IBM XL C for AIX, V10.1

はじめに

GC88-4919-00
(英文原典：GC23-8896-00)

IBM XL C for AIX, V10.1

はじめに

GC88-4919-00
(英文原典：GC23-8896-00)

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、 29 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 XL C for AIX, V10.1 (プログラム番号 5724-U80)、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。製品のレベルに合った正しい版を使用していることを確認してください。

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： GC23-8896-00
IBM XL C for AIX, V10.1
Getting Started with XL C
Version 10.1

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 2008.5

© Copyright International Business Machines Corporation 1996, 2008. All rights reserved.

目次

© Copyright IBM Corp. 1996, 2008

本書について

本書には、IBM® XL C for AIX®, V10.1 コンパイラーの概要および基本的な使用法に関する情報が含まれています。

本書の対象読者

本書は、XL C の基本的な概要および使用法に関する情報を必要とする C の開発者向けです。 コマンド行コンパイラー、 C プログラミング言語の基本的な知識、およびオペレーティング・システム・コマンドの基本的な知識に習熟していることを前提としています。 XL C に慣れていないプログラマーは、本書を使用して XL C 固有の能力や機能に関する情報を確認することができます。

本書の読み方

本書の全体を通じて、コンパイラーの動作の説明には **xlc** コンパイラー呼び出しを使用しています。 ただし、特定の環境では必要に応じてコンパイラー呼び出しコマンドの別の形式に置き換えることができます。また、コンパイラー・オプションの使用法は、特に指定がない限り同じです。

本書は、コンパイラー環境の構成、XL C コンパイラーを使用した C アプリケーションのコンパイルおよびリンクに関する情報をカバーしていますが、以下のトピックは含まれていません。

- コンパイラー・インストール: XL C のインストールの詳細については、「*XL C インストール・ガイド*」を参照してください。
- コンパイラー・オプション: コンパイラー・オプションの構文および使用法について詳しくは、「*XL C コンパイラー・リファレンス*」を参照してください。
- C プログラミング言語: C プログラミング言語の構文、セマンティクス、および IBM インプリメンテーションについては、「*XL C ランゲージ・リファレンス*」を参照してください。
- プログラミング・トピック: プログラムの移植性および最適化に焦点を置いた、XL C でのアプリケーションの開発について詳しくは、「*XL C プログラミング・ガイド*」を参照してください。

表記規則

書体の規則

© Copyright IBM Corp. 1996, 2008

次の表では、IBM XL C for AIX, V10.1 の資料で使用される書体の規則について説明します。

表 1. 書体の規則

書体	対象	例
太字	小文字コマンド、実行可能ファイル名、コンパイラー・オプション、およびディレクティブ。	コンパイラーには、基本的な呼び出しコマンドである **xlc** と、さまざまな C 言語レベルとコンパイル環境をサポートするためのその他のコンパイラー呼び出しコマンドが併せて用意されています。
イタリック	実際の名前や値をユーザーが指定するパラメーターまたは変数。新規用語を紹介する際にも、イタリックが使用されます。	要求された *size* より大きな値を返す場合は、必ず *size* パラメーターを更新してください。
下線	コンパイラー・オプションまたはディレクティブのパラメーターのデフォルト設定。	nomaf | maf
モノスペース	プログラミング・キーワードおよびライブラリー関数、コンパイラーの組み込み関数、プログラム・コードの例、コマンド・ストリング、またはユーザー定義名。	myprogram.cをコンパイルして、最適化するには、`xlc myprogram.c -O3` と入力します。

構文図

本書では、構文図は XL C の構文を表します。このセクションでは、これらの図を解釈し、使用するための説明をします。

- 構文図は、線の経路に沿って左から右、上から下に読みます。

  ▸▸── 記号は、コマンド、ディレクティブ、またはステートメントの始まりを示します。

  ──▸ 記号は、コマンド、ディレクティブ、またはステートメントの構文が次の行に続いていることを示します。

  ▸── 記号は、コマンド、ディレクティブ、またはステートメントが前の行からの続きであることを示します。

  ──▸◂ 記号は、コマンド、ディレクティブ、またはステートメントの終わりを示します。

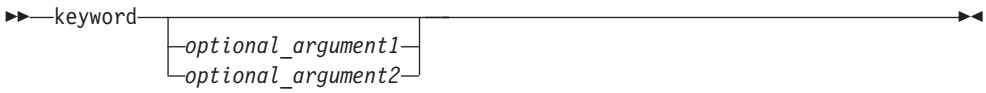
  完全なコマンド、ディレクティブ、またはステートメントではない構文単位の断片図は、 |── 記号で始まり、──| 記号で終わります。
- 必須項目は水平線 (メインパス) 上に示されます。

- オプション項目はメインパスの下側に示されます。

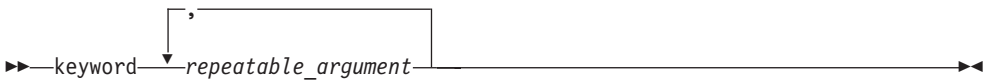
- 複数の項目から選択できる場合、それらの項目は縦に並べて表示されます。

  複数の項目から 1 つを選択しなければならない 場合、縦の並びの中の 1 つの項目がメインパス上に表示されます。

  複数の項目から 1 つを選択することがオプションの場合は、縦の並び全体がメインパスの下側に表示されます。



- メインライン上の左に戻る矢印 (繰り返し矢印) は、縦に並んだ項目から複数の項目を選択できること、または同じ項目を繰り返すことができることを示します。ブランク以外の分離文字も、次のように示されます。

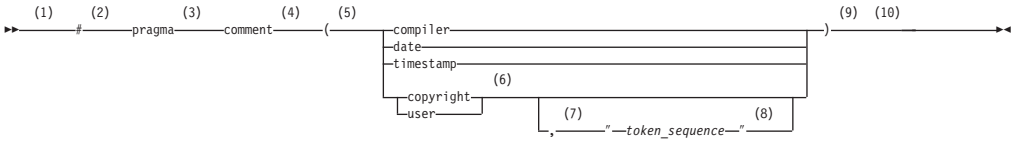


- デフォルトの項目は、メインパスの上側に表示されます。

- キーワードは非イタリック体の文字で示され、ここに示されたとおり正確に入力する必要があります。
- 変数はイタリック体の小文字で示されます。変数はユーザー指定の名前や値を表します。
- 句読記号、括弧、算術演算子、またはその他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する必要があります。

構文図の例

次の構文図の例に、**#pragma comment** ディレクティブの構文を示します。



注:

1 これが構文図の始まりです。

2 記号 # で始まります。

3 キーワード pragma は # 記号の後に続けます。

4 pragma comment の名前は、キーワード pragma の後に続けます。

5 ここに左括弧が必要です。

6 コメント・タイプとして、compiler、date、timestamp、copyright、または user のいずれかのタイプを入力します。

7 コメント・タイプ copyright または user とオプションの文字ストリングの間にはコンマを挿入します。

8 文字ストリングをコンマの後に続けます。この文字ストリングは二重引用符で囲みます。

9 ここに右括弧が必要です。

10 これが構文図の終わりです。

次の #pragma comment ディレクティブの例は、上記の図に従った構文上正しいものです。

```
#pragma comment(date)
#pragma comment(user)
#pragma comment(copyright,"This text will appear in the module")
```

本書での例

本書の例は、特に断りのない限り、ストレージの節約、エラーのチェック、高速な処理パフォーマンスの実現、特定の成果を達成するために使用可能なすべての方法の提示などの試みを省略して、単純な形式でコーディングされています。

インストール情報の例は、「*例*」または「*基本例*」というラベルが付いています。*基本例* は、基本的な、あるいはデフォルトのインストール中に実行される手順の文書化を意図したものであり、これらの基本例を変更する必要はほとんど、あるいはまったくありません。

関連情報

以下のセクションには、XL Cの関連情報が記載されています。

IBM XL C の資料

XL C には、次の形式の製品情報が用意されています。

- README ファイル

  README ファイルには、製品情報への変更と修正を含む最新の情報が収められています。 README ファイルは、デフォルトでは XL C ディレクトリーにあり、インストール CD のルート・ディレクトリーにもあります。
- インストール可能なマニュアル・ページ

  マニュアル・ページは、製品と共に提供されるコンパイラー呼び出しおよびすべてのコマンド行ユーティリティーに対して提供されています。マニュアル・ページのインストールおよびアクセス手順は、「*IBM XL C for AIX, V10.1 インストール・ガイド*」に記載されています。

- インフォメーション・センター

  検索可能な HTML ファイルのインフォメーション・センターをネットワーク上に立ち上げて、リモート側またはローカル側でアクセスすることができます。オンラインのインフォメーション・センターのインストールおよびアクセス手順は、「*IBM XL C for AIX, V10.1 インストール・ガイド*」に記載されています。

  インフォメーション・センターは、Web サイト (http://publib.boulder.ibm.com/infocenter/comphelp/v101v121/index.jsp) で表示できます。

- PDF 文書

  PDF 文書は、デフォルトでは /usr/vac/doc/*LANG*/pdf/ ディレクトリー (ここで、*LANG* は en_US、zh_CN または ja_JP のいずれかです) にあります。 PDF ファイルは、Web (http://www.ibm.com/software/awdtools/caix/library) から入手することもできます。

