Enterprise COBOL for z/OS

カスタマイズ・ガイド

バージョン 3 リリース 4

GC88-9119-03
(英文原典：GC27-1410-04)

Enterprise COBOL for z/OS

カスタマイズ・ガイド

バージョン 3 リリース 4

GC88-9119-03
(英文原典：GC27-1410-04)

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、 75 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 Enterprise COBOL for z/OS バージョン 3 リリース 4 (プログラム番号 5655-G53)、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。

資料のご注文方法については、http://www.ibm.com/jp/manuals の「ご注文について」をご覧ください (URL は、変更になる場合があります)。

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

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

http://www.ibm.com/jp/manuals/main/mail.html

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

http://www.ibm.com/jp/manuals/ の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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： GC27-1410-04
Enterprise COBOL for z/OS
Customization Guide
Version 3 Release 4

発　行： 日本アイ・ビー・エム株式会社

担　当： ナショナル・ランゲージ・サポート

第1刷 2005.9

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財) 日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

目次

図

表

本書について

本書は、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 を計画し、カスタマイズする際に必要となる情報について説明しています。さらに、本書を用いて、インストール先で Enterprise COBOL の利点が十分に生かされているかを評価することができます。

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

本書を使用して Enterprise COBOL を正常にカスタマイズするためには、Enterprise COBOL および使用中のシステムの操作環境を理解している必要があります。

構文図の読み方

本書では、以下の規則に従って、コンパイラー・オプションの構文を図示しています。

- 構文図は、左から右、上から下へと線をたどってください。以下の表は、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—必須項目————————————————————————————►◄
```

- 任意指定項目は、幹線の下に示されています。

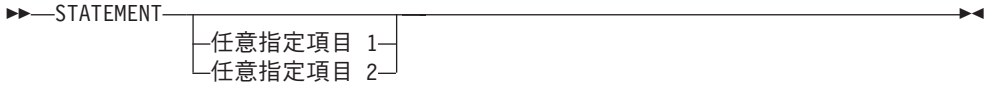
```
►►—STATEMENT———————————————————————————————————►◄
                └任意指定項目┘
```

- 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。

