF

- ユーザーズ・ガイド 第 1 巻

z/OS 言語環境プログラム

- 概念
- カスタマイズ
- デバッグのガイド
- プログラミング・ガイド
- プログラミング・リファレンス
- ランタイム・メッセージ
- ランタイム マイグレーション・ガイド
- *ILC* (言語間通信) アプリケーションの作成

z/OS MVS

- *Initialization and Tuning Guide*
- *Initialization and Tuning Reference*

z/OS SMP/E

- ユーザーズ・ガイド
- *Reference*

z/OS TSO/E

- 入門
- ユーザーズ・ガイド

CICS Transaction Server for z/OS

「CICS」には、以下の資料があります。

- アプリケーション・プログラミング・ガイド
- アプリケーション・プログラミング・リファレ
ンス
- カスタマイズ・ガイド
- 外部インターフェース・ガイド

DB2 for z/OS

「DB2 for z/OS Library」には、以下の資料があり
ます。

- *Installation and Migration Guide*
- *Internationalization Guide*

デバッグ・ツール

「Debug Tool for z/OS」には、以下の資料があります。

- リファレンスおよびメッセージ
- ユーザーズ・ガイド

他の資料

- *American National Standard ANSI INCITS 23-1985, Programming Languages - COBOL*, およびその改訂版 *ANSI INCITS 23a-1989, Programming Languages - Intrinsic Function Module for COBOL* および *ANSI INCITS 23b-1993, Programming Language - Correction Amendment for COBOL*
- *International Standard ISO 1989:1985, Programming languages - COBOL*, およびその改訂版 *ISO/IEC 1989/AMD1:1992, Programming languages - COBOL: Intrinsic function module* および *ISO/IEC 1989/AMD2:1994, Programming languages - Correction and clarification amendment for COBOL*
- *DBCS 日本語配列プログラム (DBCSOS)*, SH88-0171 (N:SH18-0144)

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アクセシビリティ

キーボード・ナビゲーション 91

本書の 92

Enterprise COBOL for z/OS の 91

z/OS の使用 91

アスタリスク (*), 固定コンパイラー・オプションを示す 19

エラー・メッセージ

フラグ 37

オーバーライドできないコンパイラー・オプション, アスタリスク (*) で示す 19

オブジェクト・コード, 再入可能 56

[カ行]

カスタマー・サポート 97

カスタマイズ

インストール・ジョブ

Enterprise COBOL 79

計画 1

コンパイラー・オプション 17, 80

キーボード・ナビゲーション 91

キーワード x

行番号のシーケンス検査 59

共有ストレージ

計画 6

コンパイラー・フェーズ 6, 8

Enterprise COBOL モジュールの配置 88, 89

計画ワークシート

説明 x

IGYCDOPT (コンパイラー・オプション) 4

IGYCDOPT (コンパイラー・フェーズ) 11

構文検査 27

構文図の読み方 ix

構文表記法

アスタリスク (*) 19

反復矢印 x

COBOL キーワード x

固定コンパイラー・オプション

アスタリスク (*) で示す 19

固定コンパイラー・オプション (続き)

コンパイラー・オプションの固定 2

コンパイラー・オプション

計画ワークシート 4

固定

アスタリスク (*) で示す 19

コンパイラー・オプションの固定

2

説明

ADATA 19

ADEXIT 19

ADV 20

AFP 21

ALLOWCBL 21

ARCH 22

ARITH 24

AWO 24

BLOCK0 25

BUF 25

COMPILE 27

COPYRIGHT 28

CURRENCY 28

DATA 30

DBCS 31

DBCSXREF 31

DECK 32

DIAGTRUNC 33

DISPSIGN 33

DLL 34

DYNAM 35

EXPORT 35

FASTSRT 36

FLAG 37

FLAGSTD 38

HGPR 40

INEXIT 40

INTDATE 41

LANGUAGE 42

LIBEXIT 43

LINECNT 43

LIST 44

LITCHAR 44

MAP 45

MAXPCF 45

MDECK 47

MSGEXIT 47

NAME 48

NUM 49

NUMCLS 49

NUMPROC 50

OBJECT 51

コンパイラー・オプション (続き)