  以下が、XL C のすべての製品情報を構成するファイルです。

表 2. XL C PDF ファイル

文書タイトル	**PDF ファイル名**	**説明**
IBM XL C for AIX, V10.1 インストール・ガイド, GC88-4921-00	install.pdf	XL C のインストール、および基本コンパイルとプログラム実行用の環境の構成に関する情報が含まれます。
IBM XL C for AIX, V10.1 はじめに, GC88-4919-00	getstart.pdf	XL C 製品の紹介、および環境のセットアップと構成、プログラムのコンパイルとリンク、コンパイル・エラーのトラブルシューティングに関する情報が含まれます。
IBM XL C for AIX, V10.1 コンパイラー・リファレンス, SC88-4917-00	compiler.pdf	並列処理に使用されるものを含めた、さまざまなコンパイラー・オプション、プラグマ、マクロ、環境変数、および組み込み関数に関する情報が含まれます。
IBM XL C for AIX, V10.1 ランゲージ・リファレンス, SC88-4956-00	langref.pdf	ポータビリティーおよび非著作権標準への準拠に対応した言語拡張機能などの、IBM がサポートする C プログラミング言語に関する情報が含まれます。
IBM XL C for AIX, V10.1 最適化およびプログラミング・ガイド, SC88-4920-00	proguide.pdf	拡張プログラミングのトピック (アプリケーション・ポーティング、Fortran コードによる言語間呼び出し、ライブラリー開発、アプリケーションの最適化と並列化、および XL C ハイパフォーマンス・ライブラリーなど) の情報が含まれます。

  PDF ファイルを読むには、Adobe® Reader を使用してください。Adobe Reader がない場合は、Adobe Web サイト (http://www.adobe.com) からダウンロードすることができます (ライセンス条項の対象となります)。

Redbooks、ホワイト・ペーパー、チュートリアル、およびその他の論文など、XL C に関する詳しい情報は、次の Web サイトから入手できます。

http://www.ibm.com/software/awdtools/caix/library

標準および仕様

XL C は、以下の標準および仕様をサポートするように設計されています。本書に記載された機能の一部について、正確な定義を確認するには、これらの標準を参照することができます。

- *Information Technology – Programming languages – C, ISO/IEC 9899:1990*、*C89* とも呼ばれています。
- *Information Technology – Programming languages – C, ISO/IEC 9899:1999*、*C99* とも呼ばれています。
- *Information Technology – Programming languages – Extensions for the programming language C to support new character data types, ISO/IEC DTR 19769*. このドラフト技術レポートは、C 標準委員会によって受諾されており、http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1040.pdf から入手できます。
- *AltiVec Technology Programming Interface Manual*、Motorola Inc。ベクトル処理テクノロジーをサポートするためのベクトル・データ型に関するこの仕様は、http://www.freescale.com/files/32bit/doc/ref_manual/ALTIVECPIM.pdf から入手できます。
- *Information Technology – Programming Languages – Extension for the programming language C to support decimal floating-point arithmetic, ISO/IEC WDTR 24732*. このドラフト技術レポートは C 標準委員会に提出済みであり、http://www.open-std.org/JTC1/SC22/WG14/www/docs/n1176.pdf から入手できます。
- *Decimal Types for C++: Draft 4* http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2006/n1977.html

その他の IBM 資料

- *AIX コマンド・リファレンス 第 1 巻 - 第 6 巻*、*SC88-6875*
- *Technical Reference: Base Operating System and Extensions, Volumes 1 & 2*、*SC23-4913*
- *AIX ナショナル・ランゲージ・サポート ガイドおよびリファレンス*、*SC88-6933*
- *AIX プログラミングの一般概念: プログラムの作成とデバッグ*、*SC88-6931*
- *AIX Assembler Language Reference*、*SC23-4923*

  AIX の全資料は http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp から入手できます。
- *Parallel Environment for AIX: Operation and Use*
- *ESSL for AIX V4.2 Guide and Reference*、*SA22-7904*、http://publib.boulder.ibm.com/clresctr/windows/public/esslbooks.html から入手できます。

その他の資料

- *Using the GNU Compiler Collection*、http://gcc.gnu.org/onlinedocs から入手可能

技術サポート

追加の技術サポートは、XL C のサポート・ページ (http://www.ibm.com/software/awdtools/caix/support) から利用することができます。このページには、膨大な技術情報やその他のサポート情報を対象に検索が行える機能を備えたポータルが用意されています。

必要な情報が見つからない場合は、compinfo@ca.ibm.com に E メールを送信してください。

XL C の最新情報については、http://www.ibm.com/software/awdtools/caix の製品情報サイトにアクセスしてください。

第 1 章 XL C の紹介

IBM XL C for AIX, V10.1 は高機能で高性能なコンパイラーであり、 Fortran プログラムでの言語間呼び出しを含め、複雑で計算負荷の高いプログラムの開発に使用することができます。

この章では、XL C のコンパイラーの機能についての概要を述べます。この章は、コンパイラーの評価を行う方、およびこの製品についてさらに知識を得たい新規ユーザーのために書かれています。

他の IBM コンパイラーとの共通点

IBM XL C for AIX, V10.1 は、IBM C、C++、Fortran コンパイラーの大規模なファミリーの一部です。

XL C は、XL C++ および XL Fortran と一緒になって、XL コンパイラーのファミリーを形成します。

これらのコンパイラーは、種々のプラットフォームおよびプログラミング言語用のコンパイラー機能と最適化テクノロジーを共有する共通のコード・ベースから派生されました。プログラミング環境には、IBM AIX、IBM Blue Gene/L™、IBM Blue Gene/P™、 Cell Broadband Engine™ アーキテクチャー、IBM i5/OS®、選択された Linux® 配布版、IBM z/OS®、および IBM z/VM® が含まれます。この共通のコード・ベースは、国際的なプログラム言語規格へのコンパイラーの準拠と共に、複数のオペレーティング・システムおよびハードウェア・プラットフォームにわたる整合性のあるコンパイラーのパフォーマンスおよびプログラムの移植の容易さをサポートするのに役立ちます。

ハードウェアおよびオペレーティング・システムのサポート

IBM XL C for AIX, V10.1 は、 AIX 5L™ for POWER™ バージョン 5.3 および AIX バージョン 6.1 をサポートします。 要件の完全なリストについては、README ファイルおよび 「*XL C インストール・ガイド*」の 『XL C のインストール前の作業』 を参照してください。

コンパイラー、そのライブラリー、およびその生成されたオブジェクト・プログラムは、必要なソフトウェアおよびディスク・スペースを備えた システム上で実行することができます。

サポートされているさまざまなハードウェア構成を最大限に活用するために、コンパイラーには、特にコンパイルされたアプリケーションの実行に使用されるハードウェアのタイプに特化した性能チューニング・アプリケーションのためのオプションがあります。

© Copyright IBM Corp. 1996, 2008

高度に構成が可能なコンパイラー

多様なコンパイラー呼び出しコマンドおよびオプションを使用して、ユーザーのユニークなコンパイル要件に合うようにコンパイラーを調整できます。

コンパイラー呼び出しコマンド

XL C には、コンパイラーを呼び出すときに使用できる、さまざまなコマンドが用意されています。例えば、xlc、c99、および c89 です。各呼び出しコマンドは、特定の言語レベル仕様を満たすためにコンパイル出力を調整するよう、コンパイラーに指示するという点で固有です。コンパイラー呼び出しコマンドは、すべての標準化された C 言語レベル、および多くの使用頻度の高い拡張機能をサポートするように提供されています。

コンパイラーはまた、ほとんどの呼び出しコマンドについて、対応する「**_r**」バージョンも提供しています。例えば、xlc_r です。「**_r**」呼び出しは、コンパイラーに、オブジェクト・ファイルをスレッド・セーフのコンポーネントおよびライブラリーにリンクしてバインドするよう指示し、コンパイラーが作成するデータおよびプロシージャーのためのスレッド・セーフのオブジェクト・コードを作成します。

XL C コンパイラー呼び出しコマンドについて詳しくは、「*XL C コンパイラー・リファレンス*」の 『コンパイラーの呼び出し』 を参照してください。

コンパイラー・オプション

コンパイラーの動作を制御するために、さまざまなコンパイラー・オプションから選択することができます。さまざまなカテゴリーのオプションは、ユーザーがアプリケーションをデバッグし、アプリケーションのパフォーマンスを最適化して調整し、他の C コンパイラーがサポートする非標準の機能および動作との互換性のために言語レベルおよび拡張機能を選択し、それなしではソース・コードの変更が必要になるような他の多くの共通のタスクを行うのに役立ちます。

XL C では、環境変数、コンパイラー構成ファイル、コマンド行オプション、およびプログラム・ソースに組み込まれているコンパイラー指示ステートメントの組み合わせを通じてコンパイラー・オプションを指定します。

XL C コンパイラー・オプションについて詳しくは、「*XL C コンパイラー・リファレンス*」の『コンパイラー・オプション・リファレンス』 を参照してください。

カスタム・コンパイラー構成ファイル

インストール・プロセスは、コンパイラー・オプションのデフォルト設定を定義するスタンザを含む、デフォルトのプレーン・テキストのコンパイラー構成ファイルを作成します。

ユーザーのコンパイル上の必要によって、XL C が提供するデフォルトの設定値以外のコンパイラー・オプション設定値を指定することが要求されるという状況がしばしば起こり得ます。その場合は Make ファイルを使用して、コンパイラー・オプションを定義することができます。または、代わりにカスタム構成ファイルを作成して、頻繁に使用するコンパイラー・オプション設定の独自のセットを定義することができます。

詳しくは、 19 ページの『カスタム・コンパイラー構成ファイルの使用』を参照してください。

言語標準への準拠

コンパイラーは、C に関する以下のプログラム言語仕様をサポートします。

- ISO/IEC 9899:1999 (C99)
- ISO/IEC 9899:1990 (C89 と呼ばれる)

標準化された言語レベルに加えて、XL C は、以下を含む言語拡張機能もサポートします。

- OpenMP (移植可能な並列化されたプログラミングをサポートするため)
- ベクトル・プログラミングをサポートする言語拡張機能
- GNU C 言語拡張機能のサブセット