```
                ┌デフォルト項目┐
►►—STATEMENT———┤必須選択項目 1├———————————————————————►◄
                └必須選択項目 2┘
```

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 (たとえば、PRINT)。キーワードは示されたとおりに入力しなければなりません。変数は、すべて小文字で示されています (たとえば、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。

マクロ計画ワークシートの使用

本書には、Enterprise COBOL のカスタマイズを準備するために使用できる計画ワークシートがあります。これらのワークシートに記入することにより、どのオプションの IBM 提供のデフォルト値を変更する必要があるかを容易に判別することができます。これらのワークシートは、実際に 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、それぞれ若干異なる場合があります。それぞれの欄のヘッディングの定義については、以下を参照してください。ワークシートは、4 ページと 10 ページにあります。

オプション
「オプション」欄は、特定のインストール・マクロに含まれるオプションを識別します。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
「固定」欄は、アプリケーション・プログラマーがオーバーライドすることのできないオプションを識別するための欄です。固定したいオプションについてのみ、「**固定の場合 * を記入**」欄にアスタリスク [*] を記入してください。

選択
「選択」欄は、各オプションに関連付ける値を識別するための欄です。各オプションに割り当てたい値を記入してください。「構文の説明」欄を使用して、適切な値を選択する際に役立つオプションについての特定の情報を探し出すことができます。

IBM 提供のデフォルト
「IBM 提供のデフォルト」欄には、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が示されています。IBM 提供のデフォルトが、インストール先で必要な値と同じである場合は、そのマクロのそのオプションを変更する必要はありません。

構文の説明
「構文の説明」欄には、そのオプションの構文図および固有の情報が記載されているページが示されています。

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。

アクセシビリティ

アクセシビリティ機能は、運動障害または視覚障害など身体に障害を持つユーザーがソフトウェア・プロダクトを快適に使用できるようにサポートします。 z/OS のアクセシビリティ機能では、Enterprise COBOL へのアクセシビリティを提供しています。

z/OS のアクセシビリティの主要機能により、ユーザーは以下のことができるようになります。

- スクリーン・リーダー (読み上げソフトウェア)、および画面拡大ソフトウェアなどの支援テクノロジー製品の使用
- キーボードのみを使用して、特定の機能、または同等の機能を実行します。
- 色、明度、およびフォント・サイズなどの表示属性をカスタマイズします。

支援機能の使用

支援テクノロジー製品は、z/OS のユーザー・インターフェースで作動します。固有のガイダンス情報については、z/OS インターフェースへのアクセスに使用する支援テクノロジー製品の資料を参照してください。

ユーザー・インターフェースのキーボード・ナビゲーション

ユーザーは、TSO/E または ISPF を使用して z/OS ユーザー・インターフェースにアクセスできます。TSO/E および ISPF インターフェースへのアクセスについての情報は、以下の資料を参照してください。

- *z/OS TSO/E 入門*
- *z/OS TSO/E ユーザーズ・ガイド*
- *ISPF ユーザーズ・ガイド 第 1 巻*

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

本書のアクセシビリティ

本書は、英語版 XHTML 形式で、IBM Problem Determination and Deployment Tools information center (http://publib.boulder.ibm.com/infocenter/pdthelp/index.jsp) から入手でき、スクリーン・リーダー (読み上げソフトウェア) を使用する視覚障害者がアクセスできます。

スクリーン・リーダー が構文図、ソース・コード例、およびピリオドやコンマといった画像記号を含むテキストを正確に読み取ることができるようにするには、句読点をすべて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。

JAWS for Windows を使用している場合、アクセス可能な構文図へのリンクが機能しない可能性があります。その場合、IBM ホームページ・リーダーを使用して、アクセス可能な構文図を読み取ってください。

変更の要約

ここでは、IBM COBOL (OS/390 および VM 版) バージョン 2 リリース 2 以降に Enterprise COBOL for z/OS 製品およびマニュアルに加えられた主要な変更を示します。技術上の変更は、テキストの左マージンに変更バーを付けてマークしています。

Enterprise COBOL に加えられた主要な変更

バージョン 3 リリース 4、2005 年 7 月

- 一部の COBOL データ項目サイズの制限のいくつかが、例えば以下のように大幅に引き上げられました。
  - 最大データ項目サイズが、16 MB から 128 MB に引き上げられました。
  - 最大 PICTURE シンボル複製が、134,217,727 に引き上げられました。
  - 最大 OCCURS 整数が、134,217,727 に引き上げられました。

  このサポートにより、以下のような大量のデータがあるプログラミングを、容易に実行できるようになりました。
  - DB2 BLOB および CLOB データ・タイプを使用する DB2/COBOL アプリケーション
  - 大規模な XML 文書を解析、または生成する COBOL XML アプリケーション
- 国別 (Unicode UTF-16) データのサポートが強化されました。 以下の何種類かの追加データ項目が、暗黙的に、または明示的に USAGE NATIONAL として記述できるようになりました。
  - 外部 10 進数 (*国別 10 進数*) 項目
  - 外部浮動小数点 (*国別浮動小数点*) 項目
  - 数字編集項目
  - 国別編集項目
  - グループ (*国別グループ*) 項目、GROUP-USAGE NATIONAL 文節によりサポート
- 多くの COBOL 言語エレメントが、以下の新規種類の UTF-16 データをサポートし、また国別データの処理を新規にサポートするようになりました。
  - USAGE NATIONAL (国別 10 進数および国別浮動小数点) がある数値データは、算術演算、および数値オペランドをサポートする任意の言語構造体で使用できます。
  - USAGE NATIONAL がある編集データは、既存の編集タイプと同じ言語構造体でサポートされます。これには、移動と関連した操作の編集および編集解除が含まれます。
  - すべての国別データを含むグループ項目は、GROUP-USAGE NATIONAL 文節を使用して定義できます。これは、ほとんどの言語構造体内で基本項目として

作動するグループになります。このサポートにより、STRING、UNSTRING、および INSPECT などのステートメントで国別グループを容易に使用できます。

– XML GENERATE ステートメントは、国別グループによるデータ項目受信をサポートし、国別編集項目、USAGE NATIONAL の数字編集項目、国別 10 進数項目、国別浮動小数点項目、および国別グループ項目それぞれのデータ項目送信をサポートします。

– 組み込み関数 NUMVAL および NUMVAL-C は、国別リテラルまたは国別データ項目を引数とすることができます。

これらの新しい国別データ機能を使用することで、すべてのアプリケーション・データに Unicode のみを使用する COBOL プログラムの開発が現実となりました。

- REDEFINES 文節が拡張されたことによって、レベル 01 ではないデータ項目の場合、記入項目のサブジェクトを再定義されるデータ項目より大きくできるようになりました。
- 新しいコンパイラー・オプションの MDECK により、library-processing ステートメントからの出力をファイルに書き込むことができます。
- DB2 コプロセッサー・サポートが拡張されました。COBOL ゾーン 10 進数 (USAGE DISPLAY SIGN LEADING SEPARATE) データ項目、および数値 Unicode (USAGE NATIONAL SIGN LEADING SEPARATE) データ項目を、DB2 のホスト変数として使用できます。「@」、「#」、および「$」を EXEC SQL INCLUDE ファイル名で使用できます。XREF は、改良されています。
- クラス national のデータ項目の VALUE 文節にあるリテラルに、英数字を使用できます。

バージョン 3 リリース 3、2004 年 2 月

- XML サポートが拡張されました。新しいステートメント XML GENERATE は、COBOL データ・レコードの内容を XML フォーマットに変換します。XML GENERATE は、Unicode UTF-16 あるいは幾つかの単一バイト EBCDIC コード・ページまたは ASCII コード・ページの 1 つにエンコードされた XML 文書を作成します。
- コンパイラーが、デバッグ・ツールの新規または改良されたフィーチャーをサポートするように拡張されました。
  – COBOL SYSDEBUG ファイル使用の際のパフォーマンスが向上しました。
  – 国別データを使用するプログラムのデバッグがより容易になりました。各国語データを、定様式ダンプで表示する場合または Debug Tool LIST コマンドを使用して表示する場合、そのデータは、CODEPAGE コンパイラー・オプションで指定されたコード・ページを使用して自動的に EBCDIC 表記に変換されます。Debug Tool MOVE コマンドを使用して、国別データ項目に値を割り当てることができます。さらに、国別データ項目を、グループ・データ項目へ、あるいはグループ・データ項目から移動することができます。国別データは、IF および EVALUATE などの Debug Tool 条件付きコマンドで被比較数として使用できます。
  – オブジェクト指向の構文を含む COBOL-Java 混合アプリケーション、COBOL クラス定義、および COBOL プログラムをデバッグすることができます。

- 統合された DB2 コプロセッサーの使用の際に、DB2 バージョン 8 SQL 機能がサポートされます。
- COBJVMINITOPTIONS 環境変数にオプションを指定する構文が、変更されました。

バージョン 3 リリース 2、2002 年 9 月

- OPTIMIZE コンパイラー・オプションは、現在 Java インターオペラビリティーのためのオブジェクト指向構文を含むプログラムで完全にサポートされています。
- COBOL プログラムで SQL ステートメントを使用する際の、 Unicode および EBCDIC コード・ページの指定方法が拡張されました。
  - Unicode ホスト変数 (USAGE NATIONAL を使用して宣言) を SQL ステートメントで使用する際、 SQL DECLARE VARIABLE ステートメントを使用する変数では、Unicode CCSID (1200) を明示的に指定する必要はありません。
  - CODEPAGE コンパイラー・オプションで有効にされた CCSID は現在、ホスト変数用の明示的な SQL DECLARE VARIABLE ステートメントなしで、SQL ステートメントで使用される単一バイトあるいは 2 バイトの EBCDIC ホスト変数に適用されます。

バージョン 3 リリース 1、2001 年 11 月

- POSIX スレッドおよび非同期シグナルの許容のサポートにより、1 つのプロセス内でマルチスレッドで実行される複数の COBOL プログラムをアプリケーションに組み込むことができるようになりました。スレッドおよび非同期シグナルの許容をサポートする新しいコンパイラー・オプション。
  - THREAD
- Unicode の基本的なランタイム・サポートを提供するために、新たな国別データ型、国別リテラル、組み込み関数、および以下に示す 2 つのコンパイラー・オプションが追加されました。リテラルおよび PICTURE 文節で使用される記号の相互協調処理をサポートする新しいコンパイラー・オプション。
  - NSYMBOL

  国別データ型およびリテラルをサポートする新しいコンパイラー・オプション。
  - CODEPAGE
- オブジェクト指向構文により、COBOL プログラムと Java プログラムとの間の相互協調処理が容易になりました。
- 基本的な XML 機能が COBOL に追加されました。
- CICS 変換プログラムがコンパイラーに統合されました。統合 CICS 変換プログラムをサポートする新しいコンパイラー・オプション。
  - CICS

以前の COBOL コンパイラーに対する変更の履歴については、「*Enterprise COBOL for z/OS コンパイラーおよびランタイム 移行ガイド*」を参照してください。

第 1 章 Enterprise COBOL のカスタマイズについての計画

この章では、Enterprise COBOL のカスタマイズを計画する際に必要な情報について説明します。この章の内容は次のとおりです。
- インストール後の変更 - なぜカスタマイズが必要か
- コンパイラーのデフォルト値を変更する計画
- Enterprise COBOL を共用ストレージに置く計画
- 追加の予約語テーブルを作成する計画

IBM デバッグ・ツールをインストールする場合は、デバッグ・ツールのモジュールを共用ストレージに置くかどうか、また、デバッグ・ツール用に作動するように CICS® 環境をセットアップするかどうかを決定することができます。

この章の内容は、カスタマイズを計画する際に役立ちます。実際のカスタマイズ手順については、57 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。

この章には、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あります。本書の計画ワークシートの説明については、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、インストール先のアプリケーション・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 をインストールすると、コンパイラー・オプションおよびフェーズの IBM 提供のデフォルト、および予約語テーブルが提供されます。Enterprise COBOL をカスタマイズして、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さらに満たすことができます。

Enterprise COBOL をインストールした後、以下のことができます。
- 追加の予約語テーブルを作成する
- コンパイラー・オプションを固定する
- コンパイラー・オプションのデフォルト値を変更する
- コンパイラー・フェーズの常駐値を変更する
- COBOL の Unicode サポートをカスタマイズする

以下のセクションでは、Enterprise COBOL をインストール先に合わせてカスタマイズするために行う必要のある計画について説明します。

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 プログラムで設定します (4 ページの表 1 および 10 ページの表 2 を参照)。デフォルト・オプション・モジュール IGYCDOPT は、インストール時に AMODE(31) および RMODE(ANY) を指定してリンク・エディットされます。

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.V3R4M0.SIGYMAC にもアクセスする必要があります。

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・フェーズを共用ストレージに置くかを指定することです。Enterprise COBOL のインストール時に提供された 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、それらの値をインストール先のプログラマーの要件に合わせて変更することもできます。また、インストール先のアプリケーション・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: IGY.V3R4M0 の高位修飾子は、Enterprise COBOL インストール時に変更されている場合があります。

共用システム・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、5 ページの『コンパイラー・フェーズを共用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・オプションを固定する目的

Enterprise COBOL では、インストール先の固有のプログラミング標準を設定することができます。たとえば、多くのインストール先では RENT を望ましいコンパイラー・オプションとして選択しますが、RENT の使用を強制する容易な方法はありません。

Enterprise COBOL の場合、IGYCDOPT プログラムを使用して、オプションを固定すること、およびコンパイル時にそのオプションを変更 (つまりオーバーライド) できないことを指定することができます。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しても失敗し、ゼロ以外のコンパイラー戻りコードが付いた診断メッセージが戻されます。

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う回する必要が生じることがあります。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センブルします。コンパイル時に、必要な IGYCDOPT モジュールを含む JOBLIB または STEPLIB を使用すれば、固定されたオプションをう回することができます。たとえば、OPT (OPTIMIZE) オプションを固定する (COBOL コンパイラーが常に最適化オブジェクト・コードを生成することを意味する) ことを選択した場合、あるアプリケーションをこの要件の対象から除外する必要が生じたときは、このオプションからアスタリスク・パラメーターを除いてから、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、結果として生じた IGYCDOPT モジュールを、コンパイル時に JOBLIB または STEPLIB としてアクセスされる一時ライブラリーに入れてください。

サンプル・インストール・ジョブ

Enterprise COBOL 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ために使用できる 2 つのサンプル・インストール・ジョブがあります (これらのジョブに変更を加えてから使用してください)。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。

これらのジョブは、COBOL サンプル・データ・セット IGY.V3R4M0.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、図 1 に示されている 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、各構文図の下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

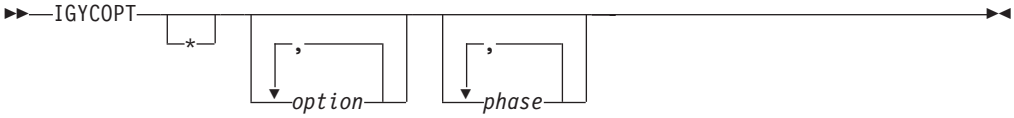


図 1. IGYCOPT コンパイラー・オプション / フェーズ・マクロの構文形式

コンパイラー・オプション用の IGYCDOPT ワークシート

以下のワークシートは、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 (10 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:

1. 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、このオプションを固定すること、およびアプリケーション・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いことを指定することになります。
2. ALOWCBL、DBCSXREF、LVLINFO、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」欄はブランクになっています。
3. ADEXIT、INEXIT、LVLINFO、LIBEXIT、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。
4. 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。

表 1. コンパイラー・オプション用の IGYCDOPT ワークシート

Compiler option	固定の場合 * を記入	選択を記入	IBM 提供のデフォルト	構文の説明
ADATA=	____	_____	**NO**	15
ADEXIT=	____	_____		15
ADV=	____	_____	**YES**	15
ALOWCBL=	____	_____	**YES**	16
ARITH=	____	_____	**COMPAT**	17
AWO=	____	_____	**NO**	17
BUF=	____	_____	**4K**	17
CICS=	____	_____	**NO**	17
CODEPAGE=	____	_____	**1140**	17
COMPILE=	____	_____	**NOC(S)**	19
CURRENCY=	____	_____	**NO**	20
DATA=	____	_____	**31**	21
DATEPROC=	____	_____	**NO**	22
DBCS=	____	_____	**Yes**	23
DBCSXREF	____	_____	**NO**	23
DECK=	____	_____	**NO**	24
DIAGTRUNC=	____	_____	**NO**	24
DLL=	____	_____	**NO**	25
DYNAM=	____	_____	**NO**	25
EXPORTALL=	____	_____	**NO**	26
FASTSRT=	____	_____	**NO**	28
FLAG=	____	_____	**(I,I)**	27
FLAGSTD=	____	_____	**NO**	26
INTDATE=	____	_____	**ANSI**	30
INEXIT=	____	_____		30
LANGUAGE=	____	_____	**ENGLISH**	31
LIB=	____	_____	**NO**	32
LIBEXIT=	____	_____		32

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

Compiler option	固定の場合 * を記入	選択を記入	IBM 提供のデフォルト	構文の説明
LINECNT=	____	_____	**60**	32
LIST=	____	_____	**NO**	33
LITCHAR=	____	_____	**QUOTE**	33
LVLINFO=	____	_____		34
MAP=	____	_____	**NO**	34
MDECK=	____	_____	**NO**	35
NAME=	____	_____	**NO**	35
NSYMBOL=	____	_____	**NATIONAL**	36
NUM=	____	_____	**NO**	36
NUMCLS=	____	_____	**PRIM**	37
NUMPROC=	____	_____	**NOPFD**	37
OBJECT=	____	_____	**YES**	38