説明 (続き)

OFFSET 52

OPTIMIZE 52

OUTDD 53

PGMNAME 54

PRTEXT 54

QUALIFY 55

RENT 56

RMODE 57

RULES 58

SEQ 59

SERVICE 59

SOURCE 60

SPACE 60

SQL 61

SQLIMS 62

SQLSSCID 61

SSRANGE 63

STGOPT 64

SUPPRESS 64

TERM 65

TEST 65

THREAD 67

TRUNC 68

VBREF 69

VLR 70

VSAMOPENFS 71

WORD 71

XMLPARSE 73

XREFOPT 73

ZONECHECK 74

ZONEDATA 75

ZWB 77

デフォルト値 2

デフォルトの設定 2, 80

変更 3, 80

矛盾するオプション 17

コンパイラー・フェーズ

共有ストレージに置く 7

固定フェーズ 7

説明

DGEN 9

DIAG 9

DMAP 9

FGEN 9

INIT 9

LIBR 9

LSTR 10

MSGT 10

OPTM 10

コンパイラー・フェーズ (続き)

説明 (続き)

PGEN 10

RCTL 10

RWT 11

SCAN 11

SIMD 11

XREF 11

デフォルト 8

変更 3

マクロ・ワークシート 11

INOUT パラメーター 8

[サ行]

再入可能オブジェクト・コード 56

サポート 97

サンプル・インストール・ジョブ 3

支援テクノロジー 92

指標検査 63

常駐モード 57

資料 97

身体障がい 91

製品サポート 97

製品登録 15

前書き ix

添え字検査 63

[タ行]

縦に重ねられた語 ix

デバッグ・ツール 66

デフォルト値

コンパイラー・オプション 2

コンパイラー・フェーズ 8

デフォルト予約語テーブル 14

[ナ行]

任意指定の語 ix

ネストされたプログラム 13

[ハ行]

必須の語 ix

フェーズ、コンパイラー

共有ストレージに置く 7

デフォルト 8

変更 3

マクロ・ワークシート 11

本製品のアクセシビリティ機能 91

[マ行]

マクロ

IGYCDOPT (コンパイラー・オプション)

計画ワークシート 4

構文形式 3

IGYCDOPT (コンパイラー・フェーズ)

計画ワークシート 11

構文形式 3

メッセージ、フラグ 37

[ヤ行]