C 言語仕様および拡張について詳しくは、 「*XL C ランゲージ・リファレンス*」の『サポートされている言語標準』 「*XL C ランゲージ・リファレンス*」の『言語標準』を参照してください。

GNU との互換性

XL C は、gcc コンパイラーを使用して開発されたアプリケーションの移植を容易にするために、GNU コンパイラー・コマンド・オプションのサブセットをサポートします。

このサポートは、gxlc 呼び出しコマンドが 選択された GNU コンパイラー・オプションと一緒に使用される場合に使用可能です。可能な場合は、コンパイラーを呼び出す前に、コンパイラーは GNU オプションを、それぞれに対応する XL C コンパイラー・オプションにマップします。

これらの呼び出しコマンドは、プレーン・テキストの構成ファイルを使用して、GNU-to-XL C オプション・マッピングおよびデフォルトを制御します。ユーザーは、この構成ファイルをカスタマイズして、ユーザーのあらゆる固有のコンパイル要件をよりいっそう満たすことができます。詳しくは、 「*XL C コンパイラー・リファレンス*」の 『gxlc による GNU C コンパイラー・オプションの再使用』 を参照してください。

ソース・コード・マイグレーションおよび規格合致検査

XL C は、コンパイラーの起動コマンドの提供により、ユーザーの既存の C ソース・コードに対する投資を保護するのに役立ちます。その起動コマンドは、特定の言語レベルにユーザーのアプリケーション・コードをコンパイルするようにコンパイラーに指示します。

ユーザーは、また、-qlanglvl コンパイラー・オプションを使用して特定の言語レベルを指定することができ、コンパイラーは、ユーザーのソース・プログラム内の言語または言語拡張エレメントがその言語レベルに準拠しない場合、警告、エラー、および重大エラー・メッセージを出します。

詳しくは、「*XL C コンパイラー・リファレンス*」の 『qlanglvl』 を参照してください。

ライブラリー

XL C には、さまざまなライブラリーが入っているランタイム環境が組み込まれています。

Mathematical Acceleration Subsystem ライブラリー

Mathematical Acceleration Subsystem (MASS) ライブラリーは、サポートされているプロセッサー・アーキテクチャーでの最適なパフォーマンス用に調整されたスカラーおよびベクトルの数学組み込み関数から成り立っています。 ユーザーは、MASS ライブラリーを選択して、広範囲のプロセッサー上での高性能コンピューティングをサポートするか、または特定のプロセッサー・ファミリーをサポートするために調整されたライブラリーを選択することができます。

MASS ライブラリー機能は、32 ビットおよび 64 ビットの両方のコンパイル・モードをサポートし、スレッド・セーフであり、そのパフォーマンスはデフォルトの libm 数学ライブラリー・ルーチンよりも改良されています。 これらのライブラリーは、ユーザーが自身のアプリケーション用に特定のレベルの最適化を要求したときに、自動的に呼び出されます。ユーザーは、また、最適化オプションが有効であるかないかに関係なく、MASS ライブラリー関数への明示的呼び出しを行うことができます。

詳しくは、 「*XL C プログラミング・ガイド*」の『Mathematical Acceleration Subsystem の使用』を参照してください。

Basic Linear Algebra Subprograms

高性能代数関数の Basic Linear Algebra Subprograms (BLAS) セットは、libxlopt ライブラリーに入れて配送されます。 これらの関数を使用すると、以下のことが行えます。

- 汎用行列またはその転置用の行列ベクトルの積を計算します。
- 汎用行列またはそれらの転置用の複合行列の乗算および追加を実行します。

BLAS 関数の使用に関する詳細については、 「*XL C プログラミング・ガイド*」の『Basic Linear Algebra Subprograms の使用』を参照してください。

その他のライブラリー

以下も、 XL C と一緒に配送されます。

- SMP Runtime Library は、明示的並列処理および自動化された並列処理の両方をサポートします。「*XL C プログラミング・ガイド*」の『SMP Runtime Library』を参照してください。
- メモリー・デバッグ・ランタイム・ライブラリーは、メモリー・リークの診断に使用されます。「*XL C プログラミング・ガイド*」の 「メモリー・ヒープの使用」を参照してください。

ツールおよびユーティリティー

XL C に組み込まれている、多くのツールおよびユーティリティーがあります。

vacndi これは、XL C を、非デフォルトのディレクトリー・ロケーションにインストールする際に使用できるスクリプトです。

IBM Debugger for AIX

IBM Debugger for AIX によって、ローカルまたはリモート側で実行中のプログラムのエラーを検出および診断することができます。コンパイルされた言語固有のブレークポイントを設定し、実行を中断し、ご自分のコードをステップスルーし、変数の内容を検査したり変更したりすることによって、ご自身のプログラムの実行を制御することができます。

このデバッガーには、特定のプログラム言語に固有なビューおよび機能が組み込まれています。コンパイルされた言語ビューを使用すれば、デバッグしているアプリケーションの変数、式、レジスター、メモリー、およびアプリケーション・モジュールをモニターすることができます。

CreateExportList コマンド

特定のオブジェクト・ファイルのセット内にあるすべてのグローバル・シンボルのリストが入っているファイルを作成します。

cleanpdf コマンド

Profile-Directed Feedback (PDF) に関連したコマンド。 cleanpdf は、PDF データが書き込まれるディレクトリーからすべてのプロファイル情報を除去します。

mergepdf コマンド

Profile-Directed Feedback (PDF) に関連したコマンド。 mergepdf は、複数の PDF レコードを単一のレコードに結合するときに、それらのレコードの重要性を評価する機能を提供します。 PDF レコードは、同じ実行可能モジュールから得られたものである必要があります。

resetpdf コマンド

cleanpdf コマンドの現在の動作は、resetpdf コマンドと同じであり、他のプラットフォーム上での前のリリースとの互換性のために保存されています。

showpdf コマンド

showpdf コマンドは、Profile-Directed Feedback トレーニング実行 (オプション -qpdf1 および -qshowpdf のもとでのコンパイル) で実行されるすべてのプロシージャーの呼び出しおよびブロック数を表示します。

gxlc ユーティリティー

gxlc 呼び出しは、 GNU C 呼び出しコマンドを対応する xlc コマンドに変換し、IBM XL C for AIX, V10.1 コンパイラーを呼び出します。 このユーティリティーの目的は、GNU コンパイラーで作成された既存のアプリケーション用に使用される makefile への変更の数を最小化し、IBM XL C for AIX, V10.1 への変換を容易にすることにあります。

プログラムの最適化

XL C は、プログラムの最適化またはパフォーマンスを制御するのに役立ついくつかのコンパイラー・オプションを提供します。

これらのオプションを使用して、以下に挙げることを行えます。

- さまざまなレベルのコンパイラーの最適化を選択する。
- ループ、浮動小数点、その他のタイプの操作に関する最適化を制御する。
- プログラムが実行される場所に応じて、プログラムを、特定のクラスのマシンまたは非常に特定度の高いマシン構成に合わせて最適化する。

変換の最適化によって、アプリケーションの全体的な実行パフォーマンスを向上させることができます。 C は、サポートされるさまざまなハードウェアに合わせた変換の最適化のポートフォリオを提供します。このような変換では、以下に挙げることを行えます。

- クリティカルな操作に対して実行する命令の数を減らす。
- 生成されたオブジェクト・コードを再編成して、PowerPC® アーキテクチャーの使用を最適化する。
- メモリー・サブシステムの使用を向上させる。
- 大量の共用メモリー並列処理を扱うアーキテクチャーの能力を活用する。

詳細については、以下を参照してください。

- 「*XL C プログラミング・ガイド*」の『アプリケーションの最適化』
- 「*XL C コンパイラー・リファレンス*」の『最適化およびチューニング・オプション』
- 「*XL C コンパイラー・リファレンス*」の『コンパイラー組み込み関数』

64 ビット・オブジェクトの機能

XL C コンパイラーの 64 ビット・オブジェクトの機能は、より大きいストレージ要件およびより強力な処理能力に対する増大する要求に対処するものです。

AIX オペレーティング・システムは、64 ビットのアドレス・スペースの使用を通じて 64 ビットのプロセッサーを活用するプログラムを開発し、実行できるようにする環境を提供します。

64 ビット・アドレス・スペース内に収めることのできる大きな実行可能ファイルをサポートするために、別個の 64 ビット・オブジェクト形式が使用されます。 バインダーはこれらのオブジェクトをバインドして、64 ビット実行可能ファイルを作成します。一緒にバインドされるオブジェクトは、すべて同じオブジェクト形式になっている必要があります。 以下のシナリオは許されず、ロード、実行、あるいはその両方に失敗します。

- 32 ビット・ライブラリーまたは共用ライブラリーからの、シンボルへの参照をもっている 64 ビット・オブジェクトまたは実行可能モジュール
- 64 ビット・ライブラリーまたは共用ライブラリーからの、シンボルへの参照をもっている 32 ビット・オブジェクトまたは実行可能モジュール

- 32 ビット・モジュールを明示してロードしようとする 64 ビット実行可能モジュール
- 64 ビット・モジュールを明示してロードしようとする 32 ビット実行可能モジュール
- 32 ビット・プラットフォーム上で 64 ビット・アプリケーションを実行しようとする試み

32 ビット実行可能モジュールは、現在 32 ビット・プラットフォームで実行しているのと同じように、64 ビット・プラットフォームおよび 32 ビット・プラットフォームの両方で、依然として実行可能です。

XL C は、主に -q64 コンパイラー・オプションおよび -qarch コンパイラー・オプションを使用することによって 64 ビット・モードをサポートします。この組み合わせは、目標のアーキテクチャー用のビット・モードと命令セットを決めます。