OFFSET=	____	_____	**NO**	39
OPT=	____	_____	**NO**	39
OUTDD=	____	_____	**SYSOUT**	40
PGMNAME=	____	_____	**COMPAT**	40
PRTEXIT=	____	_____		41
SQL=	____	_____	**NO**	45
RENT=	____	_____	**YES**	42
RMODE=	____	_____	**AUTO**	43
SEQ=	____	_____	**YES**	43
SIZE=	____	_____	**MAX**	44
SOURCE=	____	_____	**YES**	45
SPACE=	____	_____	**1**	45
SSRANGE=	____	_____	**NO**	46
TERM=	____	_____	**NO**	47
TEST=	____	_____	**NO**	47
THREAD=	____	_____	**NO**	47
TRUNC=	____	_____	**STD**	50
VBREF=	____	_____	**NO**	51
WORD=	____	_____	***NO**	52
XREFOPT=	____	_____	**YES**	53
YRWINDOW=	____	_____	**1900**	53
ZWB=	____	_____	**YES**	54

コンパイラー・フェーズを共用ストレージに置く計画

いくつかのロード・モジュールをリンク・パック域に常駐させることにより、Enterprise COBOL プログラムが実行されるとき、またはそれらのモジュールが共用されるときのモジュール検索を最小限に押さえることができます。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。*共用ストレージ* という用語は、一般に、リンク・パック域 (LPA)、拡張リンク・パック域 (ELPA)、または修正リンク・パック域 (MLPA) を指します。以降のセクションでは、特に断りのない限り、*リンク・パック域* という用語はこれら 3 つすべてに該当します。

コンパイラー・フェーズを共用ストレージに置く目的

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ー・モジュールは、z/OS マシンの共

用ストレージに置くことができます。IGYCRCTL および IGYCSIMD 以外のすべてのコンパイラー・フェーズには、RMODE(ANY) および AMODE(31) があります。IGYCRCTL および IGYCSIMD には RMODE 24 があるので、これらは、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。共用ストレージは、各仮想アドレス・スペースのための同一の 1 つのストレージ域です。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は共用することができ、それぞれのユーザー領域にロードする必要はありません。情報を共用することによって、より多くのスペースがコンパイラー作業域のために使用可能になります。

注: Enterprise COBOL、COBOL (MVS および VM 版)、COBOL (OS/390 および VM 版)、および VS COBOL II の各コンパイラーは同じモジュール名を使用しています。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 に置くことができるフェーズ・セットは 1 つだけです。

IGYCDOPT プログラムにより、各コンパイラー・フェーズをどこにロードするか、つまりユーザー領域内 (IN) にロードするか領域外 (OUT) にロードするかを指定することができます。コンパイラー・フェーズを MLPA に置くと、コンパイラーは、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用ストレージに置く必要があります。コンパイラーは、この情報を用いて、システムがコンパイラー・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。フェーズを共用ストレージに置く方法については、 77 ページの『関連資料』のリストを調べて、それぞれの環境用の「*初期設定およびチューニング*」 のマニュアルを参照してください。

以下の 4 つのフェーズを共用ストレージ域に置くことをお勧めします。

IGYCRCTL
コンパイル中、ユーザー領域に常駐します。

IGYCSIMD
コンパイル中、ユーザー領域に常駐します。

IGYCPGEN
サイズが最も大きい 2 つのコンパイラー・フェーズの 1 つです。

IGYCSCAN
サイズが最も大きい 2 つのコンパイラー・フェーズのもう一方です。

どのコンパイラー・フェーズを共用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、フェーズを共用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。十分な共用ストレージが使用可能でないなら、コンパイル時に常にユーザー領域に常駐する 2 つのフェーズ、IGYCRCTL と IGYCSIMD を優先させてください。十分な共用ストレージが使用可能でないときは、さらに、最大コンパイラー・フェーズである IGYCPGEN と IGYCSCAN も優先させてください。

共用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、共用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。IGYCPGEN および IGYCSCAN コンパイラー・フェーズは、2 番目に大きいコンパイラー・フェーズよりも、およそ 250KB 大きくなります。共用ストレージにコンパイラー・フェーズを置くと、760 KB の最小領域サイズを使用してコンパイルする場合には、かなりの影響があります。

コンパイラー・フェーズおよびそのデフォルト

各コンパイラー・フェーズをユーザー領域の内側と外側のどちらにロードするかを示すために、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目的、または変更しない目的については、5 ページの『コンパイラー・フェーズを共用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。コンパイラーは、SIZE オプションに指定された値を使用して、フェーズ用のストレージを予約します。

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、そのフェーズを共用システム域に置くことができます。しかし、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 の指定のある最大のフェーズを収容するの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 などの共用システム域に置かれている必要があります。

IGYCASM1

アセンブリー 1 フェーズです。アセンブリー 1 フェーズでは、オブジェクト・モジュールのストレージを判別し、永続レジスターおよび一時レジスターを割り振り、データおよびプロシージャー参照のアドレス可能度を最適化します。さらに、データ域のオブジェクト・テキストも作成します。

構文

IGYCASM2

アセンブリー 2 フェーズです。アセンブリー 2 フェーズでは、オブジェクト・プログラムの準備を完了し、デバッグ機能用のオブジェクト・テキスト、リスト、穿孔データ・セット、およびテーブルを作成します。

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ー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

IGYCOPTM

最適化フェーズです。最適化フェーズでは、PERFORM ステートメントを再構成し、重複した計算を除去します。

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

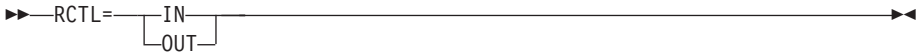
プロシージャー生成フェーズです。プロシージャー生成フェーズでは、すべてのプロシージャー・ソース動詞にコードを提供します。

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

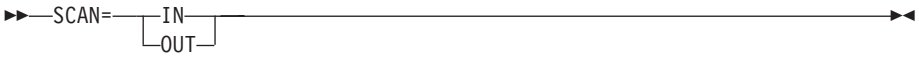
標準の予約語テーブルです。

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。

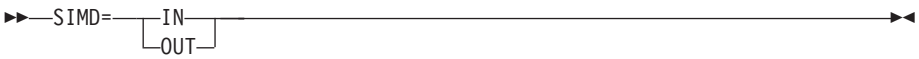
構文



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ー名を 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

以下のワークシートは、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。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、7 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. コンパイラー・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
ASM1=	**IN** / OUT	7
ASM2=	**IN** / OUT	7
DIAG=	**IN** / OUT	8
DMAP=	**IN** / OUT	8
FGEN=	**IN** / OUT	8
INIT=	**IN** / OUT	8
LIBR=	**IN** / OUT	8
LSTR=	**IN** / OUT	8
MSGT=	**IN** / OUT	9
OPTM=	**IN** / OUT	9
OSCN=	**IN** / OUT	9
PGEN=	**IN** / OUT	9
RCTL=	**IN** / OUT	6
RWT=	**IN** / OUT	9
SCAN=	**IN** / OUT	6
SIMD=	**IN** / OUT	10
XREF=	**IN** / OUT	10

追加の予約語テーブルを作成する計画

Enterprise COBOL をインストールすると、以下の予約語テーブルにアクセスすることができます。

- IGYCRWT - システム全体に提供されるデフォルトの予約語テーブル。
- IGYCCICS - 代替予約語テーブルとして提供される CICS 固有の予約語テーブル (『CICS 予約語テーブル (IGYCCICS)』を参照)。

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

以下の目的で、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予約語 (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てください。予約語テーブルの変更は、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、61 ページの『予約語テーブルを作成または変更する』を参照してください。

Enterprise COBOL で提供される予約語テーブル

以下の予約語テーブルがインストール・テープに入っています。

- 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)

デフォルトの予約語テーブルについては、「*Enterprise COBOL 言語解説書*」の付録に説明があります。

CICS 予約語テーブル (IGYCCICS)

Enterprise COBOL 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を提供します。このテーブルにより、CICS のもとでサポートされない COBOL 語はコンパイラーによってエラー・メッセージで示されます。

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

SORT ユーザー

CICS のもとで SORT ステートメントを使用したい場合は (Enterprise COBOL は CICS のもとでの 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ます)、それを使用する前に CICS 予約語テーブルを変更する必要があります。上記の下線が引かれた語 (これらは、SORT 機能のために必要です) を、制限付きとしてマークされた語のリストから削除してください。

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するデータは IGY.V3R4M0.SIGYSAMP のメンバー IGY8CICS に入っています。

注: IGY.V3R4M0 の高位修飾子は、Enterprise COBOL インストール時に変更されている場合があります。

第 2 章 Enterprise COBOL コンパイラー・オプション

この章では、ユーザーがデフォルト値を変更することの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のに役立ちます。コンパイラー・オプションの使用法について詳しくは、「*Enterprise COBOL for z/OS プログラミング・ガイド*」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、インストール先のアプリケーション・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。

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文字でない場合は、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。表 3 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を解決するのに役立ちます。

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 LIB=NO
DBCS=NO	NSYMBOL=NATIONAL
DBCSXREF=(NO 以外)	XREFOPT=NO
DLL=NO	EXPORTALL=YES
DLL=YES	DYNAM=YES RENT=NO
DYNAM=YES	CICS=YES DLL=YES 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
EXPORTALL=YES	DLL=NO DYNAM=YES RENT=NO
LIB=NO	CICS=YES SQL=YES
LIST=YES	OFFSET=YES
MDECK=YES	LIB=NO
NSYMBOL=NATIONAL	DBCS=NO
OBJECT=NO	TEST=(NO 以外)
OFFSET=YES	LIST=YES
OPT=STD または OPT=FULL	TEST=(フック位置の NONE 以外)
RENT=NO	CICS=YES DLL=YES EXPORTALL=YES THREAD=YES
SQL=YES	LIB=NO
TEST=(NO 以外)	OBJECT=NO
TEST=(フック位置サブオプションの NONE 以外)	OPT=STD または OPT=FULL
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

COBOL 85 標準に準拠するようにコンパイラー・オプションを指定する方法については、「Enterprise COBOL for z/OS プログラミング・ガイド」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、それぞれの変更可能なコンパイラー・オプションを説明しています。各構文図の下のテキストは、特定のパラメーターを選択したときの結果を説明しています。

注:

1. DUMP オプションはこのリストに含まれていません。コンパイル時に変更しない限り、DUMP は必ず 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。
2. 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、このオプションを固定すること、およびアプリケーション・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いことを指定することになります。

ADATA

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

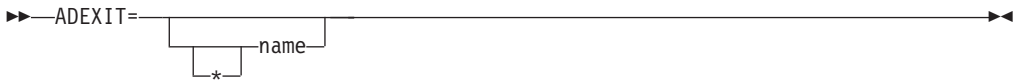
関連データ・ファイルを作成しません。

注:

1. ADATA オプションは、呼び出し時に JCL の PARM フィールド上のオプション・リストを介して指定するか、コマンド・オプションとして指定するか、またはインストール・システム・デフォルトとしてのみ指定することができます。
2. 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。
3. 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。
4. INEXIT を指定すると、関連データ・ファイル用のコンパイル・ソース・ファイルが識別できなくなります。

ADEXIT

構文



デフォルト

出口は指定されません。

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を指定すると、コンパイラーはそのモジュールをロードし、関連データ・ファイルに書き込まれるレコードごとにそれを呼び出します。詳しくは、「Enterprise COBOL for z/OS プログラミング・ガイド」を参照してください。

ADV

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:

1. 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。
2. 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。
3. 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。
4. ファイル記述 (FD) に LINAGE 文節が指定されると、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。

ALOWCBL

構文

デフォルト

ALOWCBL=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:

1. ALOWCBL 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。ALOWCBL 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。
2. 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、ALOWCBL=NO を指定してください。

ARITH

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

構文

デフォルト

AWO=NO

YES

APPLY-WRITE-ONLY 文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 文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 文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 文節をアクティブにしません。

BUF

構文

デフォルト

BUF=4K

integer
各コンパイラー作業ファイル・バッファー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: 大容量のバッファーを使用すると、コンパイラーの性能が向上します。

integerK
バッファー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:

1. コンパイラーは、BUF および SIZE の値を使用して、コンパイル時に使用するストレージの量を判別します。バッファーに割り振られる量は、コンパイラーが SIZE オプション用に使用できる主記憶域の量に含まれます。
2. 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

構文

デフォルト
CICS=NO

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。

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。

注:

1. 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。
2. 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

構文

デフォルト

CODEPAGE (1140)

ccsid

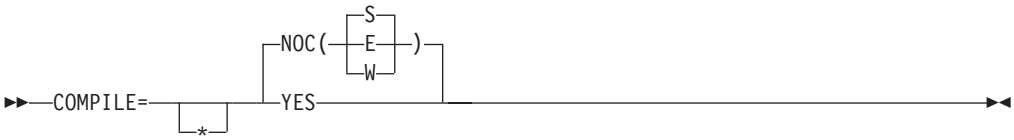
ccsid は、EBCDIC コード・ページを示す有効なコード化文字セット ID (CCSID) 番号である必要があります。

注:

1. CODEPAGE オプションは、以下のエンコードに使用されるコード・ページを指定します。
   - 実行時の英数字および DBCS データ項目の内容
   - COBOL ソース・プログラム内の英数字リテラル、国別リテラル、および DBCS リテラル
2. 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。

COMPILE

構文



デフォルト

COMPILE=NOC(S)

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。

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、コンパイルが停止し、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。

注: 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。

CURRENCY

構文

COBOL のデフォルト通貨記号はドル記号 ($) です。CURRENCY オプションを使用して、別のデフォルト通貨記号を定義することができます。

デフォルト

CURRENCY=NO

literal

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、以下のもの以外の 1 バイトの 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 ～ 9
- 大文字の英字: A B C D P R S V X Z
- 小文字の英字 a ～ 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 文節で通貨記号と見なされるため、PICTURE 文節は 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 文節で通貨記号と見なされるため、PICTURE 文節は 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 文節で通貨記号と見なされるため、PICTURE 文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、リテラル区切り文字のためのオプション設定には関係なく、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は、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、リテラル内の 1 つのアポストロフィを表すのに、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。たとえば、次のようになります。