ユーザー出口ルーチン

ADEXIT コンパイラー・オプション

19

INEXIT コンパイラー・オプション

40

LIBEXIT コンパイラー・オプション

43

MSGEXIT コンパイラー・オプション

47

PRTEXIT コンパイラー・オプション

54

予約語テーブル

計画 13

作成または変更 82

代替テーブルの指定 71

内容 14

ネストされたプログラム 13

Enterprise COBOL と一緒に提供される

IGYCCICS (CICS) 13

IGYCRWT (デフォルト) 13

A

ADATA コンパイラー・オプション 19

ADEXIT コンパイラー・オプション 19

ADV コンパイラー・オプション 20

AFP コンパイラー・オプション 21

ALOWCBL コンパイラー・オプション

21

ARCH コンパイラー・オプション 22

ARITH コンパイラー・オプション 24

AWO コンパイラー・オプション 24

B

BLOCK0 コンパイラー・オプション 25

BUF コンパイラー・オプション 25

C

CBL ステートメント 21

CICS 予約語テーブル 14

COMPILE コンパイラー・オプション 27

COPYRIGHT コンパイラー・オプション

28

CURRENCY コンパイラー・オプション

28

D

DATA コンパイラー・オプション 30

DBCS コンパイラー・オプション 31

DBCSXREF コンパイラー・オプション

31

DECK コンパイラー・オプション 32

DGEN フェーズ 9

DIAG フェーズ 9

DIAGTRUNC コンパイラー・オプション

33

DISPSIGN コンパイラー・オプション 33

DLL コンパイラー・オプション 34

DMAP フェーズ 9

DYNAM コンパイラー・オプション 35

E

Enterprise COBOL

使用可能 15

使用不可 15

ジョブの変更 79

EXPORT コンパイラー・オプション 35

F

FASTSRT オプション 36

FGEN フェーズ 9

FLAG コンパイラー・オプション 37

FLAGSTD コンパイラー・オプション 38

H

HGPR コンパイラー・オプション 40

I

IGYCCICS (CICS 予約語テーブル) 14

IGYCDOPT

計画ワークシート 4

AMODE 31 および RMODE ANY でのリンク 2

IGYCOPT

構文形式 3

IGYCRWT (デフォルト予約語テーブル)
14
INEXIT コンパイラー・オプション 40
INIT フェーズ 9
INTDATE コンパイラー・オプション 41

L

LANGUAGE コンパイラー・オプション
42
LIBEXIT コンパイラー・オプション 43
LIBR フェーズ 9
LINECNT コンパイラー・オプション 43
LIST コンパイラー・オプション 44
LITCHAR コンパイラー・オプション 44
LSTR フェーズ 10

M

MAP コンパイラー・オプション 45
MAXPCF コンパイラー・オプション 45
MDECK コンパイラー・オプション 47
MSGEXIT コンパイラー・オプション 47
MSGT フェーズ 10

N

NAME コンパイラー・オプション 48
NUM コンパイラー・オプション 49
NUMCLS コンパイラー・オプション 49
NUMPROC コンパイラー・オプション
50

O

OBJECT コンパイラー・オプション 51
OFFSET コンパイラー・オプション 52
OPTIMIZE コンパイラー・オプション
52
OPTM フェーズ 10
OUTDD コンパイラー・オプション 53

P

PGEN フェーズ 10
PGMNAME コンパイラー・オプション
54
PROCESS (CBL) ステートメント 21
PRTEXIT コンパイラー・オプション 54

Q

QUALIFY コンパイラー・オプション 55

R

RCTL フェーズ 10
RENT コンパイラー・オプション 56
RMODE コンパイラー・オプション 57
RULES コンパイラー・オプション 58
RWT フェーズ 11

S

SCAN フェーズ 11
SEQUENCE コンパイラー・オプション
59
SERVICE コンパイラー・オプション 59
SIMD フェーズ 11
SOURCE コンパイラー・オプション 60
SPACE コンパイラー・オプション 60
SQL コンパイラー・オプション 61
SQLCCSID コンパイラー・オプション
61
SQLIMS コンパイラー・オプション 62
SSRANGE コンパイラー・オプション
63
STGOPT コンパイラー・オプション 64
SUPPRESS コンパイラー・オプション
64
SYSLIN 51
SYSOUT 53
SYSPUNCH 32
SYSTEM 65

T

TERM コンパイラー・オプション 65
TEST コンパイラー・オプション 65
THREAD コンパイラー・オプション 67
TRUNC コンパイラー・オプション 68

V

VBREF コンパイラー・オプション 69
VLR コンパイラー・オプション 70
VSAMOPENFS コンパイラー・オプション
71

W

WORD コンパイラー・オプション 71

X

XMLPARSE コンパイラー・オプション
73
XREF コンパイラー・オプション 73
XREF フェーズ 11

XREFOPT オプション 73

Z

ZONECHECK コンパイラー・オプション
74
ZONEDATA コンパイラー・オプション
75
ZWB コンパイラー・オプション 77



プログラム番号: 5655-EC6

Printed in Japan

SC43-3029-04



日本アイ・ビー・エム株式会社

〒103-8510 東京都中央区日本橋箱崎町19-21