詳しくは、 「*XL C プログラミング・ガイド*」の 『32 ビット・モードおよび 64 ビット・モードの使用』 を参照してください。

共用メモリーの並列処理

XL C は、マルチプロセッサー・システム体系についてのアプリケーション開発をサポートします。

XL C での並列化アプリケーションの開発には、以下に挙げるいずれの方法でも使用することができます。

- ディレクティブ・ベースの共用メモリー並列処理 (OpenMP、SMP)
- コンパイラーへ、共用メモリー並列処理を自動的に生成するよう指示する
- メッセージ引き渡しベースの共用または分散メモリー並列処理 (MPI)
- POSIX スレッド (Pthreads) 並列処理
- `fork()` および `exec()` を使用した下位 UNIX® 並列処理

AIX オペレーティング・システムの並列プログラミング機能はスレッドの概念に基づいています。並列プログラミングは、既存の単一プロセッサー・システムでの完全なバイナリー互換性を維持しながら、マルチプロセッサー・システムの利点を活用します。これは、単一プロセッサー・システムで作業するマルチスレッドのプログラムは、再コンパイルせずにマルチプロセッサー・システムを活用することができることを意味します。

詳しくは、 「*XL C プログラミング・ガイド*」の『プログラムの並列処理』 を参照してください。

OpenMP ディレクティブ

OpenMP ディレクティブは、XL C および他の多くの IBM および IBM 以外の C、C++、および Fortran コンパイラーによってサポートされる API ベースのコマンドのセットです。

ユーザーは、OpenMP ディレクティブを使用して、特定のループを並列化する方法をコンパイラーに指示することができます。ソースにディレクティブがあると、コンパイラーが並列コードに対して並列分析を実行する必要がなくなります。

OpenMP ディレクティブは、並列処理に必要なインフラストラクチャーを提供する Pthread ライブラリーの存在を必要とします。

OpenMP ディレクティブは、アプリケーションを並列化するための重要な 3 つの問題に対処します。

1. 節 (clause) およびディレクティブは、変数のスコーピングに使用できます。変数を共用すべきではない場合が頻繁に起こります。いいかえれば、プロセッサーはそれぞれ、それ自身の変数のコピーを持っている必要があります。
2. 作業共用ディレクティブは、コードの並列領域に入っている作業を複数の SMP プロセッサー間で分散させる方法を指定します。
3. ディレクティブは、複数のプロセッサー間での同期を制御するのに使用できます。

このリリースから、 XL C は、 OpenMP API バージョン 3.0 仕様をサポートします。この機能によって導入された変更の概要については、 9 ページの『OpenMP 3.0』 を参照してください。

プログラムのパフォーマンスの最適化について詳しくは、以下を参照してください。

- 「*XL C プログラミング・ガイド*」の『アプリケーションの最適化』
- www.openmp.org

診断リスト

コンパイラー出力リストには、アプリケーションをより効率的に開発したりデバッグするのに役立つ重要な情報が提供されていることがあります。

リストされる情報は、組み込むことも、あるいは省略することもできる、任意のセクションで構成されています。適用可能なコンパイラー・オプションおよびリスト作成について詳しくは、 「*XL C コンパイラー・リファレンス*」の『コンパイラー・メッセージおよびリスト表示』 を参照してください。

シンボリック・デバッガー・サポート

コンパイル済みオブジェクトにデバッグ情報を含めるように XL C に指示することができます。その情報は、dbx、IBM Debugger for AIX、AIX XCOFF 実行可能形式をサポートするその他の任意のシンボリック・デバッガー によって調べることができ、プログラムのデバッグに役立ちます。

第 2 章 IBM XL C for AIX, V10.1 の新機能

このセクションでは、 IBM XL C for AIX, V10.1 の新機能および機能拡張について説明します。

オペレーティング・システムのサポート

IBM XL C for AIX, V10.1 は AIX V5.3 と同様に AIX V6.1 をサポートするようになりました。

コンパイラーのこのバージョンでは、AIX V5.2 をサポートしません。

XL C 言語関連の更新

Vector データ型

Vector データ型は、以下の基本データ型で使用できる演算子の一部を使用できるようになりました。

- 単項演算子
- 2 項演算子
- 関係演算子

スレッド・ローカル・ストレージ

スレッド・ローカル・ストレージ・サポートが拡張されて、 `__attribute__((tls-model("string")))` が組み込まれています。ここで、*string* は、 `local-exec`、`initial-exec`、`local-dynamic`、または `global-dynamic` のいずれかです。

OpenMP 3.0

このリリースでは、 XL C は、 OpenMP API バージョン 3.0 仕様をサポートします。 XL C 実装は、OpenMP アプリケーション・プログラム・インターフェース・ドラフト 3.0 パブリック・コメントの IBM の解釈に基づきます。

バージョン 2.5 と、バージョン 3.0 との主な違いは以下のとおりです。

- タスク・レベルの並列化の追加。新規の OpenMP 構成体 `TASK` および `TASKWAIT` は、既存の OpenMP 構成体では適切ではなかったポインター追跡または再帰的アルゴリズムなどの、不規則アルゴリズムを並列化する能力をユーザーに与えます。
- `for` ループには、`signed int` と同様に `unsigned int` の *var* 値、および `pointer` 型を入れることができるようになりました。
- スタック・サイズの制御。OMP ランタイム・ライブラリーによって、新しい環境変数 `OMP_STACKSIZE` を使用して作成されたスレッドのスタックのサイズを制御できるようになりました。

© Copyright IBM Corp. 1996, 2008

- ユーザーは、新しい環境変数 OMP_WAIT_POLICY および OMP_SET_POLICY を使用して、待機スレッドの望ましい動作に関してヒントを与えることができるようになりました。
- ストレージの再使用。PRIVATE 節のいくつかの制限は除去されました。並列構成体の reduction 節に表示されるリスト項目は、ワーク・シェアリング構成体の private 節でも表示できるように成りました。
- スケジューリング。新規の SCHEDULE 属性では、auto がコンパイラーおよび実行時システムにスケジューリングの制御を許可します。
- STATIC スケジュールを指定した連続ループ構成体で nowait を使用できるように成りました。
- ネスト・サポート - DO、FOR、PARALLELL FOR、 および PARALLEL DO ディレクティブに COLLAPSE 節が追加されて完全なループ・ネスト並列化が可能に成りました。これは、1 つのネスト内の複数のループを並列処理できることを意味します。

詳細については、以下を参照してください。

- 「*XL C プログラミング・ガイド*」の『OpenMP ディレクティブの使用』
- www.openmp.org

パフォーマンスおよび最適化

一部のフィーチャーおよび機能拡張を使用して、アプリケーションのパフォーマンス・チューニングと最適化に役立てることができます。

-qstrict の機能拡張

多くのサブオプションが -qstrict オプションに追加されることで、最適化の制御、および厳密なプログラム・セマンティクスに違反するトランスフォーメーションのよりきめの細かい制御が可能になりました。前のリリースでは、-qstrict オプションは、厳密なプログラム・セマンティクスに違反するトランスフォーメーションをすべて使用不可にしていました。これは、サブオプションなしで -qstrict を使用した場合の動作であることには変わりありません。 同様に、以前のリリースでは、-qnoqstrict はプログラムのセマンティクスの変更が可能なトランスフォーメーションを許可しました。高水準の最適化では厳密なプログラム・セマンティクスを緩和することが必要な場合があるため、サブオプションを追加することにより、すべてのセマンティクスの検査をオフにすることなく、高速コードの特定の利点を活用するために、規則を選択的に緩和することができるようになりました。

個別に使用することも、サブオプション・グループを使用して使用することもできる、16 の新しいサブオプションがあります。グループは以下のとおりです。

all 他のサブオプションにより制御されるものを含む、すべてのセマンティクス変更のトランスフォーメーションを使用不可にします。

ieeefp 個々のオペレーションが IEEE 754 セマンティクスに準拠しているかどうかを制御します。

order プログラム言語セマンティクスに違反するような方法で個々のオペレーションを再配列できるかどうかを制御します。

precision
プログラムの結果の精度に影響を与える、最適化およびトランスフォーメーションを制御します。

exceptions
プログラムにより生成される実行時例外に影響を与える、最適化およびトランスフォーメーションを制御します。

これらのサブオプションについて詳しくは、 「*XL C コンパイラー・リファレンス*」の ″-qstrict″ を参照してください。

パフォーマンス関連のコンパイラー・オプションおよびディレクティブ

次の表の項目は、新規または変更されたコンパイラー・オプションおよびディレクティブを説明しています。

ここで提示されている情報は、簡単な概要です。上記およびその他のパフォーマンス関連のコンパイラー・オプションについて詳しくは、 「*XL C コンパイラー・リファレンス*」の 『最適化およびチューニング・オプション』 を参照してください。

表 3. パフォーマンス関連のコンパイラー・オプションおよびディレクティブ

オプション/ディレクティブ	説明
-qstrict	多くの新しいサブオプションが追加され、一定のパフォーマンスの利点を活用できるよう、プログラム・セマンティクス規則の緩和についてユーザーがよりよく制御できるようにしています。
-qreport	リスト表示には、各ループ用に作成されたストリームの数、および非ストライド・ワン参照のため SIMD ベクトル化ができないループについての情報が入るようになりました。ユーザーはこの情報を使用して、アプリケーションのパフォーマンスを改善できます。
-qsmp=omp	-qsmp=omp が有効になっているとき、 OpenMP API 3.0 の追加機能が使用できるようになりました。詳しくは、9 ページの『OpenMP 3.0』を参照してください。

パフォーマンス・チューニングおよびプログラム最適化について詳しくは、 「*XL C プログラミング・ガイド*」の 『アプリケーションの最適化』 を参照してください。