  ```
  '''' or "'"
  ```

- 16 進数リテラルの指定形式は、次のとおりです。

  ```
  X'H1H2' or X"H1H2"
  ```

  ここで、H1H2 は、上記の通貨記号リテラルに関する規則に準拠した、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。

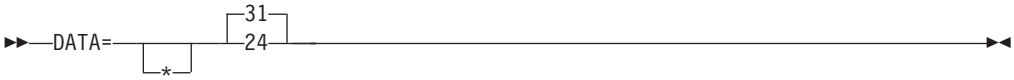
値 NO を指定すると、COBOL ソース・プログラムで CURRENCY SIGN 文節を省く場合と同じ結果になります。

注:

1. COBOL プログラムの PICTURE 文節で使用する通貨記号を選択する場合に、CURRENCY SIGN 文節 (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。
2. プログラムで CURRENCY オプションと CURRENCY SIGN 文節の両方を使用した場合、CURRENCY SIGN 文節に指定した記号を使用すると、その記号が PICTURE 文節内の通貨記号であると見なされます (CURRENCY オプションを固定 {*} した場合でも)。

DATA

構文



デフォルト

DATA=31

31 ユーザー・データ域 (作業用ストレージや 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 16MB 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、16MB 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。

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、16MB 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。

- Enterprise 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。
- Enterprise 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 16MB 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。

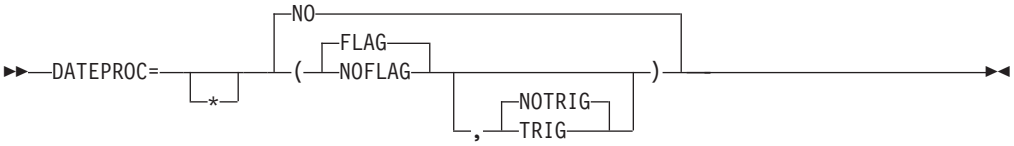
指定しなければ、呼び出し先プログラムでデータをアドレス指定することができません。

注:

1. プログラムを RENT オプションでコンパイルする場合、WORKING-STORAGE およびパラメーター・リスト (PLIST) 用のスペースの獲得方法を DATA オプションが制御します。
2. 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。

DATEPROC

構文



DATEPROC オプションにより、コンパイラーが DATE FORMAT 文節および他の言語構造体を使用して日付処理を行うかどうかが決まります。

デフォルト

DATEPROC=NO、または DATEPROC(FLAG,NOTRIG) (DATEPROC のみが指定されている場合)。

FLAG

DATE FORMAT 文節を認識し、自動日付処理を実行します。さらに、DATEPROC=FLAG を指定すると、日付処理を使用するステートメントまたは日付処理の影響を受けるステートメントに、通知レベル・メッセージまたは警告レベル・メッセージ (適用される方) を出します。

NOFLAG

DATE FORMAT 文節を認識し、自動日付処理を実行します。日付処理を使用するステートメントおよび日付処理の影響を受けるステートメントに通知レベル・メッセージおよび警告レベル・メッセージは出されません。

TRIG

DATE FORMAT 文節を認識し、コンパイラーがウィンドウ化日付フィールドに対する操作に適用する自動ウィンドウ操作に基づいて日付処理を実行します。自動ウィンドウ化日付フィールドは、日付フィールドおよび日付以外の他のフィールドの特定のトリガーまたはしきい値に依存します。これらの特定の値は無効な日付、およびテスト可能であるか上限または下限として使用できる日付を表します。

NOTRIG

DATE FORMAT 文節を認識し、コンパイラーがウィンドウ化日付フィールドに対する操作に適用する自動ウィンドウ操作に基づいて日付処理を実行します。自動ウィンドウ化日付フィールドは、日付フィールドおよび日付以外の他のフィールドの特定のトリガーまたはしきい値に依存しません。日付の年の部分の値のみが自動ウィンドウ操作に関係します。

NO

DATE FORMAT 文節をコメントとして扱い、自動日付処理を使用不可にしま

す。新しい組み込み関数の場合は、DATEPROC=NO を指定すると、新しい組み込み関数が使用されるたびにデフォルト値を戻すオブジェクト・コードが生成されます。

注:

エラー・レベル・メッセージおよび重大レベル・メッセージは、DATEPROC=FLAG または DATEPROC=NOFLAG のいずれを指定したかには関係なく出されます。

DBCS

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:

1. 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。たとえば、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。
2. DBCS=NO を指定すると、NSYMBOL(NATIONAL) と矛盾します。

DBCSXREF

構文

デフォルト

DBCSXREF=NO

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。

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) を共用システム域 (MLPA など) にロードすることを指定します。

xx DBCS 相互参照を行うための適切な配列プログラムのロード・モジュールを指定します。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。

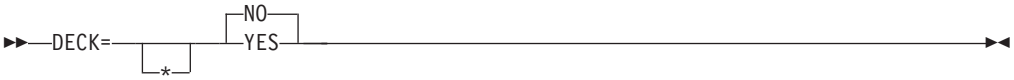
zz 指定した配列タイプが使用するエンコード・テーブルを指定します。8 文字の長さにする必要があります。このパラメーターを省略すると、特定の配列タイプに関連したデフォルトのエンコード・テーブルが使用されます。

注:

1. 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。
2. R を指定し、SIZE 値が MAX 以外である場合は、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のに十分な大きさになるようにしてください。
3. 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戻されます。
4. 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
   - コン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

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

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:

1. 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、診断が出されます (送り出しフィールドが参照変更される場合を除きます)。
2. 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、診断は出されません。

DLL

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 (たとえば、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。

DYNAM

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。しかし、CALL literal ステートメントのある個々のアプリケーションは、パスの長さが長くなるので、パフォーマンスが多少は低下します。

NO
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・ファイルを、呼び出し側プログラムと一緒に単一のモジュール・ファイルに入れます。

注:

1. 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。
2. 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。

EXPORTALL

構文

デフォルト
EXPORTALL=NO

YES
DLL を形成するためにオブジェクト・デックをリンク・エディットするときに、特定の記号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

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。(この制約事項の詳細記述は、「*Enterprise COBOL for z/OS プログラミング・ガイド*」を参照してください。)

NO

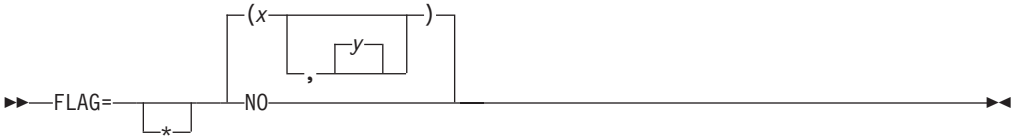
Enterprise COBOL がソートおよびマージについて入出力を実行することを指定します。

注:

1. 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
2. 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

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 I|W|E|S|U

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・リストの終わり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

ID	タイプ	戻りコード
U	回復不能エラー	16

y I|W|E|S|U

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

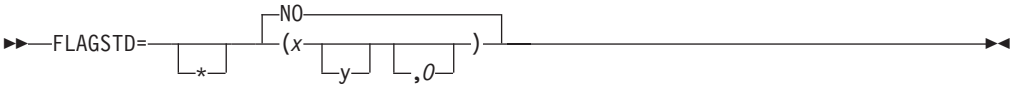
エラー・メッセージにフラグを立てないことを示します。

注:

1. 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。
2. FLAG(W|E|S) を指定すると、関連データ・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 for z/OS プログラミング・ガイド*」を参照してください。

FLAGSTD

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 中間サブセット。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ます。

H = ANS 高サブセット。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ます。

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)。

O 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:

1. 以下のエレメントは、COBOL 85 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 (DATEPROC コンパイラー・オプションによって使用可能にされる) で使用される言語構文 (DATE FORMAT 文節など)
   - 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
2. 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、以下のことを示します。
   - 言語エレメントが廃止であるか、非準拠の標準であるか、または非準拠の非標準であるか (廃止かつ非準拠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)
   - 非準拠または廃止である構文を含んでいる文節、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
3. 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。
4. 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、コンパイル時に警告メッセージが出されます。

ADV=NO	LITCHAR=APOST
DATEPROC=(NO 以外)	NUM=YES
DBCS=YES	NUMPROC=PFD
DYNAM=NO	SEQ=YES
FASTSRT=YES	TRUNC=OPT または BIN
LIB=NO	WORD=(NO 以外、または RWT 以外)
	ZWB=NO

   - 以下のオプションを FLAGSTD=(NO 以外) と共に指定すると、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が戻されます。

ADV=NO	LITCHAR=APOST
DATEPROC=(NO 以外)	NUM=YES
DBCS=YES	NUMPROC=PFD
DYNAM=NO	SEQ=YES
LIB=NO	TRUNC=OPT または BIN
	WORD=(NO 以外、または RWT 以外)
	ZWB=NO

5. 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。

INEXIT

構文

デフォルト

出口は指定されません。

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を指定すると、コンパイラーは SYSIN データ・セットを読み取らずに、その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。詳細については、「*Enterprise COBOL for z/OS プログラミング・ガイド*」を参照してください。

注:

INEXIT を指定すると、コンパイル・ソース・ファイルを識別できなくなります。

INTDATE

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戻します。

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、言語環境プログラムの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戻します。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:

1. 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。

2. 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

構文

デフォルト

LANGUAGE=EN

XX

コンパイラー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 4. LANGUAGE コンパイラー・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:

1. 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 文字に続く文字を使用してもかまいませんが、言語名の判別に使用されるのは最初の 2 文字だけです。
2. このコンパイラー・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。
3. 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。
4. 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。
5. 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。
6. 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。

7. EVENTS オプションまたは 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ことがあります。

LIB

構文

デフォルト

LIB=NO

YES

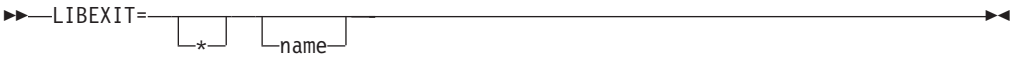
ソース・プログラムに COPY または BASIS ステートメントが入っていることを示します。

NO

ソース・プログラムに COPY または BASIS ステートメントが入っていないことを示します。

LIBEXIT

構文



デフォルト

出口は指定されません。

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を指定すると、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、その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。詳細については、「*Enterprise COBOL for z/OS プログラミング・ガイド*」を参照してください。

LINECNT

構文

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。たとえば、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。

注: LINECNT インストール・オプションは、LINECOUNT コンパイル時オプションと同等です。

LIST

構文

デフォルト
LIST=NO

YES
以下のもの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
- 作業用ストレージに関する情報
- グローバル・テーブル
- リテラル・プール

NO
このリストを抑止します。

注: 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

構文

デフォルト
LITCHAR=QUOTE

QUOTE
1 つまたは複数の引用符 (") 文字を表すために表意定数 [ALL] QUOTE または [ALL] QUOTES が必要な場合は、QUOTE を使用します。QUOTE は COBOL 85 標準に準拠しています。

APOST
1 つまたは複数のアポストロフィ (') 文字を表すために表意定数 [ALL] QUOTE または [ALL] QUOTES が必要な場合は、APOST を使用します。

注:

1. APOST または QUOTE オプションのいずれが有効であるかには関係なく、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。
2. 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。

LVLINFO

構文

デフォルト

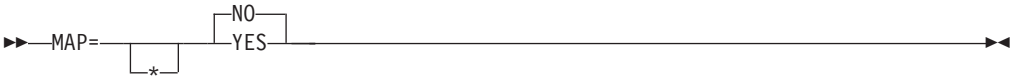
文字は指定されません。

xxxx

リリース番号に続くリスト・ヘッダーに挿入される 1 ～ 4 文字の英数字 (シグニチャー域の最後の 4 バイト) を指定します。このオプションは、リスト・ヘッダー内の「コンパイラー・レベル」情報を識別するために使用することができます。

MAP

構文



デフォルト

MAP=NO

YES

DATA DIVISION で宣言された項目をマップします。マップ出力には、以下のものが含まれます。

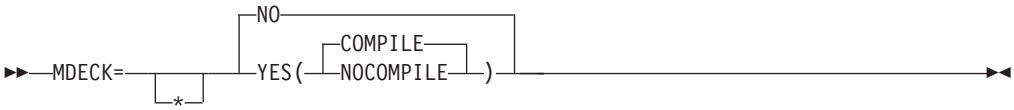
- DATA DIVISION のマップ
- グローバル・テーブル
- リテラル・プール
- プログラム統計
- プログラムの作業用ストレージのサイズ、およびオブジェクト・コードにおけるその位置 (RENT コンパイラー・オプションを指定せずにプログラムをコンパイルする場合)

NO

マッピングを実行しません。

MDECK

構文



デフォルト

MDECK=NO

COMPILE

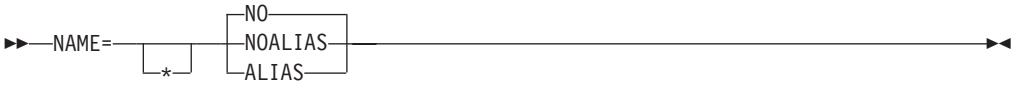
ライブラリーが処理され、MDECK 出力ファイルが生成された後、コンパイルは正常に継続します。

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。

NAME

構文



デフォルト

NAME=NO

NOALIAS

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、リンケージ・エディターの 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 ステートメントから、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。

ALIAS

PROGRAM-ID に対応する NAME ステートメントの前に、プログラム内の ENTRY ステートメントごとにリンケージ・エディターの ALIAS ステートメントを置きます。

NO

リンケージ・エディターの NAME ステートメントを付加しません。

注:

1. 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ます。これは、動的呼び出しのために有用です。

NSYMBOL

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 文節で定義し、USAGE 文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 文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 文節で定義し、USAGE 文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 文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:

1. 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) オプションでは、N 記号を 200x COBOL 標準に従って処理します。
2. NSYMBOL(NATIONAL) を指定すると、DBCS オプションが強制的に実行されます。

NUM

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ー・マップに、コンパイラー生成の行番号ではなく、ソース・プログラムの行番号を使用します。

NO

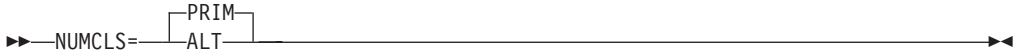
エラー・メッセージおよびプロシージャー・マップに、コンパイラー生成の行番号を使用します。

注:

1. 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

構文



デフォルト

NUMCLS=PRIM

ALT

次のように定義されているデータ項目についての数値クラス・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 文節で「S」によって指定)
- DISPLAY または COMPUTATIONAL-3 (パック 10 進) を使用する
- SIGN 文節に SEPARATE 句を指定しない

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:

1. NUMPROC と NUMCLS の両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。
2. NUMCLS オプションは、NUMPROC=MIG または 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

構文

デフォルト

NUMPROC=NOPFD

MIG

OS/VS COBOL アプリケーション・プログラムを Enterprise COBOL にマイグレーションする際に役立ちます。MIG を指定した処理は、次のようになります。

- 比較および算術演算に、既存の符号を使用します。
- MOVE および算術演算の結果に対して、望ましい符号を生成します。(これらの結果は NUMPROC=PFD の使用基準に一致します。)
- 論理比較ではなく数値比較を実行します。

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 OPT=STD または OPT=FULL の場合に、生成コードを最適化します。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 '-' にする必要があります。

Enterprise COBOL における基本 MOVE 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 および再定義を行うと、非準拠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=PFD 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=PFD 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=MIG および NUMPROC=NOPFD 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。

注:

1. 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=MIG または NUMPROC=NOPFD 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=PFD 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

構文

デフォルト
OBJECT=YES

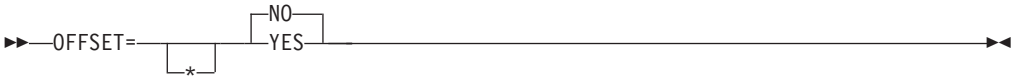
YES
生成されたオブジェクト・コードを、SYSLIN で定義されたファイルに出力します。

NO
オブジェクト・コードを SYSLIN に出力しません。

注: 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す。

OFFSET

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ー部分には、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。さらに、以下のものが生成されます。

- グローバル・テーブル
- リテラル・プール
- プログラム統計
- プログラムの作業用ストレージのサイズ、およびオブジェクト・コードにおけるその位置 (NORENT 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)

NO
リストを圧縮せず、上記のものを生成しません。

注: 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が戻されます。矛盾の解決に関する詳細は、13 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

構文

デフォルト
OPT=NO

NO
コードが最適化されないことを指定します。

STD
最適化オブジェクト・コードを生成します。

パフォーマンスの考慮: OPT=STD または OPT=FULL を使用すると、通常は、より効率的な実行時コードが生成されます。

FULL
未使用のデータ項目をすべて廃棄し、これらのデータ項目の VALUE 文節すべてに対してコードを生成しません。OPT(FULL) と MAP の両オプションを指定すると、MAP 出力において、廃棄されたデータ項目に XXXX という BL 番号が付けられます。これは、その番号が使用されないことを示しています。

注:

1. OPTIMIZE コンパイラー・オプションは、現在 Java インターオペラビリティーのためのオブジェクト指向構文を含むプログラムで完全にサポートされています。
2. デバッグ・ツールを使用して最適化されたオブジェクト・コードのデバッグを行う場合、TEST オプションに指定できるフック位置サブパラメーターは NONE だけです。その他の組み合わせを指定すると、インストー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。
3. S レベル以上のエラーが発生すると、最適化はオフになります。

OUTDD

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。

注:

1. OUTDD が MSGFILE と相互作用する方法については、「*言語環境プログラム プログラミング・リファレンス*」の MSGFILE ランタイム・オプションの説明を参照してください。
2. 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。

PGMNAME

構文

デフォルト
PGMNAME=COMPAT

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。

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。

注:

1. 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
   - ネストされたプログラムへの呼び出しおよび別個にコンパイルされたプログラムへの静的呼び出しにおけるプログラム名参照
   - 静的 SET procedure-pointer TO ENTRY literal ステートメントにおけるプログラム名参照
   - ネストされたプログラムの CANCEL におけるプログラム名参照
2. クラス名およびメソッド名は、PGMNAME オプションの影響を受けません。
3. 詳細については、「*Enterprise COBOL for z/OS プログラミング・ガイド*」を参照してください。

PRTEXIT

構文

デフォルト
出口は指定されません。

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を指定すると、コンパイラーは SYSPRINT データ・セットに書き込む代わりに、そのモジュールをロードして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。詳細については、「*Enterprise COBOL for z/OS プログラミング・ガイド*」を参照してください。

RENT

構文

デフォルト

RENT=YES

YES

COBOL プログラム用に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定します。RENT=YES を使用すると、プログラムを 16MB 境界より上で実行するために、共用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

COBOL プログラム用に生成されるオブジェクト・コードを再入不可にすることを指定します。

注:

1. プログラムを 16MB 境界より上の仮想記憶アドレスで実行する場合は、RENT=YES または RMODE(ANY) を指定してコンパイルする必要があります。
2. CICS 上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。
3. Enterprise COBOL でコンパイルしたプログラムには、必ず AMODE(ANY) があります。プログラムに割り当てられる RMODE は、RENT/NORENT および RMODE コンパイラー・オプションによって異なります。有効な組み合わせには、次のものがあります。

表 5. RENT および RMODE が常駐モードに与える影響

RENT/NORENT 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	RMODE ANY
NORENT	24	RMODE 24

4. THREAD コンパイラー・オプションを指定するときは、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。

RMODE

構文

デフォルト

RMODE=AUTO

AUTO

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

24 NORENT または RENT のいずれ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

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

注:

1. AMODE 24 で実行中のプログラムにデータを渡す必要のある Enterprise COBOL NORENT プログラムの場合、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 16MB 境界の上または下のいずれに置かれるかは、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す。
2. Enterprise COBOL でコンパイルしたプログラムは、必ず AMODE ANY となります。プログラムに割り当てられる RMODE は、RMODE および RENT/NORENT コンパイラー・オプションによって異なります。有効な組み合わせには、次のものがあります。

表 6. RMODE および RENT/NORENT が常駐モードに与える影響

RMODE 設定	RENT/NORENT 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	RMODE ANY
24	RENT	RMODE 24
24	NORENT	RMODE 24

SEQ

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。

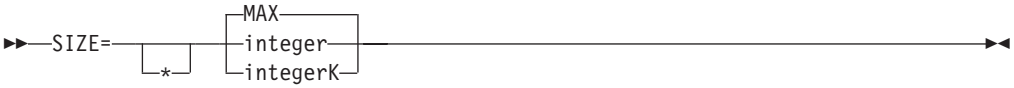
NO
順序検査を行いません。

注:

1. 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。

SIZE

構文



デフォルト
SIZE=MAX

integer
使用できる仮想記憶域の量をバイト単位で指定します。最小許容値は 778240 です。

integerK
使用できる仮想記憶域の量を 1024 バイト (1K) 単位で指定します。

最小許容値は 760K です。

MAX
コンパイル時に、ユーザー領域内のすべての使用可能スペースを要求します。拡張アーキテクチャーの場合、コンパイラーは、16MB 境界より上にある最大の連続するブロックのフリー・ストレージを取得します。

注:

1. SIZE=MAX を使用すると、SIZE オプションの特定の値を決定する必要がないため、コンパイラー呼び出しが簡単になります。
2. コンパイラーを呼び出すときに、ユーザー領域内で特定量の未使用ストレージを使用可能にしておくことが必要な場合には、SIZE=MAX を使用しないでください。
3. 拡張アーキテクチャーの場合に SIZE=MAX を使用すると、コンパイラーは、ユーザー領域内の 16MB 境界より上の使用可能なすべてのストレージ、および 16MB 境界より下になければならない作業ファイル・バッファーとコンパイラー・モジュールのための 16MB 境界より下のストレージを取得します。この割

り振りは、拡張アーキテクチャー環境用に調整が行われない限り適用されます。このため、インストール時にこのオプションを SIZE=MAX に固定しないほうがよい場合もあります。

SOURCE

構文

デフォルト

SOURCE=YES

YES

コンパイラー生成の出力に、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。

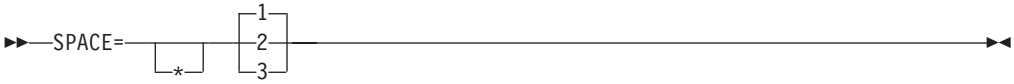
NO

出力にソース・ステートメントを入れません。

注: 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 コンパイラー・オプションを有効にしておく必要があります。

SPACE

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

構文

デフォルト

SQL=NO

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 コンパイラーを呼び出す前に別の SQL プリプロセッサーを使用して SQL ステートメントを処理する場合に使用してください。

YES
DB2 コプロセッサー機能を使用可能にし、DB2 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 SQL ステートメントが含まれており、そのプログラムが DB2 プリコンパイラー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。

注:

1. SQL 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、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・システム・デフォルトのいずれかに指定することができます。
2. DB2 サブ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。
3. DB2 サブオプションを SQL オプションのカスタマイズの一部として指定することはできません。(DB2 サブオプションがサポートされるのは、SQL コンパイラー・オプションが呼び出しオプションとして指定されるか、または CBL または PROCESS カードに指定された場合だけです。) 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。
4. SQL=YES オプションは LIB=NO オプションと矛盾します。

SSRANGE

構文

デフォルト
SSRANGE=NO

YES
ソース・プログラム内の添え字、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 (それらが、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) コードを生成します。生成されたコードは、関数の引き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も検査します。

生成されたコードはさらに、OCCURS DEPENDING ON オブジェクトの設定が誤りである結果として可変長項目が定義済み最大長を超えていないかどうかも検査します。

パフォーマンスの考慮: コンパイル時に SSRANGE=YES を指定すると、オブジェクト・コードのサイズが増大し、また、範囲検査を行うための実行時の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:

1. コンパイル時に SSRANGE オプションが有効であれば、範囲検査を行うためのコードが生成されます。
2. 言語環境プログラムのランタイム・オプション CHECK(OFF) を指定すれば、実行時に範囲検査を使用禁止にすることができます。しかし、その場合でも、範囲検査コードはオーバーヘッドを要し、オブジェクト・コード内で休止状態になっています。
3. 範囲検査コードは、その後、任意で、予期しないエラーの解決のために、再コンパイルせずに使用することができます。

TERM

構文

デフォルト
TERM=NO

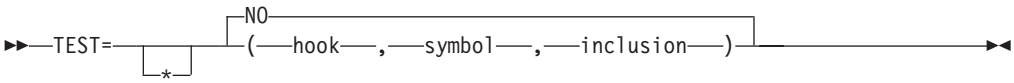
YES
進行メッセージと診断メッセージを SYSTERM ファイル (特に指定しない限り、デフォルトとしてユーザーの端末が使用される) に送ることを指定します。

NO
メッセージを SYSTERM ファイルに送らないことを指定します。

注: 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 SYSTERM DD ステートメントを指定する必要があります。

TEST

構文



デフォルト
TEST=NO

NO 以外
デバッグ・ツールを使用して実行できるオブジェクト・コードを生成します。

フック (hook)、記号 (symbol)、および組み込み (inclusion) サブオプションに値を指定する必要があります。

hook 値:

ALL すべてのコンパイル済みフックの生成をアクティブにします。すべてのステートメント、すべてのパス点、およびすべてのプログラム入り口点と出口点において、フックが生成されます。さらに、DATEPROC=FLAG オプションまたは DATEPROC=NOFLAG オプションが有効であれば、すべての日付処理ステートメントにおいてフックが生成されます。

NONE すべてのコンパイル済みフックの生成を抑止します。TEST(NONE) は、OPT コンパイラー・オプションと互換性があります。

STMT すべてのステートメントとラベル、およびすべてのプログラム入り口点と出口点において、フックが生成されます。さらに、DATEPROC=FLAG オプションまたは DATEPROC=NOFLAG オプションが有効であれば、すべての日付処理ステートメントにおいてフックが生成されます。

PATH すべてのパス点 (プログラム入り口点と出口点を含む) において、フックが生成されます。

BLOCK
すべてのプログラム入り口点と出口点において、フックが生成されます。

symbol 値:

SYM 記号辞書情報テーブルの生成をアクティブにします。

NOSYM
記号辞書情報テーブルの生成を非アクティブにします。

inclusion 値:

SEPARATE
オブジェクト・プログラムとは別個のデータ・セットに、デバッグ情報テーブルを生成します。SEPARATE を指定できるのは、symbol サブオプションの値として SYM を指定した場合だけです。

NOSEPARATE
デバッグ情報テーブルをユーザーのオブジェクト・プログラムに組み込みます。

パフォーマンスの考慮: TEST=(NONE 以外のフック位置サブオプション) は追加のコードを生成するので、実稼働環境で使用すると、実行時にかなりの性能低下を引き起こす可能性があります。

NO
デバッグ・ツールを使用したシンボリック・デバッグができないオブジェクト・コードを生成します。

注:

1. TEST=(NONE 以外のフック位置サブオプション) を指定すると、コンパイル時に以下のオプションが有効になります。
   OBJ=YES
   OPT=NO

2. OPT コンパイラー・オプションと互換性があるのは、フック位置 NONE だけです。矛盾の解決に関する詳細は、13 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。
3. NONE 以外のフック位置サブオプションを指定してコンパイルしたプログラムの場合は、実行時間が増加する場合があります。
4. 日付処理ステートメントとは、日付フィールドを参照するステートメント、または日付フィールドを参照する EVALUATE または SEARCH ステートメント WHEN 句のことです。
5. 実動プログラムの場合、実動モジュールのサイズを増大させたくなければ、TEST(NONE,SYM,SEPARATE) を指定してプログラムをコンパイルしてください。それでも、最小限のデバッグ機能は得られます。

THREAD

構文

デフォルト

THREAD=NO

NO

複数の POSIX スレッドまたは PL/I タスクを含む言語環境プログラムのエンクレーブで COBOL プログラムを実行不可能にするよう指示するには、NO を使用します。

YES

複数の POSIX スレッドまたは PL/I タスクを含む言語環境プログラムのエンクレーブで COBOL プログラムを実行可能にするよう指示するには、THREAD を使用します。

注:

1. THREAD コンパイラー・オプションを指定すると、プログラムをスレッド化アプリケーションで使用できるようになります。ただし、スレッド化されていないアプリケーションで使用することもできます。たとえば、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 オプションでコンパイルされたプログラムを CICS 環境で実行できます。