新規または変更されたコンパイラー・オプションおよびディレクティブ

コンパイラー・オプションは、コマンド行上で、あるいはアプリケーションのソース・ファイルに組み込まれているディレクティブを通じて、指定することができます。これらのコンパイラー・オプションの詳細説明および使用法に関する情報については、「*XL C コンパイラー・リファレンス*」を参照してください。

表 4. 新規または変更されたコンパイラー・オプションおよびディレクティブ

オプション/ディレクティブ	説明
-qstrict	多くのサブオプションが -qstrict オプションに追加されることで、最適化の制御、および厳密なプログラム・セマンティクスに違反するトランスフォーメーションの制御が向上しました。詳しくは、10 ページの『パフォーマンスおよび最適化』を参照してください。
-qshowmacros	-E オプションとともに使用した場合、-qshowmacros オプションはプリプロセスされた出力をマクロ定義で置き換えます。定義済みマクロおよびユーザー定義マクロの出力をさらに正確に制御するためサブオプションが提供されています。
-qreport	自動並列化またはベクトル化を使用可能にするコンパイラー・オプションと共に使用すると、 -qreport オプションはループに含まれるストリームの数を報告し、ループが、非ストライド・ワン参照によって SIMD ベクトル化されないときに情報を作成するようになりました。
-qsmp=omp	XL C は、OpenMP 3.0 のいくつかの機能をサポートするようになりました。 詳しくは、9 ページの『OpenMP 3.0』を参照してください。
#pragma init および #pragma fini	プログラマーは #pragma init および #pragma fini を使用して、main() の前または後、あるいは共用ライブラリーがロードまたはアンロードされる際に、実行する関数のリストを指定できます。これらの関数は、初期化およびクリーンアップを実行するために使用できます。 **注:** C++ 呼び出し (xlC など) または再配布可能ツール (linkxlC または makeC++SharedLib) は、リンク時に使用する必要があります。
-qtimestamps	このオプションは生成されたバイナリーからタイム・スタンプを除去するために使用されます。
-qtls	スレッド・ローカル・ストレージ・サポートが拡張されて、`__attribute__((tls-model("`*string*`")))` が組み込まれています。ここで、*string* は、 `local-exec`、`initial-exec`、`local-dynamic`、または `global-dynamic` のいずれかです。
-qinfo	サブオプション `als` および `noals` は、ANSI 別名割り当て規則の考えられる違反を報告する (または報告しない) ために qinfo オプションに追加されました。
-qunique	-qunique は、C および C++ の両方に適用できるようになりました。

バージョン 9.0 に追加された機能拡張

このセクションでは、前のバージョン バージョン 9.0 のコンパイラーの新機能および機能拡張について説明します。

C 言語関連の更新

C コンパイルのデフォルト言語レベルが変更され、 C プログラミング言語に対する拡張機能が新規のサポートが導入されました。

デフォルト言語レベルの変更 - extc99

デフォルトの -qlanglvl コンパイラー・オプションの設定は、xlc 呼び出しを使用して C コンパイラーを呼び出す場合は、extc99 のままです。この変更によって、extc99 サブオプションを明示的に指定せずに C99 機能およびヘッダーを使用することができます。

新規デフォルトの -qlanglvl=extc99 設定でコンパイルした場合、以下の問題が起こる場合があります。

- C99 では、ポインターは、restrict で修飾できます。そのため、restrict は識別子として使用できません。
- long long データの C99 における扱いは、 C89 における long long データの処理方法とは異なります。
- C99 ヘッダー・ファイルは、新しいマクロ LLONG_MAX を limits.h で、va_copy を stdarg.h で定義します。
- マクロ __STDC_VERSION__ の値が、199409 から 19990 に変更されています。

以前の xlc の動作に戻すためには、コンパイラーの呼び出し時に、-qlanglvl=extc89 を指定します。

C の 10 進浮動小数点のサポート

10 進浮動小数点数演算によって、数値データの I/O が通常は 10 進形式で行われる業務および財務アプリケーションにおいて、計算の性能と精度が向上しました。固有の丸め誤差がデータ変換中に累積すると、10 進型と 2 進浮動小数点型間のデータ変換が回避されます。

XL C では、以下の新規のコンパイラー・オプションを指定した 10 進の浮動小数点数演算のサポートが追加されました。

表 5. 10 進浮動小数点コンパイラー・オプション

オプション/ディレクティブ	説明
-qdfp|-qnodfp	-qdfp を指定すると、10 進浮動小数点のデータ・タイプおよびリテラルのコンパイラー・サポートが使用可能になります。 10 進浮動小数点計算をサポートしないアーキテクチャー用にコンパイルするとき -qdfp を指定すると、コンパイラーは -qfloat=dfpemulate とみなし、10 進浮動小数点の計算のソフトウェア・エミュレーションを使用可能にします。
-qfloat= dfpemulate|nodfpemulate	-qfloat=dfpemulate を指定すると、10 進浮動小数点の計算を処理するときはソフトウェア・エミュレーションを使用するようにコンパイラーに指示します。 このオプションは XL Cによりサポートされるすべてのアーキテクチャーで使用できます。
y	定数式の丸めを制御する y オプション用の 10 進浮動小数点算術に特定のサブオプションがあります。

注: 10 進浮動小数点演算のコンパイラー・サポートには、5300-06 Technology Level の AIX 5L for POWER バージョン 5.3 またはそれ以降が必要です。

詳細については、Extension for the programming language C to support decimal floating-point arithmetic: TR 24732を参照してください。

アーキテクチャーおよびプロセッサー・サポート

-qarch および -qtune コンパイラー・オプションは、コンパイラーにより生成されるコードを制御します。これらのコンパイラー・オプションは、命令、スケジューリング、およびその他の最適化を調整して、指定されたターゲット・プロセッサーまたはプロセッサーの範囲に最適なパフォーマンスが得られるようにします。

-qarch、-qtune の新規デフォルト設定

新規デフォルト -qarch および -qtune の設定は以下のとおりです。

- -qarch=ppc
- -qtune=balanced

-qtune=balanced サブオプションは今回のリリースからのもので、特定の -qarch 設定が指定された場合のデフォルトの -qtune 設定になります。-qtune=balanced を使用すると、POWER6™ などの新しいプロセッサー・アーキテクチャーの範囲全体で、生成されたコードを調整してパフォーマンスを最適にするようにコンパイラーに指示します。

重要事項: -qarch デフォルト・サブオプション設定への変更は、ユーザーのプログラムでの浮動小数点単精度算術計算の結果に影響を及ぼす可能性があります。 以前のリリースのコンパイラーで使用されていた -qarch=com デフォルトは、そうした計算を倍精度命令を使用して実行してから丸めることにより行っていました。新規の -qarch=ppc デフォルトは、単精度浮動小数点命令を使用するコードを生成するようにコンパイラーに指示します。 計算方法の違いは、計算結果の精度に影響を及ぼすことがあります。以前の -qarch=com デフォルトの動作を実現するには、アプリケーションのコンパイル時に、新規の -qfloat=nosingle コンパイラー・オプションを指定します。

POWER6 プロセッサーの新規サポート

XL C バージョン 9.0 は -qarch および -qtune サブオプションのリストを拡張して、新規に使用可能な POWER6 プロセッサーをサポートします。

以下の -qarch および -qtune オプションが現在使用可能です。

- -qarch=pwr6
- -qarch=pwr6e
- -qtune=pwr6

-qipa コンパイラー・オプションも、新規のアーキテクチャー・クローン作成サブオプションを追加して、POWER6 プロセッサーでのプロシージャー間分析 (IPA) の最適化をサポートします。

- -qipa=clonearch=pwr6

選択したプロセッサーのサポートの除去

XL C バージョン 9.0 は、POWER、 POWER2™、および PowerPC 601 などの、AIX V5.2 がサポートしていないプロセッサー・アーキテクチャーのサポートを除去しました。 その結果、次の -qarch および -qtune サブオプション設定はサポートされなくなりました。

- -qarch= com | pwr | pwr2 | pwr2s | p2sc | 601 | 602 | 603
- -qtune= pwr | pwr2 | pwr2s | pwrx | p2sc | 601 | 602 | 603

コンパイラーはこれらのサブオプション設定の認識を継続し、依然としてそれらに対応するアーキテクチャー用のコードを生成します。ただし、一部のケースでは、コードの動作が以前のバージョンのコンパイラーで生成されたコードと異なることがあります。また、これらのサポートされていないアーキテクチャー用に生成されたコードは、サポートされている AIX システムでは、実行されないこともあります。これはアーキテクチャーが異なるためです。

サポートされない -qarch および -qtune サブオプション設定を今後も使用する場合は、注意してください。

パフォーマンスおよび最適化

パフォーマンス・チューニングおよびプログラム最適化を支援する多くの機能拡張が行われました。

パフォーマンス関連のコンパイラー・オプションおよびディレクティブ

次の表の項目は、新規または変更されたコンパイラー・オプションおよびディレクティブを説明しています。

ここで提示されている情報は、単なる簡単な概要です。上記およびその他のパフォーマンス関連のコンパイラー・オプションについて詳しくは、 「*XL C コンパイラー・リファレンス*」の 『最適化およびチューニング・オプション』 を参照してください。

表 6. パフォーマンス関連のコンパイラー・オプションおよびディレクティブ

オプション/ディレクティブ	説明
-qalias= global|noglobal	これらの新規の -qalias サブオプションによって、リンク時の最適化中に、コンパイル単位全体で言語固有の別名割り当て規則のアプリケーションを使用可能または使用不可にすることができます。
-qalias= restrict|norestrict	これらの新規の -qalias サブオプションは、制限修飾ポインターの最適化を有効または無効にします。 -qalias=restrict の指定によって、通常は制限修飾ポインターを使用するコードのパフォーマンスが向上します。-qalias=norestrict を使用すると、V9.0 より前のバージョンのコンパイラーでコンパイルされたコードとの互換性を保存することができます。

表 6. パフォーマンス関連のコンパイラー・オプションおよびディレクティブ (続き)

オプション/ディレクティブ	説明
-qfloat= fenv|nofenv	これらの新規の -qfloat サブオプションは、浮動小数点の状況および制御レジスターの明示的な読み取りまたは書き込みのように、コードが浮動小数点ハードウェア環境で依存関係を持つかどうかをコンパイラーに通知します。 -qfloat=nofenv の指定は、ハードウェア環境での依存関係がないことを指示し、コンパイラーは積極的な最適化を実行することができます。
-qfloat= hscmplx|nohscmplx	-qfloat=hscmplx の指定によって、複素数の除法および複素数の絶対値に関連する演算の最適化が向上します。
-qfloat= rngchk|norngchk	-qfloat=rngchk の指定によって、ソフトウェア除算およびインラインの根号演算の入力引数の範囲検査が可能になります。 -qfloat=norngchk の指定は、コンパイラーに範囲検査をスキップするように指示し、特定の状況でのパフォーマンスを向上させることができます。-qnostrict コンパイラー・オプションの指定によって、-qfloat=norngchk が設定されます。
-qfloat= single|nosingle	-qfloat=single の指定は、現行のすべての PowerPC プロセッサーでサポートされている単精度算術演算命令を使用して単精度浮動小数点値を計算するようにコンパイラーに指示します。 POWER および POWER2 プロセッサーのような以前のプロセッサー用に元々コンパイルされたアプリケーションでの計算動作を保持する必要がある場合は、-qfloat=nosingle を使用します。場合によっては、同じ計算結果を取得するには、-qfloat=norndsngl を指定する必要もあります。
-qipa=clonearch=pwr6	-qipa=clonearch コンパイラー・オプションには現在、新規の pwr6 サブオプションが組み込まれ、POWER6 プロセッサーにおけるプロシージャー間分析 (IPA) の最適化をサポートします。
-qipa=threads= [auto|noauto|number]	この新規の -qipa サブオプションを使用すると、コンパイラーが第 2 IPA パス時にコード生成に割り当てるスレッド数を指定できます。
-qminimaltoc|-qnominimaltoc	-qminimaltoc を指定すると、toc エントリーをオブジェクト・ファイルごとに別々のデータ・セクションに入れることによって、 64 ビットのコンパイルでの toc オーバーフロー条件を避けるのに役立ちます。
-qpdf	-qpdf オプションを使用すると、特定のオブジェクトについての Profile-Directed Feedback を提供することができます。詳しくは、 「*XL C プログラミング・ガイド*」の 『オブジェクト・レベル Profile-Directed Feedback』を参照してください。
-qsmp= threshold=n	-qsmp=auto が有効な場合、この新規サブオプションによって、コンパイラーが自動的な並列化を検討する前にループに必要な作業量を指定することができます。

表 6. パフォーマンス関連のコンパイラー・オプションおよびディレクティブ (続き)

オプション/ディレクティブ	説明
-qspeculateabsolutes|-qnospeculateabsolutes	プログラムの最適化中に、 -qnospeculateabsolutes は非 IPA リンクの場合には -qtocmerge -bI:file、IPA リンクの場合は -bI:file を処理し、絶対アドレスにおける変数の推測を使用不可にします。
#pragma expected_value(param, value)	#pragma expected_value ディレクティブを使用して、関数呼び出しで渡されるパラメーターが実行時に利用するのに最も適した値を指定します。コンパイラーはこの情報を使用して、関数のクローン作成やインライン化などの特定の最適化を実行することができます。

バージョン 9.0 の組み込み関数

XL C によって提供される組み込み関数の詳細については、「*XL C コンパイラー・リファレンス*」の『コンパイラー組み込み関数』を参照してください。

PowerPC キャッシュ制御

PowerPC アーキテクチャーは、dcbst および dcbf キャッシュ・コピー命令を指定します。以下の新規組み込み関数によって、これらの命令に直接プログラマーがアクセスできます。

- `void __dcbst(const void* addr);  /* データ・キャッシュ・ブロック・ストア */`
- `void __dcbf(const void* addr);  /* データ・キャッシュ・ブロック・フラッシュ */`

POWER6 プリフェッチ拡張機能およびキャッシュ制御

POWER6 プロセッサーには、ストア・ストリーム・プリフェッチおよびプリフェッチ深さ制御のためのキャッシュ制御およびストリーム・プリフェッチ拡張機能があります。 XL C は、以下の新規組み込み関数を提供して、これらの命令にプログラマーが直接アクセスできるようにしています。

- `void __dcbfl(const void* addr);  /* pwr6 - L1 データ・キャッシュのみからのデータ・キャッシュ・ブロック・フラッシュ */`
- `void __protected_unlimited_stream_set(unsigned int direction, const void* addr, unsigned int ID);  /* pwr5 および pwr6 によりサポートされます */`
- `void __protected_unlimited_store_stream_set(unsigned int direction, const void* addr, unsigned int ID);  /* pwr6 によりサポートされます */`
- `void __protected_store_stream_set(unsigned int direction, const void* addr, unsigned int ID);  /* pwr6 によりサポートされます */`
- `void __protected_stream_count_depth(unsigned int unit_cnt, unsigned int prefetch_depth, unsigned int ID);  /* pwr6 によりサポートされます */`

その他の新規または変更されたコンパイラー・オプション

コンパイラー・オプションは、コマンド行上で、あるいはアプリケーションのソース・ファイルに組み込まれているディレクティブを通じて、指定することができます。これらのコンパイラー・オプションの詳細説明および使用法に関する情報につ

いては、「*XL C コンパイラー・リファレンス*」を参照してください。

表 7. その他の新規または変更されたコンパイラー・オプション

オプション/ディレクティブ	**説明**
-C!	-C! コンパイラー・オプションの指定によって、プリプロセスされた出力からコメントが除去されます。
-qoptdebug|-qnooptdebug	-O3 以上の最適化レベルで使用されると、新規の -qoptdebug オプションは、シンボリック・デバッガーが読み取ることができる最適化された疑似コードを作成するようにコンパイラーに指示します。
-qreport	-qreport オプションは、自動並列化またはベクトル化を使用可能にするコンパイラー・オプションと共に使用すると、プログラム・ループがどのように並列化され、ベクトル化されるかを示す疑似コード・リストを作成します。 このレポートも、コンパイラーが指定されたループを並列化またはベクトル化することができない場合は、診断情報を提供します。
-qsaveopt|-qnosaveopt	以前のリリースでは、 -qsaveopt オプションは、ファイルをコンパイルするのに使用したコマンド行オプションを結果のオブジェクト・ファイルに保管していました。 バージョン 9.0 では、オブジェクト・ファイルに保管される情報が、コンパイル時に呼び出された各コンパイラー・コンポーネントのバージョンおよびレベル情報も含むように拡張されました。
-qsmp=stackcheck	この新規の -qsmp サブオプションは、実行時にスレーブ・スレッドによってスタック・オーバーフローの有無を確認し、残りのスタック・サイズが XLSMPOPTS 環境変数の stackckeck オプションに指定されたバイト数に満たない場合は警告を出すようにコンパイラーに指示します。
-qversion=verbose	-qversion オプションは、新規の verbose サブオプションを追加しています。 -qversion=verbose の指定は、コンパイル時に呼び出されたコンパイラー・コンポーネントのバージョンおよびレベル情報を表示するようにコンパイラーに指示します。

第 3 章 XL C のセットアップとカスタマイズ

完全な前提条件およびインストール情報については、「*XL C インストール・ガイド*」の『インストール前の作業』を参照してください。

カスタム・コンパイラー構成ファイルの使用

コンパイラーの設定およびオプションのカスタマイズは、デフォルトの構成ファイルを変更するか、ユーザー独自の構成ファイルを作成することによって行うことができます。

デフォルトのコンパイラー構成ファイルは、XL C コンパイラーのインストール中に作成されます。この構成ファイルを直接変更して、特定のニーズのためのデフォルト・オプションを追加することができます。ただし、後でコンパイラーに更新を適用する場合、ユーザーが行ったすべての変更を新規にインストールされた構成ファイルに対しても再適用する必要があります。

これを避けるために、独自のカスタム・コンパイラー構成ファイルを作成することができます。コンパイラーは、デフォルト構成ファイル内で指定されているコンパイラー設定とともに、ユーザーのカスタム構成ファイル内で指定したコンパイラー設定を認識して解決する能力を持っています。

カスタム構成ファイルを使用するようコンパイラーに指定した場合、コンパイラーは、デフォルトのシステム構成ファイル内の設定を確認する前に、カスタム構成ファイル内の設定を調べて処理します。後にデフォルト構成ファイルの設定に影響を与える可能性があるコンパイラーの更新でも、カスタム構成ファイル内の設定には影響を与えません。

詳しくは、「*XL C コンパイラー・リファレンス*」の『カスタム・コンパイラー構成ファイルの使用』を参照してください。

© Copyright IBM Corp. 1996, 2008

第 4 章 XL C によるアプリケーションの開発

C アプリケーション開発は、編集、コンパイル、リンク (デフォルトではコンパイルと結合された単一ステップ)、 および実行というサイクルを繰り返すことで成り立っています。

注:

1. コンパイラーを使用する前に、まず XL C が適切にインストールされ、構成されていることを確認する必要があります。 詳細については、「*XL C インストール・ガイド*」を参照してください。
2. C プログラムの作成について学習するには、「*XL C ランゲージ・リファレンス*」を参照してください。