2. THREAD コンパイラー・オプションを指定すると、直列化論理が自動的に生成されることによって実行時のパフォーマンスが低下する可能性があります。
3. THREAD コンパイラー・オプションを指定するときは、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。
4. COBOL プログラムをスレッド化アプリケーションで実行するには、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。

5. THREAD コンパイラー・オプションを指定する場合、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。
   - ネストされたプログラム
   - PROGRAM-ID 文節の INITIAL 句
   - ALTER ステートメント
   - DEBUG-ITEM 特殊レジスター
   - プロシージャー名を含まない GO TO ステートメント
   - RERUN
   - STOP リテラル・ステートメント
   - 分割モジュール
   - USE FOR DEBUGGING ステートメント
   - WITH DEBUGGING MODE 文節
   - SORT または MERGE ステートメント
6. 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ときは、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

TRUNC

構文

デフォルト

TRUNC=STD

STD

COBOL 85 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 文節内の桁数に切り捨てられます。

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 文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。切り捨ての結果は、生成されるコードの順序によって異なり、また、OS/VS COBOL と Enterprise COBOL とで必ずしも同じであるとは限りません。

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。以下のことを指定します。

1. 出力 2 進数フィールドは、COBOL の 10 を基数とするピクチャー境界ではなく、S/390 のハーフワード、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。
2. 入力 2 進数フィールドは、S/390 のハーフワード、フルワード、ダブルワードとして扱われます。値は、基数 10 の PICTURE 文節によって暗黙指定される値に限定されるとはみなされません。
3. 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。

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。

注:

1. 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。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。
2. S/390 形式の 2 進データを持つその他の製品 (CICS、DB2、FORTRAN、および PL/I など) に接続する場合、TRUNC=BIN の使用をお勧めします。これは、フルワードで 10 桁以上、ハーフワードで 5 桁以上のデータを扱う可能性があるときに、特に当てはまります。

VBREF

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。

WORD

構文

デフォルト

WORD=*NO

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。

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 ～ 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。

ASM1	LIBO	LVL8	RDSC
ASM2	LIBR	OPTM	RWT
DIAG	LSTR	OSCN	SAW
DMAP	LVL0	PGEN	SCAN
DOPT	LVL1	RCTL	SIMD
FGEN	LVL2	RDPR	XREF
INIT	LVL3		

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、11 ページの『CICS 予約語テーブル (IGYCCICS)』を参照してください。

注:

1. WORD オプションのデフォルトは、アスタリスク付きで指定されます。このオプションを、デフォルト (WORD=*NO) を用いてインストールした場合、アプリ

ケーション・プログラマーは、コンパイル時に、代替予約語テーブルを指定するためにこのオプションをオーバーライドすることはできません。

2. 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 42 個の組み込み関数名は、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。
3. WORD=XXXX オプションのデフォルト値を変更すると、FLAGSTG に指定された NO 以外のすべての値と矛盾します。

XREFOPT

構文

デフォルト

XREFOPT=FULL

SHORT

明示的に参照された変数のみを含む相互参照リストを生成します。

FULL

ソート / 組み込み済みの相互参照リストを生成します。SOURCE=YES も指定すると、リストの行番号は、特定のデータ名の回帰参照を相互参照します。

NO

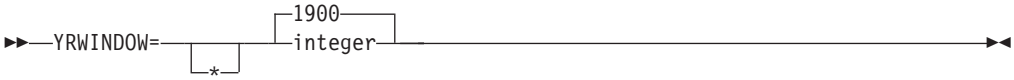
相互参照リストを抑止します。

注:

1. XREFOPT オプションは、コンパイラー・オプション XREF のデフォルト値を設定します。
2. XREFOPT=NO オプションは、DBCSXREF に指定された NO 以外の値と矛盾します。

YRWINDOW

構文



デフォルト

YRWINDOW=1900

integer

COBOL ウィンドウ化日付フィールドで使用される 100 年ウィンドウの最初の年を指定します。次のいずれかです。

- 符号なし 10 進整数 1900 ～ 1999。

- 負の 10 進整数 -1 ～ -99。これは、実行時の現行年からのオフセットを表します。現行年は、それぞれのコンパイル単位ごとに、それが最初に初期化されるときに決定されるか、またはそのコンパイル単位を参照する CANCEL ステートメントの実行後に再初期化されるときに決定されます。

注:

1. DATEPROC=FLAG または DATEPROC=NOFLAG を指定しない限り、YRWINDOW は有効ではありません。
2. 実行時に、以下の 2 つの条件が成立している必要があります。
   a. 100 年ウィンドウの開始年が 1900 ～ 1999 の範囲内である。
   b. 現行年が、そのコンパイル単位の 100 年ウィンドウ内にある。
3. ウィンドウ化日付はすべて、基本年に対する相対的な年を持ちます。たとえば、基本年を 1965 として指定すると、ウィンドウ化年の値はすべて、1965 ～ 2064 という 100 年ウィンドウの中の年として解釈されます。したがって、ウィンドウ化年の値 67 は 1967 年を表し、ウィンドウ化年の値 05 は 2005 年を表します。

   基本年を -30 として指定すると、100 年ウィンドウは、アプリケーションがいつ実行されたかによって異なります。アプリケーションを 2000 年に実行すると、1970 ～ 2069 という 100 年ウィンドウになります。したがって、ウィンドウ化年の値 70 は 1970 年を表し、ウィンドウ化年の値 69 は 2069 年を表します。
4. YRWINDOW インストール・オプションは、YEARWINDOW コンパイル時オプションと同等です。

ZWB

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:

1. 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。

2. 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

Enterprise COBOL のインストールが完了した後、この製品に変更を加えることができます。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受け入れなかった場合は、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。そうすれば、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。

注: Enterprise COBOL をインストールするのに必要なすべての情報は、製品と一緒に提供されるプログラム・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.V3R4M0.SIGYSAMP に置かれます。 58 ページの表 7 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、以下のセクションで詳しく説明します。

IBM が提供する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。

IGY.V3R4M0.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 つにコピーしてください。このようにして、変更をやり直したいときのために、未変更のバックアップ・コピーを用意しておきます。

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。変更した JCL は、今後、IGYWMLPA のために参照できるように保管してください。

表 7. Enterprise COBOL 用のユーザー変更ジョブの要約

説明	カスタマイズ・ジョブ	ページ
コンパイラー・モジュールを変更する	IGYWDOPT	59
固定コンパイラー・オプションをオーバーライドするためのオプション・モジュールを作成する	IGYWUOPT	59
追加の予約語テーブルを作成する	IGYWRWD	60
Enterprise COBOL のモジュールを共用ストレージに置く	IGYWMLPA	66

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を変更したり、固定オプションをオーバーライドするためにコンパイラー・オプション・モジュールを作成するには、オプション・モジュール IGYCDOPT のソースを IGY.V3R4M0.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。次いで、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・オプションに一致させます。変更する 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。

- 継続文字 (ソース中の X) は、IGYCOPT 呼び出しの各行 (最終行以外) の桁 72 に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。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があってはなりません。
- 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。たとえば、LVLINFO=(Fix1) と LVLINFO=(FIX1) はどちらも有効です。
- ストリング・サブオプションの 1 つに特殊文字 (たとえば、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 (’) または単一アンパーサンド (&) を使用するには、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に、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したい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 IBM 提供のデフォルトを提供します。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、3 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。コンパイラー・オプションの説明については、13 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。

コンパイラー・オプション・デフォルト・モジュールを変更する

Enterprise COBOL 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 を使用します。13 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て、デフォルト値を選択してください。

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用ストレージに置いてください。詳しくは、 7 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 66 ページの『Enterprise COBOL のモジュールを共用ストレージに置く方法』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. 必要に応じて、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.V3R4M0.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 が戻されます。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。

固定オプションをオーバーライドするためのオプション・モジュールを作成する

コンパイラー・デフォルト・オプション・モジュール内で一部のオプションを固定として指定した場合には、固定オプションをオーバーライドする必要のあるアプリケーションが見つかることがあります。その他のオプションを指定するには、そのアプリケーションのコンパイル時に STEPLIB または JOBLIB (V3R4M0.SIGYCOMP データ・セットの前) としてアクセスされる別個のデータ・セットに、オプション・モジュールの一時コピーを作成します。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。アセンブリーおよびリンク・エディットは SMP/E 制御の外側で行われるので、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. 必要に応じて、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.V3R4M0.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 提供の接頭部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 ステートメントの DSNAME=YOURLIB を、IGYCDOPT モジュールのリンク先となる区分データ・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 を戻します。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。

追加予約語テーブルの作成または変更

Enterprise COBOL コンパイラー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) は、代替予約語テーブルとして提供されます。11 ページの『CICS 予約語テーブル (IGYCCICS)』を参照してください。予約語テーブル・ユーティリティー (IGY8RWTU) を使用して予約語を変更できるので、追加の予約語テーブルを作成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・ユーティリティーは、予約語テーブルの作成または変更のためにユーザーが使用できる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。

- 既存の予約語の代替形式を追加する。
- 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。
- 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 ～ 4 文字の ID が必要です。使用できない 1 ～ 4 文字の文字ストリングのリストについては、コンパイラー・オプション「WORD=xxxx」を参照してください。

コンパイル時に、コンパイラー・オプション **WORD(xxxx)** の値により、使用する予約語テーブルを識別します。**xxxx** は、メンバー名 IGYCxxxx にユーザーが指定した固有の 1 ～ 4 文字の 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: 予約語テーブル内の項目の合計数は、1536 (1.5 KB) 以内にしてください。

予約語テーブルを作成または変更する

予約語テーブルを作成または変更するには、以下の適切なソース・ファイルのコピーを編集する必要があります。

- IGY.V3R4M0.SIGYSAMP のメンバー IGY8RWRD (IBM 提供のデフォルト予約語テーブル)
- IGY.V3R4M0.SIGYSAMP 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. 通知メッセージが出される必要のない CODASYL 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 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廃止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をコーディングする』を参照)。
6. JCL を変更し、実行します (65 ページの『新しい予約語テーブルを作成するために JCL を変更し、実行する』を参照)。さらに、新しい予約語テーブルの固有の 1 ～ 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をコーディングする

予約語テーブルを作成するためには、制御ステートメントおよび制御ステートメント内のオペランドに関する構文規則を理解する必要があります。

図 2 は、予約語プロセッサー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. 予約語プロセッサー制御ステートメントの構文形式

図 2 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。

注: 制御ステートメント・キーワード INFO および RSTR で識別した語はすべて、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、メッセージで示されます。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ときは、次の規則に従ってください。

- 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 (:) 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 ～ 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をコーディングする

以下のリストは、制御ステートメントにコーディングするオペランドのタイプを示しています。

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。

comments
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ときは、次の規則に従ってください。

- 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 1 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 1 度だけ指定することができます。

以下のセクションに、各タイプの制御ステートメントのコーディング例が示されています。

ABBR ステートメント

ABBR reserved-word: user-word [comments]
指定し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。

注:

1. 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。
2. reserved-word は、元の定義を持つ予約語のままです。
3. 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。
4. コンパイラー出力 (他のリスト、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 COBOL-word[(0 | 1)] [comments]

このステートメントは、コンパイラーによってフラグが設定される COBOL 語を指定します。

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。1 または 0 を選択して、特定の COBOL-word を 1 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、通知メッセージ 0086 が出されます。

1 指定した COBOL-word は 1 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。

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
```