コンパイラー・フェーズ

標準的なコンパイラー呼び出しは、以下に挙げる活動のいくつかまたはすべてを順序どおりに実行します。リンク時の最適化については、一部の活動がコンパイル中に複数回実行されます。各々のプログラムが実行されるときに、結果が、順序に並んでいるものの中の次のステップに送られます。

1. ソース・ファイルのプリプロセッシング
2. コンパイル (指定されるコンパイラー・オプションによって異なりますが、例えば以下のフェーズで構成されます)
   a. フロントエンド構文解析およびセマンティック分析
   b. 高水準最適化
   c. 低水準最適化
   d. レジスター割り振り
   e. 最終アセンブリー
3. アセンブリー (.s) ファイル、および、プリプロセス後の未プリプロセス・アセンブラー (.S) ファイルのアセンブル
4. 実行可能アプリケーションを作成するためのオブジェクト・リンク

コンパイラーがこれらのフェーズをステップスルーするのを確認するには、ユーザーのアプリケーションをコンパイルするときに -v コンパイラー・オプションを指定します。 コンパイラーが各フェーズにかける時間を確認するには、 -qphsinfo を指定します。

C ソース・ファイルの編集

C ソース・プログラムを作成するには、vi または emacs のようなご使用のシステムで使用可能な任意のテキスト・エディターを使用することができます。

ソース・プログラムは、認識されたファイル名接尾部を使用して保管する必要があります。 XL C が認識する接尾部のリストについては、 24 ページの『XL C 入力および出力ファイル』を参照してください。

© Copyright IBM Corp. 1996, 2008

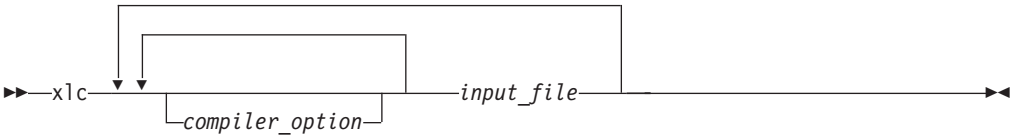
C ソース・プログラムが有効なプログラムであるためには、「*XL C ランゲージ・リファレンス*」で指定されている言語定義に準拠している必要があります。

XL C によるコンパイル

XL C は、コマンド行コンパイラーです。呼び出しコマンドおよびオプションは、特定の C アプリケーションのニーズにしたがって選択できます。

コンパイラーの呼び出し

ソース・プログラムをコンパイルするには、以下に示す基本呼び出し構文を使用します。



コンパイラー呼び出しコマンドは、C ソース・ファイルをコンパイルして、任意の .s ファイルおよび .S ファイルをアセンブルし、オブジェクト・ファイルとライブラリーを 1 つの実行可能プログラムにリンクするのに必要なすべての手順を実行します。

新しいアプリケーションの作業では、 xlc またはスレッド・セーフな対応のものを指定してコンパイルする必要があります。

追加の呼び出しコマンドは、特殊なコンパイル要件を満たすために使用でき、主として、C 言語のさまざまなレベルおよび拡張機能に対する明示的なコンパイル・サポートを提供します。 GNU コンパイル環境からXL C に移行する開発者を支援するための特殊な呼び出しを含む、使用可能なコンパイラー呼び出しコマンドについて詳しくは、 「*XL C コンパイラー・リファレンス*」の 『コンパイラーの呼び出し』 を参照してください。

並列化 XL C アプリケーションのコンパイル

XL C は、マルチプロセッサー環境で使用される並列化アプリケーションをコンパイルするのに使用できるスレッド・セーフのコンパイラー呼び出しコマンドを提供します。

これらの呼び出しは、コンパイル済みのオブジェクトをスレッド・セーフ・コンポーネントおよびライブラリーにリンクしバインドする以外は、対応する基本コンパイラー呼び出しに似ています。以下は、汎用 XL C スレッド・セーフ・コンパイラー呼び出しです。

- xlc_r、xlc_r7、xlc128_r、xlc128_r7

XL C は、特定のコンパイル要件を満たすために、追加スレッド・セーフ呼び出しを提供します。 詳しくは、 「*XL C コンパイラー・リファレンス*」の『コンパイラーの呼び出し』 を参照してください。

注: これらのコマンドのいずれかを単独で使用することは、並列処理を暗黙指定することを意味しません。コンパイラーが SMP または OpenMP ディレクティブを認識し並列処理を活動化するためには、-qsmp コンパイラー・オプションも指定する必要があります。つまり、これらのスレッド・セーフ呼び出しコマンドのうちの 1 つと一緒に -qsmp オプションのみを指定すればいいということです。-qsmp を指定すると、ドライバーは構成ファイルのアクティブ・スタンザにある smp ライブラリー行で指定されたライブラリーにリンクします。

並列処理アプリケーションについて詳しくは、 「*XL C プログラミング・ガイド*」の『プログラムの並列処理』 を参照してください。

コンパイラー・オプションの指定

コンパイラー・オプションは、コンパイラー特性の設定、作成されるオブジェクト・コードの記述、出される診断メッセージの制御、および一部のプリプロセッサー関数の実行など、さまざまな機能を実行します。

コンパイラー・オプションは以下のようにユーザーが指定できます。

- コマンド行コンパイラー・オプションを指定して、コマンド行で
- ソース・コードでディレクティブ・ステートメントを使用して
- Make ファイルで
- コンパイラー構成ファイルにあるスタンザで
- または、これらの手法を任意に組み合わせたものを使用して

複数のコンパイラー・オプションを指定した場合、オプションの競合および非互換が起きる可能性があります。このような競合を整合性のある方法で解決するために、通常、コンパイラーは次の一般的な優先順位をほとんどのオプションに適用します。

1. ソース・ファイルのディレクティブ・ステートメントは、コマンド行設定値を*オーバーライド* する。
2. コマンド行コンパイラー・オプション設定値は、構成ファイル設定値を*オーバーライド* する。
3. 構成ファイル設定値は、デフォルト設定値を*オーバーライド* する。

一般に、コンパイラーを呼び出すときにコマンド行上で同じコンパイラー・オプションが複数回指定されると、最後に指定されたオプションが優先されます。

注: コンパイラー・オプションの中には、上記の優先順位に従わないものもあります。

例えば、-I コンパイラー・オプションは特別なケースです。コンパイラーは、コマンド行で -I を使用して指定されたディレクトリーを検索する前に、vac.cfg ファイル内の -I で指定された任意のディレクトリーを検索します。これらのオプションは、優先的ではなく、累積的です。

コンパイラー・オプションとその使用法に関する詳細については、「*XL C コンパイラー・リファレンス*」を参照してください。

コンパイラー・オプションをリンカー、アセンブラー、およびプリプロセッサーに渡すこともできます。コンパイラー・オプションおよび指定方法について詳しくは、 「*XL C コンパイラー・リファレンス*」の 『コンパイラー・オプション・リファレンス』 を参照してください。

gxlc による GNU C コンパイラー・オプションの再使用

XL C には、 gxlc を含む、 GNU C コンパイラーから XL C への移行に役立つさまざまな機能が含まれています。

このユーティリティーは、GNU C コンパイラー・オプションを受け入れて、それらを同等の XL C オプションに変換します。このユーティリティーは、XL C オプションを使用して、xlc 呼び出しコマンドを作成し、それが次にコンパイラーを呼び出すのに使用するユーティリティーです。 gxlc ユーティリティーは、ユーザーが、以前に GNU C を使用して開発されたアプリケーション用に作成された Make ファイルを再使用するのを援助するために提供されています。 ただし、XL C の機能を十分に活用するためには、xlc および cc 呼び出しコマンドを、それらに関連するオプションと併せて使用する必要があります。

gxlc のアクションは、構成ファイル gxlc.cfg によって制御されます。 XL C に対応するもののある GNU C オプションは、このファイルに示されています。すべての GNU オプションが対応する XL C オプションをもっているわけではありません。 gxlc ユーティリティーは、それが変換できない任意の GNU C オプションに対して警告を戻します。

gxlc オプションのマッピングは変更可能です。 gxlc 構成ファイルの使用については、 「*XL C コンパイラー・リファレンス*」の 『gxlc による GNU C コンパイラー・オプションの再使用』 を参照してください。

XL C 入力および出力ファイル

これらのファイル・タイプが XL C によって認識されます。

コンパイラーが使用する以下のファイル・タイプおよび追加のファイル・タイプに関する詳細については、 「*XL C コンパイラー・リファレンス*」の 『入力ファイルのタイプ』 および 「*XL C コンパイラー・リファレンス*」の『出力ファイルのタイプ』 を参照してください。

表 8. 入力ファイル・タイプ

ファイル名拡張子	説明
.a	アーカイブまたはライブラリー・ファイル
.c	C ソース・ファイル
.i	プリプロセスされたソース・ファイル
.o	オブジェクト・ファイル
.s	アセンブラー・ファイル
.S	プリプロセスされていないアセンブラー・ファイル
.so	共有オブジェクト・ファイル

表 9. 出力ファイル・タイプ

ファイル名拡張子	説明
a.out	コンパイラーによって作成される実行可能ファイルのデフォルト名
.d	Make ファイルに含めるのが適切なターゲット・ファイル
.i	プリプロセスされたソース・ファイル
.lst	リスト・ファイル
.o	オブジェクト・ファイル
.s	アセンブラー・ファイル
.so	共有オブジェクト・ファイル
.u	Make 依存関係ファイル

XL C によるコンパイル済みアプリケーションのリンク

デフォルトでは、ユーザーは、XL C プログラムをリンクするのに、特別なことは何もする必要はありません。コンパイラー呼び出しコマンドは、自動でリンカーを呼び出して実行可能出力ファイルを生成します。