RSTR ステートメント

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

このステートメントは、プログラムで使用できない COBOL 語を指定します。

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。1 または 0 を選択して、特定の COBOL-word を 1 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 が出されます。

1 指定した COBOL-word は 1 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。

注: 以下の予約語は、1 のオプションを指定してのみ制限することができます。

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

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 COBOL 85 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
```

新しい予約語テーブルを作成するために JCL を変更し、実行する

新しい予約語テーブルを作成するために使用する JCL には、STEP1、STEP2、および STEP3 が含まれています。それらは、それぞれ次のことを行います。

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。
変更済み予約語テーブルをアセンブルする。
オブジェクト・モジュールからランタイム・ロード・モジュールを作成する。

このジョブを実行すると、新しい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 ～ 4 文字の ID を加えた名前が付けられます。

非 SMP/E JCL を変更し、実行する

新しい予約語テーブルを作成するには、IGY.V3R4M0.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、予約語ユーティリティーへの入力として IGY.V3R4M0.SIGYSAMP のメンバー IGY8---- で配布されており、IGY.V3R4M0.SIGYCOMP でロード・モジュール IGYC---- を作成します。このジョブを実行するには、次のようにしてください。

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ー・ターゲット・データ・セット名に一致させます。
4. 変更済み予約語テーブルを指し示すために、次のいずれか を実行します。
   - //RSWDREAD DD DSN=... 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。

   注: 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 ステートメント上のデータ・セット名を変更し、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 ステートメント上のデータ・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 ステートメント上のデータ・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。

Enterprise COBOL のモジュールを共用ストレージに置く方法

IGY.V3R4M0.SIGYCOMP 内の再入可能なモジュールはすべて、共用ストレージに入れることができます。そのためには、次の手順を実行します。

- データ・セット IGY.V3R4M0.SIGYCOMP を許可します。
- IGY.V3R4M0.SIGYCOMP を LNKLSTnn 連結に入れます (オプション)。
- システムの IPL 時に MLPA に常駐するモジュールをリストするメンバー IEALPAnn を、SYS1.PARMLIB 内に作成します。

IGYWMLPA が IGY.V3R4M0.SIGYSAMP にインストールされていますので、IEALPAnn メンバーを作成するときの例として使用してください。

z/OS では、モジュールを LPA にロードできるようにするために IGY.V3R4M0 を LNKLSTnn 連結に入れる必要はありません。LNKLSTnn 連結に追加しない場合は、次のいずれか の方法で、LPA に置かれていないモジュールを、Enterprise COBOL アプリケーションのコンパイル・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、LNKLSTnn 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。

IGY.V3R4M0.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 に置くという目的を果たさなくなります。

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 LNKLSTnn 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の詳細は、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*、*SA88-8563*
- *z/OS MVS 初期設定およびチューニング 解説書*、*SA88-8564*

コンパイラー・フェーズを共用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ー・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。詳しくは、58 ページの『コンパイラー・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ーの調整

インストール先では、用途に応じてカタログ式プロシージャー IGYWC、IGYWCL、IGYWCLG、IGYWCG、IGYWCPL、IGYWCPLG、IGYWCPG、および IGYWPL を調整しなければならない場合があります。次のような変更について、検討する必要があります。

- Enterprise COBOL または言語環境プログラムのターゲット・データ・セットについて、IBM 提供以外の接頭部を使用するために、データ・セット名の接頭部を変更する。
- LINKLIST 連結に IGY.V3R4M0.SIGCOMP および CEE.SCEERUN を置いた場合に、STEPLIB の DD ステートメントを除去する。
- インストール先では、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多い場合に、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。

第 4 章 COBOL のための Unicode サポートのカスタマイズ

Enterprise COBOL では、Unicode の基本的なランタイム・サポートが、国別データ型、国別リテラル、組み込み関数、およびコンパイラー・オプションにより提供されています。また、Java インターオペラビリティーおよび XML サポートのための COBOL オブジェクト指向構文は、Unicode 機能を暗黙的に使用します。 COBOL ソース・プログラムは、引き続き EBCDIC (SBCS または DBCS) コード・ページでエンコードされます。

COBOL プログラムの以下のタイプのコンパイル、あるいは実行の前に、プログラムをコンパイルする開発システムと、プログラムを実行する実動システムの両方で、Unicode サービスをインストールして、アクティブ化 (初期化) し、構成してください。

- Java との相互運用のためにオブジェクト指向構文を使用する COBOL プログラム
- 国別データ型、国別リテラル、あるいは DISPLAY-OF / NATIONAL-OF 組み込み関数を含む COBOL プログラム
- XML GENERATE ステートメントを使用する COBOL プログラム

必要な変換サポートが利用可能でない場合は、重大度 3 の言語環境プログラム条件がランタイムで発生するか、またはコンパイラーの診断が発行されます。

Unicode サービスのインストール、セットアップ、およびアクティブ化

ご使用のシステムが、IPL 時に Unicode サービスをアクティブにするようにセットアップされていることを確認してください。Unicode サービスをアクティブにするには、SYS1.PARMLIB の IEASYS00 メンバー内で UNI=*xx* をコーディングする必要があります。ここで、*xxx*は変換環境の構成用の CUNUNI*xx* parmlib メンバーを示します。 //publibz.boulder.ibm.com/epubs/pdf/iea2un41.pdf から入手できる「*z/OS Support for Unicode: Using Conversion Services* (SA22-7649)」に記載の指示に従ってください。 (z/OS support for Unicode は、オペレーティング・システムの一部として組み込まれています。)

COBOL のための変換イメージの作成

システムに Unicode サービスをセットアップした後に、Unicode サービスのシステム上での使用をサポートする変換イメージを作成して、アクティブにする必要があります。この節では、COBOL の要件をサポートする変換イメージの作成方法を説明します。

変換イメージを作成するためには、制御ステートメントを使用して、変換イメージ生成プログラムを起動してください。これらの制御ステートメントにより、次のものが組み込まれます。

- 指定のソース CCSID およびターゲット CCSID 用の変換テーブル
- 変換のために使用される変換技法

以下の CCSID の対が、使用する言語機能に応じて COBOL プログラムで必要です。

プログラムの説明	ソース CCSID	ターゲット CCSID
Unicode データを使用するプログラム。	CODEPAGE コンパイラー・オプションで有効にされた CCSID。	CCSID 1200 (UTF-16)
	CCSID 1200	CODEPAGE コンパイラー・オプションで有効にされた CCSID。
明示的に CCSID が指定された NATIONAL-OF 組み込み関数を含むプログラム。	組み込み関数で指定された CCSID。	CCSID 1200
明示的に CCSID が指定された DISPLAY-OF 組み込み関数を含むプログラム。	CCSID 1200	組み込み関数で指定された CCSID。
Java インターオペラビリティーのための、オブジェクト指向構文を含むプログラム。	CODEPAGE コンパイラー・オプションで有効にされた CCSID。	CCSID 1208 (UTF-8)
	CCSID 1200	CCSID 1208
XML GENERATE ステートメントを含むプログラム。	CODEPAGE コンパイラー・オプションで有効にされた 1140 および CCSID。	CCSID 1200
	CCSID 1200	CODEPAGE コンパイラー・オプションで有効にされた 1140 および CCSID。

変換技法は、*技法検索順序* と呼ばれます。 COBOL ではデフォルトの技法検索順序が使用されるので、制御ステートメントでの技法検索順序は省略する必要があります。たとえば、CCSID 1140 用の変換テーブルを COBOL 用の CCSID 1200 変換に構成するには、次の制御ステートメントをコーディングします。

```
CONVERSION 1140,1200;
```

COBOL が使用する技法検索順序の選択は固定されており、アプリケーション・プログラムでは変更できません。

オブジェクト指向構文に必要となる CCSID の対を除くと、CCSID の対は常時、CCSID 1200 と他の CCSID の組み合わせになります。

Java インターオペラビリティーのためのオブジェクト指向構文を含むプログラムでは、以下の CCSID の対が必要です。

- CODEPAGE コンパイラー・オプションで指定された CCSID と、CCSID 1208
- CCSID 1200 と CCSID 1208

CODEPAGE コンパイラー・オプションの IBM 出荷時のデフォルトは 1140 (ユーロ記号を含む EBCDIC Latin-1) です。

例: Java インターオペラビリティーのためのオブジェクト指向構文を使用するプログラム

以下の特性があるアプリケーションの、インストールを検討してみましょう。

- アプリケーションが CODEPAGE コンパイラー・オプションのデフォルト値 (CCSID 1140) のみを使用する。
- DISPLAY-OF あるいは NATIONAL-OF 組み込み関数で、アプリケーションが明示的に CCSID 値を指定しない。
- アプリケーションが Java インターオペラビリティーのためのオブジェクト指向構文を使用する。

このケースでは、少なくとも、以下の変換のセットを使用して Unicode サービスを構成してください。

- CCSID 1140 と 1200
- CCSID 1200 と 1140
- CCSID 1140 と 1208
- CCSID 1200 と 1208

例: Unicode データを使用するプログラム

以下の例では、DISPLAY-OF および NATIONAL-OF 組み込み関数と共に国別データ項目を使用しています (**太字**のテキストはソース・プログラムの一部ではありません)。

```
CBL CODEPAGE(1140) NSYMBOL(NATIONAL)
Identification Division.
Program-ID. Sample1.
. . .
Data Division.
Working-Storage Section.
01 Latin1-data PIC X(10).
01 ASCII-Latin1-data PIC X(10).
01 Japanese-MBCS-data PIC X(20).
01 National-data PIC N(10) Usage National.
. . .
Procedure Division.
    . . .
(1) Move 'ABCDE12345' to Latin1-data
(2) Move Latin1-data to National-data
(3) Move Function DISPLAY-OF(National-data,1252)
      To ASCII-Latin1-data
(4) Move Function DISPLAY-OF(National-data,1399)
      To Japanese-MBCS-data
    . . .
(5) Move Function NATIONAL-OF(Japanese-MBCS-data, 1399)
      To National-data
(6) Display National-data
    . . .
    Goback.
```

このプログラムは以下のような CCSID の対の変換テーブルが必要です。

- CCSID 1140 と CCSID 1200 (CODEPAGE コンパイラー・オプションで有効にされた CCSID と CCSID 1200)
- CCSID 1200 と CCSID 1140 (CCSID 1200 と CODEPAGE コンパイラー・オプションで有効にされた CCSID)

- CCSID 1200 と CCSID 1252 (上記のステートメント 3 にある DISPLAY-OF 関数に必要)
- CCSID 1200 と CCSID 1399 (上記のステートメント 4 にある DISPLAY-OF 関数に必要)
- CCSID 1399 と CCSID 1200 (上記のステートメント 5 にある NATIONAL-OF 関数に必要)

COBOL DB2 プログラムの考慮事項

COBOL プログラムに DB2 SQL ステートメントが含まれる場合、DB2 に必要な変換テーブルと COBOL に必要な変換テーブルの両方を持った Unicode サービスを構成する必要があります。DB2 の場合の主要な相違点は、DB2 では、変換技法検索順序にデフォルトではなく ER が使用されることです。したがって、たとえ COBOL と DB2 で、同じ CCSID の対の間の変換が必要になったとしても、その対に 2 つの変換テーブルを作成する必要があります。1 つは、技法検索順序が省略された COBOL 用変換テーブル、そしてもう 1 つは技法検索順序 ER を使用する変換テーブルです。

推奨: 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、COBOL アプリケーション (CODEPAGE(*nnnnn*) コンパイラー・オプションを使用して指定) 、および 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) と同じコード・ページを使用してください。

Unicode サービス構成のための DB2 要件の追加情報については、www.ibm.com/software/data/db2/zos/library.html にある DB2 ライブラリーを参照してください。 www.ibm.com/software/data/db2/zos/v8books.html にある、以下の資料を特に参照してください。

- *DB2 UDB for z/OS V8 インストレーション・ガイド*
- *DB2 UDB for z/OS Internationalization Guide* (Unicode)

例: 変換イメージの生成のための JCL

以下のバッチ・ジョブの例は、次の 3 つのタイプの COBOL プログラムをサポートします。

- 上記の言語エレメントを含む COBOL プログラム
- Java インターオペラビリティーのための OO 構文を含む COBOL プログラム
- DB2 による CCSID 1200 と CCSID 1140 間の変換を必要とする COBOL DB2 プログラム

このサンプル JCL は、Unicode サービス・パッケージで提供される *hlq*.SCUNJCL(CUNJIUTL) のバリエーションです。

```
//CUNMIUTL EXEC PGM=CUNMIUTL
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SYSIMG   DD   DSN=UNI.IMAGES(CUNIMG01),DISP=SHR
//TABIN    DD   DSN=UNI.SCUNTBL,DISP=SHR
//SYSIN    DD   *
  /*************************************************************/
```

```
  /* Conversion image input for conversions between national      */
  /* data (1200) and alphanumeric, DBCS or MBCS data with         */
  /* CCSID in effect for CODEPAGE compiler option                 */
  /* Default "technique search order" is used by COBOL            */
  /*****************************************************************/
  CONVERSION 1140,1200; /* Latin-1 to UTF-16, RECLM               */
  CONVERSION 1200,1140; /* UTF-16 to Latin-1, RECLM               */

  /*****************************************************************/
  /* Conversion image input for CCSIDs specified                  */
  /* in DISPLAY-OF and NATIONAL-OF intrinsic functions            */
  /* DISPLAY-OF requires conversion from 1200 to specified CCSID  */
  /* NATIONAL-OF requires conversion from specified CCSID to 1200 */
  /* Default "technique search order" is used by COBOL            */
  /*****************************************************************/
  CONVERSION 1200,1252; /* UTF-16 to ASCII Latin-1, RECLM         */
  CONVERSION 1200,1399; /* UTF-16 to Japanese, RECLM              */
  CONVERSION 1399,1200; /* Japanese to UTF-16, RECLM              */

  /*****************************************************************/
  /* Conversion image input for COBOL OO support                  */
  /* Default "technique search order" (RECLM) is used by COBOL    */
  /*****************************************************************/
  CONVERSION 1140,1208; /* Latin-1 to UTF-8, RECLM                */
  CONVERSION 1200,1208; /* UTF-16 to UTF-8, RECLM                 */

  /*****************************************************************/
  /* Conversion image input for DB2                               */
  /* "technique search order" used by DB2 is ER                   */
  /*****************************************************************/
  CONVERSION 1140,1200,ER; /* Latin-1 to UTF-16, ER               */
  CONVERSION 1200,1140,ER; /* UTF-16 to Latin-1, ER               */
/*
```

COBOL は、文字変換サービスは使用しますが、Unicode サービスのケース変換、あるいは正規化サービスは使用しません。他の製品やアプリケーションが必要としない限り、変換イメージ作成のために、CASE 制御ステートメント、あるいは NORMALIZE 制御ステートメントを組み込む必要はありません。

付録. 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あり、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。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

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で提供し、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。

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

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
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すが、有償の場合もあります。

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

プログラミング・インターフェース情報

Enterprise COBOL for z/OS には、Enterprise COBOL for z/OS のサービスを使用するプログラムをお客様が作成する際に使用できるマクロはありません。

重要: Enterprise COBOL for z/OS の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てはなりません。

商標

以下は、IBM Corporation の商標です。

CICS	DFSORT	OS/390
COBOL/370	IBM	S/390
DB2	Language Environment	z/OS

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. 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。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

資料名リスト

Enterprise COBOL for z/OS

移行ガイド、GC88-9118

カスタマイズ・ガイド、GC88-9119

Debug Tool User's Guide、SC18-7171

Debug Tool Reference and Messages、SC18-7172

Fact Sheet、GC27-1407

言語解説書、SC88-9117

Licensed Program Specifications、GC27-1411

プログラミング・ガイド、SC88-9121

z/OS 言語環境プログラム

z/OS 言語環境プログラム 概念、SA88-8555

z/OS 言語環境プログラム デバッグのガイド、GA88-8548

z/OS 言語環境プログラム ランタイム・メッセージ、SA88-8554

z/OS 言語環境プログラム カスタマイズ、SA88-8552

z/OS 言語環境プログラム プログラミング・ガイド、SA88-8549

z/OS 言語環境プログラム プログラミング・リファレンス、SA88-8550

z/OS 言語環境プログラム ランタイム マイグレーション・ガイド、GA88-8553

z/OS 言語環境プログラム ILC (言語間通信) アプリケーションの作成、SA88-8551

関連資料

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* と *ANSI INCITS 23b-1993*、*Programming Languages - Correction Amendment for COBOL*

CICS Transaction Server アプリケーション・プログラミング・ガイド、SC88-7689

CICS Transaction Server アプリケーション・プログラミング・リファレンス、SC88-7690

CICS Transaction Server カスタマイズ・ガイド、SC88-7686

CICS Transaction Server 外部インターフェース・ガイド、SD88-7026

日本語配列プログラム プログラム解説書、N:SH18-0144

DB2 UDB for z/OS インストレーション・ガイド、GC88-9811

DB2 UDB for z/OS Internationalization Guide (Unicode)

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*、と *ISO/IEC 1989/AMD2:1994*、*Programming languages - Correction and clarification amendment for COBOL*

z/OS MVS 初期設定およびチューニング ガイド、SA88-8563

z/OS MVS 初期設定およびチューニング 解説書、SA88-8564

z/OS Support for Unicode: Using Conversion Services、SA22-7649

IBM SMP/E for z/OS User's Guide、SA22-7773

IBM SMP/E for z/OS Reference、SA22-7772

z/OS TSO/E 入門、SA88-8632

z/OS TSO/E ユーザーズ・ガイド、SA88-8638

ISPF ユーザーズ・ガイド 第 1 巻、SC88-8965

ソフトコピー資料

以下のコレクション・キットには、IBM COBOL および関連製品の資料が含まれています。

- *z/OS Collection*、SK3T-4269

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]

[カ行]

[サ行]

[タ行]

[ナ行]

[ハ行]

[マ行]

[ヤ行]

[ワ行]

A

B

C

D

E

F

I

L

M

N

O

P

R

S

T

V

W

X

Y

Z

プログラム番号: 5655-G53

Printed in Japan

GC88-9119-03

日本アイ・ビー・エム株式会社
〒106-8711　東京都港区六本木3-2-12