例えば、以下のコマンドを実行すると、

```
xlc file1.c file2.o file3.c
```

`file1.c` および `file3.c` がコンパイルされてオブジェクト・ファイル `file1.o` および `file3.o` が作成され、次にすべてのオブジェクト・ファイル (`file2.o` も含まれる) がリンカーに処理依頼されて 1 つの実行可能モジュールが作成されます。

別個のステップでのコンパイルおよびリンク

後でリンクできるオブジェクト・ファイルを作成するには、-c オプションを使用します。

```
xlc -c file1.c
  # 1 つのオブジェクト・ファイル (file1.o) を作成する
xlc -c file2.c file3.c
  # または複数のオブジェクト・ファイル (file2.o、file3.o) を作成する
xlc file1.o file2.o file3.o
  # オブジェクト・ファイルをデフォルト・ライブラリーとリンクする
```

プログラムのコンパイルおよびリンクについて詳しくは、以下を参照してください。

- 「*XL C コンパイラー・リファレンス*」の 「リンク」

既存の実行可能ファイルの再リンク

リンカーは実行可能ファイルを入力として受け入れるので、既存の実行可能ファイルを更新されたオブジェクト・ファイルにリンクすることができます。

ただし、前に -qipa オプションを使用してリンクされた実行可能ファイルを再リンクすることはできません。

いくつかのソース・ファイルから構成されているプログラムを持っていて、局所的な変更をいくつかのソース・ファイルに対して行うだけの場合は、必ずしも個々の

ファイルを再コンパイルする必要はありません。その代わりに、変更されたファイルのコンパイル時に、実行可能ファイルを最後の入力ファイルとして組み込むことができます。

```
xlc -omansion front_door.c entry_hall.c parlor.c sitting_room.c \
  master_bath.c kitchen.c dining_room.c pantry.c utility_room.c

vi kitchen.c # OVEN 関数の問題を解決します

xlc -o newmansion kitchen.c mansion
```

2 回目にコンパイルしてリンクするファイルの数を制限すれば、コンパイル時間、ディスクのアクティビティー、メモリー使用量が減少します。

注: リンク操作に習熟していない限り、この種のリンクは避けるべきです。リンクが正しく行われないと、インターフェース・エラーおよびその他の問題が発生する可能性があります。 問題が発生した場合は、 -qextchk コンパイラー・オプションを指定してコンパイルを実行すると、リンクに関する問題を診断することができます。

動的および静的リンク

XL C を使用すれば、ご使用のプログラムは、動的リンクと静的リンクの両方のためのオペレーティング・システム機能の利点を利用できるようになります。

動的リンクとは、プログラムが初めて実行されるときに、なんらかの外部ルーチン用のコードが探し出されてロードされることを意味します。共用ライブラリーを使用するプログラムをコンパイルするときに、共用ライブラリーはデフォルトでプログラムに動的にリンクされます。動的にリンクされたプログラムでは、共用ライブラリーのルーチンを複数のプログラムが使用していると、使用するディスク・スペースと仮想メモリーが少なくて済みます。リンクが行われているときに、それらのプログラムについては、ライブラリー・ルーチンとの命名上の競合を回避するためになんらかの特別な予防措置を講じる必要はありません。いくつかのプログラムが同時に同じ共用ルーチンを使用する場合は、静的にリンクされたプログラムよりも良好に実行する場合があります。また、それらを使用すれば、再リンクしないで共用ライブラリー内のルーチンをアップグレードすることができます。

このリンク形式はデフォルトなので、これを作動させるのに追加のオプションは必要ありません。

静的リンクとは、プログラムによって呼び出されるすべてのルーチン用のコードが実行可能ファイルの一部になることを意味します。

静的にリンクされたプログラムは、XL C ランタイム・ライブラリーがないシステムに移動してそのシステム上で実行することができます。静的にリンクされたプログラムが、ライブラリー・ルーチンへの呼び出しを多数行ったり、多数の小さなルーチンを呼び出す場合、それらのプログラムは動的にリンクされたプログラムよりも良好に実行する場合があります。ライブラリー・ルーチンとの命名の競合を回避したい場合は、プログラム内のデータ・オブジェクトおよびルーチンの名前を選択するときに、何らかの予防措置をとる必要があります。また、それらのプログラムをオペレーティング・システムのあるレベルでコンパイルした後、そのオペレーティング・システムの別のレベルで実行すると、それらは機能しない場合があります。

コンパイル済みアプリケーションの実行

XL C コンパイラーによって作成されたプログラム実行可能ファイルのデフォルト・ファイル名は、a.out です。 -o コンパイラー・オプションを指定して別の名前を選択することができます。

プログラムを実行するためには、コマンド行にプログラム実行可能ファイルの名前を任意の実行時引数と一緒に入力します。

間違ったコマンドを誤って実行することがあり得るので、プログラム実行可能ファイルに、システムまたはシェルのコマンドと同じ名前 (例えば `test` または `cp` など) を付けないでください。プログラム実行可能ファイルにシステム・コマンドまたはシェル・コマンドと同じ名前を付けるように決めた場合には、実行可能ファイルが存在するディレクトリーに、`./test` といったパス名を指定することによって、そのプログラムを実行することができます。

実行の取り消し

プログラムの実行を中断するには、プログラムがフォアグラウンドにある間に Ctrl+Z キーを押してください。実行を再開するには、fg コマンドを使用してください。

実行中のプログラムを取り消すには、プログラムがフォアグラウンドにある間に Ctrl+C キーを押してください。

実行時オプションの設定

環境変数の設定値を使用して、XL C コンパイラーで作成されたアプリケーションの特定の実行時オプションおよび動作を制御することができます。 他の環境変数は、実際のランタイムの動作を制御しませんが、アプリケーションの実行の仕方に影響を与えることがあります。

環境変数の詳細、および環境変数が実行時にアプリケーションにどのような影響を与えるかに関する詳細については、「*XL C インストール・ガイド*」を参照してください。

他のシステムでのコンパイル済みアプリケーションの実行

通常、以前のバージョンの AIX を使用してシステムにリンクされたアプリケーションは、 AIX のさらに新しいバージョンでも作動します。 ただし、新しいバージョンの AIX を使用してシステムにリンクされたアプリケーションは、以前のバージョンの AIX では作動しないこともあります。

XL C コンパイラー診断エイド

XL C は、ユーザーのアプリケーションをコンパイルしているときに問題に遭遇すると、診断メッセージを出します。ユーザーは、そのような問題を識別して訂正する援助として、コンパイラー出力リストで提供されるこれらのメッセージおよびその他の情報を使用することができます。

アプリケーションの問題の解決に役立つリスト作成、診断、関連コンパイラー・オプションについて詳しくは、「*XL C コンパイラー・リファレンス*」の以下のトピックを参照してください。

- 『コンパイラー・メッセージおよびリスト表示』
- 『エラー検査およびデバッグ・オプション』
- 『リスト、メッセージ、およびコンパイラー情報のオプション』

コンパイル済みアプリケーションのデバッグ

シンボリック・デバッガーを使用して、 XL C を使用してコンパイルされたアプリケーションをデバッグすることができます。

コンパイル時に -g または -qlinedebug コンパイラー・オプションを指定することにより、XL C コンパイラーに、コンパイルされた出力にデバッグ情報を組み込むように指示できます。デバッグ・オプションについて詳しくは、「*XL C コンパイラー・リファレンス*」の『エラー検査およびデバッグ・オプション』を参照してください。

次に、AIX XCOFF 実行可能形式をサポートする dbx、IBM Debugger for AIX、 またはその他の任意のシンボリック・デバッガーを使用して、コンパイル済みアプリケーションの動作をステップスルーし、検査できます。

最適化されたアプリケーションをデバッグすると、特殊な問題が発生します。 高度に最適化されたアプリケーションをデバッグするときは、-qoptdebug コンパイラー・オプションの使用を検討してください。コードの最適化について詳しくは、「*XL C プログラミング・ガイド*」の 『アプリケーションの最適化』 を参照してください。

どのレベルの XL C がインストールされているかの判別

ソフトウェア・サポートに連絡して支援を得るには、特定のマシンにインストールした XL C のレベルを確認する必要があります。

ご使用のシステムにインストールされたコンパイラーのバージョンおよびリリース・レベルを表示するために、-qversion コンパイラー・オプションを指定してコンパイラーを呼び出します。

例えば、バージョン情報の詳細を表示するには、コマンド行に以下を入力します。

```
xlc -qversion=verbose
```

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。

〒106-8711
東京都港区六本木 3-2-12
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。

© Copyright IBM Corp. 1996, 2008

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。

Lab Director
IBM Canada Ltd. Laboratory
8200 Warden Avenue
Markham, Ontario L6G 1C7
Canada

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。

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証できません。 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にお願いします。

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、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。お客様は、IBM 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。

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. 1998, 2008. All rights reserved.

商標

IBM、IBM ロゴ、および ibm.com は、International Business Machines Corporation の米国およびその他の国における商標です。この資料が公開された時点で、米国において IBM が所有する登録商標または商標には、初出時に商標記号 (® または ™) を付けて示しています。このような商標は、その他の国においても、登録商標または商標である可能性があります。現時点での IBM の商標については、http://www.ibm.com/legal/copytrade.shtml の「Copyright and trademark information」をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

Cell Broadband Engine, Cell/B.E は、米国およびその他の国における Sony Computer Entertainment, Inc. の商標であり、同社の許諾を受けて使用しています。

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。

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。

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]

[ラ行]

[数字]

A

D

G

M

© Copyright IBM Corp. 1996, 2008

O

S

V

X

［特殊文字］

プログラム番号: 5724-U80

Printed in Japan

GC88-4919-00

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